2018-09-11

Development of a New Parallel In-Situ Combustion Simulator

He, Ruijian

UNIVERSITY OF CALGARY

Development of a New Parallel In-Situ Combustion Simulator

by

Ruijian He

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN CHEMICAL AND PETROLEUM ENGINEERING

CALGARY, ALBERTA

SEPTEMBER, 2018

Abstract

As a competitive recovery method for heavy oil, In-Situ Combustion (ISC) shows its great potential accompanied by technological advances in recent years. Reservoir simulation will play an indispensable role in the prediction of the implementation of ISC projects. With the computational complexity, it is imperative to develop an effective and robust parallel in-situ combustion simulator.

First of all, a mathematical model for In Situ Combustion is proposed. The model takes full consideration for related physical phenomena, including multi-dimensional multi-component three-phase flow, heat convection and conduction, chemical reactions, and mass transfer between phases. In the mathematical model, different governing equations and constraints are involved, forming a complicated PDE (partial differential equation) system. For physical and chemical behaviors, some special treatments for the ISC simulator are discussed and applied. Also, a modified PER (Pseudo-Equilibrium Ratio) method is proposed in the thesis.

To solve the PDE system, a fully implicit scheme is applied, and discretization is implemented with the FDM (Finite Difference Method). In solving nonlinear systems, the Newton Method is introduced, and both numerical and analytical Jacobian matrices are applied. Due to the complexity of an ISC problem, an appropriate decoupling method must be considered. Thus the Gauss-Jordan transformation is raised. Then, with certain preconditioners and iterative solvers, a numerical solution can be obtained.

For the treatment of a thin combustion front, a direct idea is to apply a fine grid to an ISC model. However, the scale of a numerical problem will increase significantly. The approach

given in this thesis is parallelization. Because of their outstanding characteristics, clusters are considered to be the platform of the simulator. Communications and synchronizations are deemed to be the significant factors that affect the parallelization. A dynamic load balancing grid partitioning method, HSFC, is applied for this purpose.

The results of different models are given, which are validated with the results from CMG STARS. Also, the scalability of parallelization is proved, indicating the excellent performance of parallel computing. This accurate, efficient, parallel ISC simulator applies to complex reservoir models.

Keywords: **In-situ combustion, reservoir simulator, parallel**

Acknowledgements

First and foremost, the deepest gratitude goes to Professor Dr. Zhangxing Chen, my supervisor. Without his guidance and encouragement, the thesis could not have reached its present form. I am also thankful to my dissertation committee members, Dr. Robert G. Moore, Dr. Hassan Hassanzadeh, Dr. Haiping Huang and Dr. Shengnan Chen, who gave me valuable comments and suggestions.

This project is a joint work with Bo Yang, I gratefully acknowledge his cooperation in the development of this simulator. I offer special gratitude to Dr. Hui Liu for his valuable advice, continuous guidance all the way through the completion of this work. I wish to thank Min Yang for the ISC models and ideas about mechanisms.

I wish to thank Dr. He Zhong, Dr. Lihua Shen, Dr. Kun Wang and Dr. Jia Luo for their kind help and knowledge in mathematics and computer science.

I am also grateful for all colleagues in Reservoir Simulation Group and all sponsors of Reservoir Simulation Group.

Besides, this research was enabled in part by support provided by Westgrid (www.westgrid.ca) and Compute Canada (www.computecanada.ca).

Table of Contents

List of Figures and Illustrations

List of Tables

List of Symbols, Abbreviations and Nomenclature

Symbols

δt	Time step, t
γ_α	Specific weight of phase α, M/L^3
μ_α	Viscosity of phase α, $M/(Lt)$
$\mu_{c,\alpha}$	Viscosity of component c in phase α, $M/(Lt)$
$\nabla\cdot$	Divergence operator, L
∇	Gradient operator, L
ϕ	Porosity of the reservoir
ϕ_{coke}	Porosity occupied by coke
ϕ_{ref}	Reference porosity at certain pressure and temperature
ρ_α	Molar density of phase α, $mole/L^3$

$\rho_{c,\alpha}$	Molar density of component c in phase α, $mole/L^3$
$\vec{f}$	Right-hand-side vector
$\vec{u}$	Unknown vector
$\vec{u}_\alpha$	Velocity of phase α, L/t
A	A matrix
A	Constant of EOS calculation
a	Constant of EOS calculation, $M^{\frac{1}{2}}T^{\frac{3}{2}}/(t^{\frac{1}{2}}T)$
A_1, A_2, A_3	Cross-sectional area to x, y, z direction, L^2
A_R	Reaction frequency factor of of R-th reaction, unit is variable
avg	Coefficient of viscosity calculation for gas phase, $M/(Lt)$
$avisc$	Coefficient of viscosity calculation, $M/(Lt)$
B	Constant of EOS calculation
b	A vector
b	Constant of EOS calculation, $1/T^2$
bvg	Coefficient of viscosity calculation, T
$bvisc$	Coefficient of viscosity calculation, T
C_c	Concentration of coke, $mole/L^3$

$C_{c,max}$	Limitation of concentration of solid phase, $mole/L^3$
cond	Bulk thermal conductivity, $ML/(Tt^3)$
cp	Compressibility to pressure of density calculation, Lt^2/M
*cpg*1	Constant of vapor enthalpy calculation, $ML^2/(Tt^2mole)$
*cpg*2	Constant of vapor enthalpy calculation, $ML^2/(T^2t^2mole)$
*cpg*3	Constant of vapor enthalpy calculation, $ML^2/(T^3t^2mole)$
*cpg*4	Constant of vapor enthalpy calculation, $ML^2/(T^4t^2mole)$
cpor	Compressibility to pressure of porosity calculation, Lt^2/M
cpt	Compressibility to pressure and temperature of density calculation, $Lt^2/(MT)$
cptpor	Compressibility to pressure and temperature of density calculation, $Lt^2/(MT)$
*ct*1	Compressibility to temperature of density calculation, $1/T$
*ct*2	Second-order compressibility to temperature of density calculation, $1/T^2$
ctot	Compressibility coefficient for calculating porosity
ctpor	Compressibility to pressure and temperature of porosity calculation, $1/T$

$E_{a,R}$	Reaction activation energies of R-th reaction, $ML^2/(t^2 mole)$
ev	Constant of heat of vaporization calculation
$f_{c,\alpha}$	Fugacity of component c in phase $alpha$
g	Acceleration of gravity, L/t^2
H	Enthalpy, $ML^2/(t^2 mole)$
h_1, h_2, h_3	Grid step size in x, y, z direction, L
H_g	Enthalpy of gas, $ML^2/(t^2 mole)$
H_R	Reaction enthalpy of of R-th reaction, $ML^2/(t^2 mole)$
H_v	Enthalpy of heat of vaporization, $ML^2/(t^2 mole)$
hvr	Reference heat of vaporization of heat of vaporization calculation, $ML^2/(t^2 mole)$
K	K-value of component c for phase equilibrium
k	Absolute permeability, L^2
K^*	Pseudo K-value of component c for phase equilibrium
K_T	Bulk thermal conductivity, $ML/(Tt^3)$
$k_{r\alpha}$	Relative permeability of phase α
k_{rocw}	Oil-water two-phase relative permeability to oil at connate water saturation

k_{rog}	Oil-gas two phase relative permeability to oil
k_{row}	Oil-water two phase relative permeability to oil
$kv1$	Constant of K-value calculation, $M/(Lt^2)$
$kv2$	Constant of K-value calculation, Lt^2/M
$kv3$	Constant of K-value calculation, no unit
$kv4$	Constant of K-value calculation, T
$kv5$	Constant of K-value calculation, T
N	Stoichiometric matrix
N_p	Number of processors
N_α	Number of phases
n_α	Total amount of material in phase α, *mole*
$n_{c,\alpha}$	Molar amount of component c in phase α, *mole*
n_{cg}	Number of gas components
n_{co}	Number of oil components
N_c	Number of components
N_R	Number of reactions
nx	Number of cells in x-direction

ny	Number of cells in y-direction
nz	Number of cells in z-direction
p	Pressure, $M/(Lt^2)$
p_{bh}	Bottom hole pressure, $M/(Lt^2)$
p_{crit}	Critical pressure, $M/(Lt^2)$
p_{ref}	Reference pressure, $M/(Lt^2)$
$q_{\alpha,well}$	Well rate of phase α, $mole/(L^3t)$
$q_{c,reac}$	Reaction rate of phase α, $mole/(L^3t)$
Q_{loss}	Heat loss rate, ML^2/t^3
Q_{reac}	Total reaction enthalpy, ML^2/t^3
R	Universal gas constant, $ML^2/(Tt^2mole)$
r_e	Equivalent radius, L
r_R	Reaction rate of R-th reaction, $mole/(L^3t)$
r_w	Radius of the well, L
s	Skin factor of a well
S_α	Saturation of phase α
$s_{c,R}$	Stoichiometric coefficient of component c in reaction R

T	Temperature, T
t^n	n-th Time step, t
t_{crit}	Critical temperature, T
t_{end}	End time, t
T_{ref}	Reference temperature, T
t_{start}	Start time, t
U	Internal energy, $ML^2/(t^2 mole)$
$u(x)$	Heaviside unit step function. If $x > 0$, $u = 1$; if $x < 0$, $u = 0$
wi	Well index of a well, L^3
$x_{c,\alpha}$	Molar fraction of component c in phase α
x_c	Molar fraction of component c in oil phase
y_c	Molar fraction of component c in gas phase
Z	Compressibility factor of real gas
z	Depth, L
z_{bh}	Reference depth of p_{bh}, L

Subscripts and superscript

α	Phases (oil, water, gas)

bh	Bottom hole
c	Center cell
c	Coke
c	Component
$face$	Spatial direction, defined by number 0 to 5, referring to up, down, front, back, left right seperately
G	Gas component
g	Gas phase
i	Index of component in a phase
i, j, k	Index of cell in three directions (x, y, z)
k	Newton iteration level
n	Time iteration level
O	Oil component
o	Oil phase
$perf$	Perforation
r	Rock
$reac$	Reaction

s	Solid phase
W	Water component
w	Water phase
$well$	Well

Chapter 1

Introduction

1.1 Background

The International Energy Agency reports that global demand for oil will rise from 90 million barrels per day in 2012 to 121 million barrels per day in 2040. The main driver of the increase will be China and India (Conti, J. et al. (2016)). However, reserves of conventional crude oil are declining, and the emphasis is shifting from conventional oil production to the development of unconventional sources, including oil sands. Canada's proven petroleum hydrocarbon reserves include the oil sands in the Athabasca Basin in northeastern Alberta and Saskatchewan. These deposits, regarded as a safe and secure source of oil for North America, are the second largest in the world (Kelly, E. N. et al. (2010)). However, these oil sands resources have extremely high viscosity and cannot flow under the reservoir conditions. To reduce its viscosity, thermal recovery methods, for example, Steam Flooding, Cyclic Steam Stimulation (CSS), Steam Assisted Gravity Drainage (SAGD) and In-Situ Combustion (ISC), are widely used in reservoirs in Canada.

In-Situ Combustion was first discovered in the 1900s and has been researched for decades. Heat generated in-situ by oxidation of hydrocarbons in a reservoir can effectively reduce the viscosity of bitumen, which makes ISC highly energy efficient and productive with outstanding recovery rates, thus becoming attractive especially to deep or thin reservoirs. However, field applications of ISC have not yielded as much success in Canada as compared to other thermal methods such as SAGD. It is considered that the challenge of controlling and predicting the process of ISC is the key obstacle.

The first planned successful in-situ combustion project took place in October 1920 in southern Ohio, U.S (Mills, 1923). Despite the long history and commercial success of many field projects, the process has not found widespread acceptance among operators due to the excessive number of failures in many early field trials. Most of these failures can be traced to the inappropriate application of the process in poorer prospects (Sarathi, P. (1999)). By adopting the most updated technology planning, implementation, operation and managing firefloods, and applying reliable numerical simulation of ISC, operators will be able to minimize the risk of failures.

In contrast to the steam injection method, steam in the ISC process is generated in the reservoir. An oxygen-containing gas such as air is injected into the reservoir. And ignition is done near the wellbore, forming an oxidation zone, where oxidation reactions occur between hydrocarbons and oxygen. Heat generated from the reactions raised the temperature of a reservoir adjusted to an oxidation zone, where the viscosity of heavy oil is reduced. Also, steam and gases produced by the reactions, besides the propagation of a combustion front, displaced oil near the oxidation zone toward a producing well (Moore, R. G. et al. (1995);

Mahinpey, N., Ambalae, A., and Asghari, K. (2007); Ursenbach, M. G., Moore, R. G., and Mehta, S. A. (2010); Chen, X. et al. (2014)).

Usually, a dry forward ISC is implemented. In this process, hydrocarbons are ignited near an injection well using methods such as spontaneously ignition, ignition with electric heaters or burners, and chemical ignition. The combustion begins at the injection well and then propagates towards a producing well. In the process, behind a combustion zone is the burnt zone, where saturation of injected gas is almost 100% and nearly all hydrocarbons have been displaced. An oxidizing agent (air or oxygen-containing gases), which is injected continuously, gets contact with fuel, usually coke, a reaction product of oxidation and cracking reaction. In this zone, the temperature is typically above 500 ^{o}C for heavy oil. Beside the combustion front, there is a cracking zone, where the cracking reaction occurs. In the reaction, oil cracks and forms coke in place, providing fuel for the combustion zone; light components and water vaporize and are being pushed forward. For this reason, the zone is also called the vaporization zone. In front of this zone, the vapor gets in contact with the cold reservoir, carrying away a large amount of heat, by heat conduction and latent heat release in liquefaction increases the temperature of the reservoir. With the enormous heat, the viscosity of oil decreased rapidly, thus increasing its mobility so that it is possible for the oil to flow forward in the direction of the producing well. Because there is steam formed from connate water and water produced from the reaction, this zone is defined as a steam zone, where the temperature is reduced to a level slightly higher than that of the native reservoir. And mobilized oil formed an altered saturation zone (Moore, R. G. et al. (1995); Sarathi, P. (1999); Sequera, B. et al. (2010); Ursenbach, M. G., Moore, R. G., and Mehta, S.

A. (2010)). Figure 1.1 indicates the process of the dry forward ISC, with each characteristic region depicted, and temperatures and saturations are also indicated.

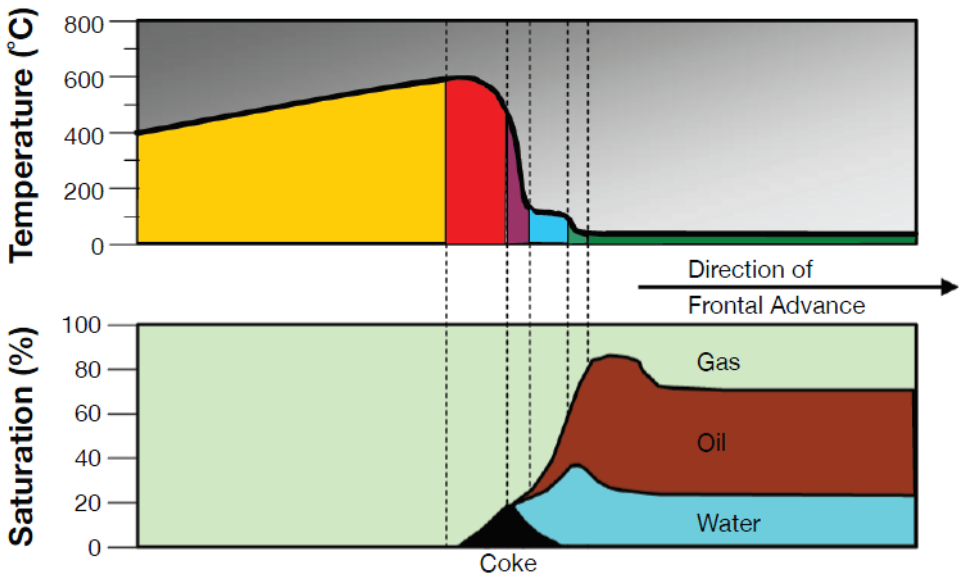

Figure 1.1: Schematic diagram illustrating the temperatures and saturations associated with the dry in-situ combustion process for heavy oils and bitumen (Ursenbach, M. G., Moore, R. G., and Mehta, S. A. (2010))

Besides water flooding, ISC is the IOR (Improved Oil Recovery) method that is most widely applicable. And it has significant economic advantages.

First, it is the most thermally efficient recovery process. ISC usually uses air, the least expensive and the most readily available fluid, as the injectant. Also, ISC only takes a heavy component in oil as the fuel in the combustion process, which merely consists 10% of the oil in the reservoir, a small amount considering its high recovery rate. On the other hand, with the cracking reactions of heavy components, the quality of heavy oil can be significantly improved.

ISC can recover oil economically from a variety of reservoir settings. The process has proven to produce economical oil from not only shallow reservoirs but also deep ones (even for

11,000 ft.), which other recovery methods are incapable of. For example, in deep reservoirs, steam injection is not effective due to excessive heat losses, and water flooding is economically unattractive for high lifting costs. In such reservoirs, combustion and gas are the only processes that can be applied to recover oil economically.

Though most combustion methods were originally designed especially for heavy oil reservoirs, they are increasingly used to recover light oil from deep reservoirs. In the U.S., more combustion projects are in operation in light oil reservoirs than in heavy oil reservoirs, which is known as the High-Pressure Air Injection process (HPAI) (Fassihi, M. R., Yannimaras, D. V., and Kumar, V. K. (1997); Gutierrez, D., Skoreyko, F., et al. (2009); Montes, A. R. et al. (2010)).

Also, it is an ideal process for thin formations. Economically, successful projects have been implemented in reservoirs with a gross pay of 4-150 feet. Pernicious effects due to heat loss to underburden and overburden cannot be neglected for the steam injection process, but it is not the case for the combustion process. Reservoir pressure does not affect the technical success of the process and permeability has little effect on the process. ISC has been successfully implemented in formations whose permeability ranges from 5 md to 10,000 md. A crucial factor of ISC is that its recovery efficiency is better than other oil recovery processes. Its displacement efficiency can be as high as 95% and compound recovery efficiency can be 56% (Hasiba, H. H. and Wilson, L. A. (1974)).

Furthermore, this process can be implemented as a follow-up to water flood and steam injection processes. Combustion projects allow for the use of wider well spacing and achieve higher ultimate oil recovery in comparison to steam injection methods. However, one of the

main limitations of ISC is that its field performance is difficult to control and predict.

Reservoir simulation is an area of reservoir engineering in which computer models are used to predict the flow of fluids (typically, oil, water, and gas) through porous media. The purpose of the simulation is to estimate field performance, for example, oil recovery under one or more producing schemes. A model can be run many times at a low expense over a short time span, while the field can be produced only once. Results of simulation, representing different producing conditions, assist the selection of an optimal set of strategies for the reservoirs. With an effective leverage of reservoir simulation, the economic return can be maximized during the recovery of hydrocarbons.

Before a reservoir project is implemented, computer modeling helps management identify the proper recovery process, well layout and completion, the present and future needs for artificial lift and the expected production of oil, water, and gas. After the project is put into operation, computer modeling can help improve oil recovery by adjusting production strategies. Highly deviated and horizontal wells can also be simulated to improve production rates. In addition, the re-injection of produced gas, or a different fluid, as an enhanced oil recovery process, can be evaluated during the later period of field production. Simulation can assist in minimizing project risk and making informed investment decisions based on results in a field operation.

Grid cells, the core of a discrete computer model, are traced by a structured or unstructured grid and store the physical properties of a reservoir. Typically, the computer model can be a geological model that describes static properties, or a reservoir simulation model that describes the flow, thermal and chemical processes throughout the entire life of the

recovery of the reservoir by solving partial differential equations numerically.

Many attempts were made for the early mathematical models. In the late 1950s, conduction was assumed to be the only one mode of heat transfer. And in 1960, convection was also included, but heat losses to the strata were neglected (Ramey Jr, H. J. (1959); Bailey, H. R. and Larkin, B. K. (1959); Bailey, H. R. and Larkin, B. K. (1960)). Chu proposed a numerical model for the radial flow of combustion with conduction, convection and heat loss (Chu, C. (1963)). Thomas also described an analytical treatment of combustion in an infinite system (Thomas, G. W. (1963)). Since computers were introduced to the oil industry in the 1950s, engineers began to leverage mathematical models in performing reservoir engineering calculations.

Gottfried presented a linear numerical model which described the thermal recovery of oil. In his model, three-phase flow, conduction, convection, heat loss and chemical reactions between oxygen and oil were all included. Mass balance equations and Darcy's laws were introduced, which still constitute the fundamentals of reservoir simulation today. There were some limitations due to the complexity of the numerical model and uncertainties in numerical values, although the model still represents a major advance to simulate the ISC process (Gottfried, B. S. (1965)).

James and Farouq Ali developed a 2D mathematical model of the forward combustion process. This model assumed a single phase flow and a constant fuel consumption. The conventional implicit differencing scheme was adopted with several explicit techniques. The model incorporated changes of permeability in a burnt zone, heat transfer by conduction and convection, and heat loss in a reservoir (Smith, J. T. and Ali, S. M. (1971)).

Alder presented a new mathematical model of in-situ combustion as a further development of Gottfried's model. The model included physical-chemical processes, including combustion of the oil components both in the oil phase and the gas phase, as well as the effect on the viscosities and heat transfer. The oil is considered as a mixture of two components with evaporation. And the solution of combustion gas in the oil and aqueous phases also constituted an important factor (Adler, G. (1975)).

Farouq Ali developed a 2D three-phase four-component (one oil component) simulator for in-situ combustion. The solution scheme employed an iterative solution in pressure and an explicit solution in other variables. The in-place oil was assumed to be consumed by oxidation, representing an Arrhenius-type reaction equation, neglecting coke formation and oxidation. Also, an implicit pressure and explicit saturation (IMPES) method was used, which was somewhat slow, and, therefore, a higher implicit level was suggested (Ali, S. M. (1977)).

Crookston et al. presented a semi-implicit two-dimensional three-phase-flow model with vaporization-condensation effects. A well rate was calculated semi-analytically and Newton iteration was not used for well calculation. The primary focus was on the simulation of in-situ combustion, where four chemical reactions were accounted for: formation of coke from the heavy hydrocarbon component and the oxidation of coke and both heavy and light components (Crookston, R. B., Culham, W. E., and Chen, W. H. (1979)).

Rubin and Vinsome developed a comprehensive one-dimensional in-situ combustion simulator, which was highly implicit and stable, providing an option of the sequential implicit scheme that made it possible to extend to a two- or three-dimensional model (Rubin, B. and

Vinsome, P. K. W. (1980)).

Grabowski et al. introduced the sequential implicit method into a thermal model, describing both in-situ combustion and steam processes. They also discovered that the fully implicit scheme of the solution method guaranteed simplicity of programming and robustness with a proper time step (Grabowski, J. W. et al. (1979)).

Youngren described a 3D three-phase in-situ combustion simulator which factored in five components: water, oxygen, two arbitrary volatile components distributed between the oil and gas phases, and non-volatile oil for combustion. The equations of the model were five material balance equations, an energy conservation equation and a gas-phase mole fraction constraint. In the interest of time for calculation, only the most important essential variables, such as combustion reactions and fluid flow, were treated implicitly (Youngren, G. K. (1980)).

Coats put forward a general fully-implicit four-phase multi-component multi-dimensional combustion model. This model extended the generality of previous models and reduced the computing time. Any numbers of phases and components were allowed and any chemical reactions were extensive. In Youngren's simulator, it was assumed that a combustion rate equaled an oxygen flux, but it was no longer required in Coats's model. And a fully implicit scheme is provided, which maximized efficiency and minimized time consumption; with an implicit scheme, a longer time step size was allowed, so the number of total time steps was reduced (Coats, 1980).

Hwang, Jines, and Odeh developed a new black-oil simulator for an ISC process named THERMS. It simulated field scale ISC projects, with experimental data acquired from lab-

oratory combustion tube runs. Notably, a moving front was considered as a burning zone, which was both a moving heat source and a displacement pump that enhanced oil flow. The model was designed with field-oriented features, and the simulator could also be used for other recovery methods such as hot water flooding, steam injection and gas displacement (Hwang, M. K., Jines, W. R., and Odeh, A. S. (1982); Odeh, A. S. (1969)).

Sensitivity studies were made for the in-situ combustion simulators. Anis et al. found that flow convection parameters, such as a grid block size and dimensionality, were the crucial factors. Numerical distortion of the convection effects with large grid blocks limits the utility of such models (Anis, M., Hwang, M. K., and Odeh, A. S. (1983)). Lin et al. used Crookston's model and found that simulation results were always sensitive to the K-values of a light oil component. Therefore, more than two crude oil components were required to simulate the evaporation effect of crude oil accurately (Lin, C. Y. et al. (1984)).

Rubin and Buchanan improved the previous model by introducing a fully coupled implicit wellbore model, which eliminated the lagging of a rate or pressure between the wellbore and the reservoir. They reduced the implicitness by expressing the pressure loss along the well through an explicit hydraulic gradient. Also, the simulator made use of powerful incomplete factorization iterative methods to allow the practical solution of large problems (Rubin, B. and Vinsome, P. K. W. (1985)).

Ito and Chow focused on the field-scale simulation of ISC and developed a new and numerically stable algorithm to achieve the desired fuel consumption. This algorithm involved an oil-flow-enhancement scheme. Arrhenius-type reactions were found to be inadequate in field-scale combustion simulation. A new pseudo-kinetic scheme was introduced to field-scale

studies in which a reaction zone was not simulated accurately because of a large size of grid blocks (Ito, Y. and Chow, A. K. Y. (1988)).

Oklany noticed that the in-situ combustion models in the past were private and he developed a model accessible to the academic community. His model included multi-phase multi-component and multi-dimension. A code in FORTRAN was also presented in his thesis. He noticed that when a multi-dimensional case runs on the simulator, the computational power turned into a bottleneck at that time (Oklany, J. S. F. (1992)).

For the ISC process, simulation is complicated, because a very fine grid is required to capture a combustion front. Because of a thin reaction zone, many previous models are not adequate to resolve a front structure, and an accurate field simulation is extremely time-consuming.

Traditionally, kinetic upscaling has been applied to field scale ISC simulation. It is well known that the kinetics developed from laboratory scale experiment simulation cannot be directly used for field scale simulation to produce a reliable prediction. The Arrhenius equation shows that the reaction rate depends on temperature and time through the fuel dependence. The dependence on temperature makes the reactions sensitive to a grid size because large grid blocks need more heat to achieve a certain temperature level (Gutierrez, D., Moore, R. G., et al. (2012)). Several attempts have been made to upscale kinetics for field scale simulation. In some cases, the activation energy was adjusted to a low level and, therefore, the oil burned more easily (Coats, K. H. (1983)).

Zhu et al. proposed a new upscaling method with non-Arrhenius kinetics. In their method, the most important parameters (e.g., the fuel amount) for ISC design were ex-

tracted, which were then incorporated into the field scale reservoir simulation. When a reaction front was propagating through a reservoir, the reaction source and sink terms were calculated based on the fuel amount. In this method, the ISC front was treated as a subgrid phenomenon. Equivalent reaction effects (e.g., the amount of heat released, fuel consumed, and reaction products) were substituted into field scale simulation grid blocks (Zhu, Z. (2011a); Zhu, Z. (2011b)).

Nissen et al. further extended previous work and proposed the Work-Flow-Based Upscaling-for-Grid-Independence (WUGI) scheme, in which Arrhenius kinetics was replaced by equivalent sources terms to represent heat generation and fuel consumption (Nissen, A. et al. (2015)).

Akkutlu and Yortsos proposed a different approach for ISC simulation with an analytical framework for an ISC combustion front. They used a transport limiting approach for reactions, treated a combustion zone as a discontinuous front and derived appropriate jump conditions relating a change in variables across the front. Then the problem was reduced to modeling the combustion front dynamics and the regions on either side of the front (Akkutlu, I. Y. and Yortsos, Y. C. (2003)).

Kristensen designed two ISC models: the Virtual Combustion Tube (VCT) and the Virtual Kinetic Cell (VKC). VCT is an equation of state compositional ISC model, which accounts for the simultaneous flow, transport of heat and mass, chemical reactions and phase equilibrium. The VKC is a zero-dimensional version of the VCT, which models a multiphase, chemical batch reactor, which facilitates the isolated study of chemical reactions and phase behavior. Together, the two models provide a flexible platform and play an important role in ISC simulation (Kristensen, M. R. (2008)).

A dynamic gridding technique is a popular approach for field scale ISC simulation (Sammon, P. H. (2003); Christensen, J. R. et al. (2004)). It allows the use of small grids in the regions where most of the sharp fronts occur (i.e., a combustion front) and coarser grid blocks in the other region. An implementation of dynamic grid refinement has been achieved in a commercial thermal simulator, STARS, a product of Computer Modelling Group Ltd. (CMG), which is the industry standard today for ISC simulation. The technique can reduce a grid number in the simulation and decreases the computation time. It was found that dynamic gridding is able to reduce the CPU time by up to two-thirds while keeping good accuracy when applied to combustion and SAGD simulations (CMG (2015)).

1.2 Objective

Because of the complexity of the physical process, the in-situ combustion simulators are computationally complex. Parallel supercomputers become more and more computational powerful in the past years. With the parallelization, the power of a simulator for the in-situ combustion process will increase significantly. However, ISC simulators on supercomputers, especially on shared-memory parallel computers, which can run on thousands of CPUs, are not currently available yet. The primary objective of this study is to design a capable and robust parallel In-Situ Combustion simulator on shared-memory computers. The simulator aims to be run on advanced research computing (ARC) systems provided by Compute Canada. The number of CPUs employed is only limited by the hardware. Therefore, the computational power can increase to any required level ideally.

In the development of the simulator, our research will focus on some parts.

First of all, the model in the simulator is adopted from historical research. However, some modifications are required for a general model. Also, the simplicity of parallelization should also be considered. For the phase equilibrium, a modified PER (Pseudo-Equilibrium Ratios) method is purposed, which will prevent a singular phenomenon of an ISC problem.

Besides, in the solution process of the ISC simulator, traditional decoupling methods are not so powerful. A new decoupling method is proposed especially for this complex problem. With this method, the equation alignment problem is also solved.

Furthermore, for parallelization, some new parallel algorithms are first introduced to the ISC simulator. Parallel grid management, parallel preconditioners, and parallel solvers are all employed. Especially for the grid partitioning method, a dynamic load balancing method is newly adopted from thermal simulators, achieving high partitioning quality in a shorter time.

As a result, the simulator with the improvements above illustrates its effectiveness. The accuracy is expected to be shown in the validation with the results from CMG STARS. Also, the profiles of some variables are shown to coincide with their physical phenomena. In parallelization, scalability and parallelization should be guaranteed in the simulator.

The development will become an essential module of a Reservoir Simulation platform in the Reservoir Simulation Group. With specific physical and chemical properties directly coming from the user's input, the simulator is not merely an illustration of several particular models but a general one. Finally, with our simulator, accurate predictions can be made on the performance of reservoirs.

1.3 Outline of the thesis

This thesis comprises six chapters. Specifically, Chapter 1 is an introduction to the topic together with the major research objectives. Chapter 2 introduces a comprehensive mathematical model of the ISC process. Chapter 3 presents the methodologies of numerical solutions. Chapter 4 focuses on the parallel implementation of the ISC simulator. Chapter 5 provides the results and validations as well as the performance of parallelization of the simulator. Finally, Chapter 6 put forward conclusions and future work for the whole project.

Chapter 2

Mathematical Model

The mathematical model and governing equations for in-situ combustion are introduced. For the ISC process, the mathematical model is a system of nonlinear partial differential equations describing multi-phase multi-component fluid flow and heat transfer. Chemical reactions and pressure-volume-temperature (PVT) relationships of the reservoir fluids are all included.

2.1 Mathematical description

Several characteristics of physical and chemical processes are described, including:

- Viscosity, gravity, and capillary forces;
- Heat conduction and convection processes;
- Heat losses to overburden and underburden of a reservoir;
- Mass transfer between phases;

- Effects of temperature on the physical property parameters of oil, gas, and water;
- Chemical reactions, resulting in a change in mass of certain components and temperature;
- Rock compression and expansion.

These factors can all be reflected in different governing equations, constraints and calculations of properties. With all equations and properties combined, the In-situ Combustion process can be well described.

2.1.1 Governing equations

2.1.1.1 Darcy's Law

The velocity of a fluid phase is the key to the flow of that fluid, which can measure material and energy spatial exchange. In porous media, permeabilities of different phases are different, where effective and relative permeabilities are introduced. In addition, the pressure and viscosity of different phases differ. Therefore, th velocity of each phase is considered separately by Darcy's law. The solid phase, representing the existence of coke, does not flow in porous media. There are three flow equations for the gas phase, the oil phase and the water phase, with a lower case denoting each phase with g, o, w separately

(Chen, Z. (2007)).

$$\begin{aligned}
\vec{u}_w &= -\frac{k_{rw}}{\mu_w}\vec{k}\,(\nabla p_w - \gamma_w \nabla z) \\
\vec{u}_o &= -\frac{k_{ro}}{\mu_o}\vec{k}\,(\nabla p_o - \gamma_o \nabla z) \\
\vec{u}_g &= -\frac{k_{rg}}{\mu_g}\vec{k}\,(\nabla p_g - \gamma_g \nabla z)
\end{aligned} \tag{2.1}$$

In the above equations, $\gamma_\alpha, \alpha = w, o, g$, is the specific weight of a certain phase, which is computed by the phase density ρ_α and acceleration due to gravity g. μ_α and $k_{r\alpha}$ denote the dynamic viscosity and relative permeability of a phase. Darcy's law applies to laminar flow, where the fluid movement is dominated by viscous forces. This occurs when a fluid moves slowly, along parallel streamlines. When a flow velocity increases or a kinetic viscosity decreases, a fluid moves chaotically and the streamlines are no longer in parallel formation. Flow acts in turbulent flow and the inertial forces are more influent than the viscous forces.

2.1.1.2 Mass conservation equations

For a multi-phase, multi-component system, we introduce the molar fraction $x_{c,\alpha}$ of a component in the α-phase. The molar number of a component in a phase and the total molar number of the phase are denoted as $n_{c,\alpha}$ and n_α, respectively. Thus the molar fractions are

$$x_{c,\alpha} = \frac{n_{c,\alpha}}{n_\alpha}. \tag{2.2}$$

The total molar number of component c is conserved as below (Chen, Z. (2007)):

$$\frac{\partial}{\partial t}\left(\phi\Sigma_{\alpha}^{N_\alpha}\rho_\alpha S_\alpha x_{c,\alpha}\right) = -\nabla\cdot\left(\Sigma_{\alpha}^{N_\alpha}\rho_\alpha S_\alpha \vec{u}_\alpha\right) + \Sigma_{\alpha}^{N_\alpha} q_{\alpha,well} x_{c,\alpha} + q_{c,reac}. \tag{2.3}$$

Also, we have a single mass conservation for coke (Oklany, J. S. F. (1992)):

$$\frac{\partial}{\partial t} C_c = q_{c,reac}. \tag{2.4}$$

Since the solid phase does not flow, there is no convection term in the equation. Only chemical reaction terms can affect the coke concentration.

In this equation, it is noticeable that different from other models, the "mass" conserved here is only the molar number rather than the mass. Also, ρ_α and q_α are the molar density and molar production/injection of phase α. Specifically, $q_{c,reac}$ here represents a reaction term, which has a similar effect as a well term. When a component is the reactant of a reaction, the sign of the reaction term can be negative, since the molar number of that component will decrease when the reaction takes place. Also, if a component is the product of a reaction, the sign can be positive due to an increase in its molar number during the reaction. This reaction term can be a sum of several reactions, since a component can be a reactant or a product in different reactions. For example, oxygen is consumed in each oxidation reaction with different rates, so this term can be a superposition.

2.1.1.3 Energy conservation equation

The energy conservation equation must be taken into account besides the mass conservation equation for a thermal process (Chen, Z. (2007)):

$$\begin{aligned} & \frac{\partial}{\partial t}\left(\phi(\rho_w S_w U_w + \rho_o S_o U_o + \rho_g S_g U_g) + (1-\phi)U_r + C_c U_c\right) \\ = \quad & \nabla \cdot (K_T \nabla T) - \nabla \cdot (\rho_w H_w \vec{u}_w + \rho_o H_o \vec{u}_o + \rho_g H_g \vec{u}_g) \\ & + (q_{w,well} H_w + q_{o,well} H_o + q_{g,well} H_g) + Q_{reac} - Q_{loss}. \end{aligned} \tag{2.5}$$

In the equation above, the only causes for an energy change are heat conduction, heat convection, and kinetic and potential energy. The effects of a viscous force and an energy addition due to radiation and electromagnetism are ignored.

On the left-hand side of the equation, the variations in the bracket represent the molar internal energy of all fluid phases and the solid phase (which only contains coke as a component), as well as the rock. U denotes the volumetric internal energy.

On the right-hand side, the first term represents the conduction term. This is caused by a difference in temperature, where the rate of conduction is constraint by K_T, the bulk thermal conductivity. The thermal conductivity here is a combination of liquid, rock and solid matter, where a linear mixing rule is applied (CMG (2015)). For the ISC process, coke is considered in the normal mixing rule:

$$K_T = \phi\left[S_w K_w + S_o K_o + S_g K_g\right] + (1-\phi)K_r + C_c K_c. \tag{2.6}$$

In the equation, K_w, K_o, K_g, K_r, K_c denote thermal conductivities for water phase, oil phase, gas phase, rock and coke separately.

The second term denotes the heat convection, a result of fluid flow. The molar enthalpy of different phases are different and an enthalpy change occurs when a fluid flows. The third term represents the energy exchange rate due to a source/sink well, which is negative for production and positive for injection. The expression is similar to the convection term, since both describe an enthalpy change due to fluid flow. For the fourth term, a thermal effect of reactions is described, which can be a positive term (exothermic reaction) or a negative term (endothermic reaction).

A heat loss term to underburden and overburden is also considered in the last term (Chen, Z. (2007)):

$$Q_{loss} = AK_{ob}\left(\frac{T(ob) - T_{ini}}{d} - \rho\right), \tag{2.7}$$

where A is the interface area with overburden/underburden, $T(ob)$ is the temperature at the interface, and ρ is a fitted parameter.

2.1.1.4 Reaction terms

For an ISC simulator, the main difference between the ISC simulator and a pure thermal recovery simulator is the involvement of chemical reactions. As mentioned above, reaction terms are added to the mass conservation equations and energy conservation equation. For the mass conservation equation of component c, the reaction term $q_{c,reac}$ can be expressed

as below (Chen, Z. (2007)):

$$q_{c,reac} = \sum_{R}^{N_R} s_{c,R} r_R, \tag{2.8}$$

where r_R is the reaction rate of reaction R and $s_{c,R}$ is the stoichiometric coefficient of component c in reaction R. With a stoichiometric coefficient, if component c is a reactant, then we note that this coefficient is negative. If a component is a product, then it will be taken as a positive number. If it does not participate in a reaction, the coefficient will then be zero. With the stoichiometric coefficients, the chemical reaction equation can be expressed as follows:

$$\sum_{c}^{N_c} s_{c,R} \cdot (\text{Chemical formula of component } c) \longrightarrow 0. \tag{2.9}$$

With all the stoichiometry matrix listed together, the following stoichiometry matrix can be obtained:

$$N = (s_{c,R})_{c,R} = \begin{bmatrix} s_{1,1} & s_{2,1} & \dots & s_{N_c,1} \\ s_{1,2} & s_{2,2} & \dots & s_{N_c,2} \\ \dots & \dots & \dots & \dots \\ s_{1,N_R} & s_{2,N_R} & \dots & s_{N_c,N_R} \end{bmatrix}. \tag{2.10}$$

When a chemical reaction model is input into our simulator, all the coefficients must be checked. Otherwise, mass conservation of the total system may not hold. For chemical

reactions that obey the law of conservation of matter, the following equation holds:

$$N\vec{M} = \vec{0}, \tag{2.11}$$

where $\vec{M}$ is the vector of molar masses of components. The c-th entry of the vector is the molar mass of component c.

As for the reaction term in the energy conservation equation, it can be expressed as:

$$Q_{reac} = \sum_{R}^{N_R} H_R r_R, \tag{2.12}$$

where H_R is the reaction enthalpy of the R-th reaction (Coats, K. H. (1980)).

Expressions of reaction terms in both the mass conservation equations and energy conservation equation are all in the form of a linear combination of reaction rates. Therefore, calculations of a reaction rate are crucial in the ISC simulator.

2.1.2 Constraint equations

Furthermore, some constraint equations are required to solve these conservation equations. Constraint equations are functions that do not involve any convection or conduction term and are only dependent on local properties. They express all secondary variables and parameters as functions of a set of primary thermodynamic variables selected.

2.1.2.1 Phase saturation constraint

The saturation constraint forces all the available pores of rock to be saturated with fluids, which is encountered for in all multi-phase flows (Chen, Z. (2007)):

$$S_w + S_o + S_g = 1. \tag{2.13}$$

This constraint indicates that a pore volume is occupied only by the water, oil, and gas phases. Specifically, we only take these three phases into account as coke does not take a volume obviously, and, therefore, a saturation for the solid phase is not introduced.

2.1.2.2 Capillary pressure constraints

A constraint is a conversion from the oil pressure (the default system pressure) to the water pressure and gas pressure. A capillary pressure P_c is the pressure difference across the interface between two immiscible fluids arising from capillary forces. These capillary forces are surface tension and interfacial tension. Capillary pressures are usually functions of saturation. In a three-phase system, we have the following relationship (Chen, Z. (2007)):

$$p_w = p_o - p_{cow}(S_w), \quad p_g = p_o + p_{cog}(S_g). \tag{2.14}$$

These capillary pressures are assumed to be known.

2.1.2.3 Phase composition constraints

A constraint implies that the sum of all the components' mole fractions in a phase adds up to one, which is usually encountered for in compositional flow (Chen, Z. (2007)):

$$\Sigma_{\alpha}^{N_\alpha} x_{c,\alpha} = 1, \quad \alpha = w, o, g. \tag{2.15}$$

It comes from the total mole number of a given phase that

$$\Sigma_{\alpha}^{N_\alpha} n_{c,\alpha} = n_\alpha, \quad \alpha = w, o, g. \tag{2.16}$$

2.1.2.4 Phase equilibrium constraints

A constraint characterizes the variation of mass distribution of a given component in its phases. Based on the assumption that the mass interchange between phases occurs much faster than the flow in porous media, all the phases are in the equilibrium state. Consequently, the distribution of each component is subjected to the condition of a stable thermodynamic equilibrium, which is given by minimizing the Gibbs free energy of the compositional system. The component distributed phases are arbitrary and determined by the component properties. The following equation expresses the thermodynamic equilibrium state of component c being distributed in two different phases α_1 and α_2 (Chen, Z. (2007)):

$$f_{c,\alpha_1}(p_{\alpha_1}, T, x_{1,\alpha_1}, \cdots, x_{N_c,\alpha_1}) = f_{c,\alpha_2}(p_{\alpha_2}, T, x_{1,\alpha_2}, \cdots, x_{N_c,\alpha_2}). \tag{2.17}$$

In the equilibrium equation, $f_{c,\alpha}$ denotes the fugacity of component c in phase α, which

is an effective partial pressure that replaces the mechanical partial pressure in an accurate computation of the chemical equilibrium constant. It is a function of pressure, temperature, and molar fractions of all components existing in a phase.

Several mathematical techniques handle the mass interchange in different phases of chemical components. The most common ones are based on (1) the K value approach, (2) an equation of state, and (3) a variety of empirical tables from experiments.

In a multi-component system, a K value (or an equilibrium ratio) is defined as the ratio of the mole fractions of a component in its distributed two phases:

$$K_{c,\alpha_1,\alpha_2} = \frac{x_{c,\alpha_1}}{x_{c,\alpha_2}}. \tag{2.18}$$

For ideal solutions, a K value is a function of only the system pressure and temperature, regardless of the overall composition of the mixture. It may be calculated from an analytic equation as:

$$K = \left(\frac{kv_1}{p} + kv_2 p + kv_3\right) \exp\left(\frac{kv_4}{T - kv_5}\right). \tag{2.19}$$

For our simulator, the K value approach is used. Due to the complexity of a non-isothermal process, the analytic equation is used for phase saturation constraint calculations. However, for a real solution, a K value is not only a function of pressure and temperature, but also depends on the composition of the system.

While the K value approach is easy to set up, it lacks generality and may result in inaccurate reservoir simulation. In recent years, an EOS has been more widely employed due to the

more consistent compositions, densities and molar volumes produced. The most famous EOS is the van der Waals EOS (Van der Waals, J. D. (1910)). Here we discuss three more accurate and reliable EOS: Peng–Robinson (PR), Redlich–Kwong (RK), and Redlich–Kwong–Soave (RKS) (Peng, D. Y. and Robinson, D. B. (1976); Redlich, O. and Kwong, J. N. (1949); Soave, G. (1972)). The main advantage of using an EOS is that the same equation can be used to model the behavior of all phases, thereby assuring consistency when performing phase equilibria calculations. Introduction of the compositions into the K value functions, however, adds considerable complexity to the flash computation (CMG (2015); Coats, K. H. (1976)).

2.1.3 Unknowns and equations

In our ISC simulator, $3N_c + 11$ unknowns need to be solved for. They consist of three pressures, saturations, velocities for each phase, and molar fraction in each phase of each component. Also, temperature and concentration of coke, which is the only component of the solid phase, are also involved. As for the equations, mass conservation, coke conservation, energy conservation and Darcy's law are included, which add up to $N_c + 5$ equations. The phase equilibrium constraints, phase composition constraints, phase saturation constraint and capillary pressure constraints add up to $2N_c + 6$ equations. Therefore, we have $3N_c + 11$ unknowns and $3N_c + 11$ equations; this problem can be solved. Initial conditions and closed boundary conditions are applied.

The table below lists the unknowns and equations.

Since the dimension of this problem is huge, a lot of computing resources would be

Unknowns	**Definition**	**Number**
p_w, p_o, p_g	Pressure of 3 phases	3
T	Temperature	1
S_w, S_o, S_g	Saturation of 3 phases	3
$\vec{u}_w, \vec{u}_o, \vec{u}_g$	Velocity of 3 phases	3
$x_{c,w}, \quad c = 1, 2, \cdots, N_c$	Mole fraction of single component in water phase	N_c
$x_{c,o}, \quad c = 1, 2, \cdots, N_c$	Mole fraction of single component in oil phase	N_c
$x_{c,g}, \quad c = 1, 2, \cdots, N_c$	Mole fraction of single component in gas phase	N_c
C_c	Coke concentration	1
Summation		$3N_c + 11$

Table 2.1: Unknowns of the mathematical model

consumed when running the simulator. To reduce the simulation time, some unknowns need to be eliminated in the problem.

Darcy's law is used to eliminate the velocities of the three phases; K value calculations are used to eliminate the molar fraction of each component in the water and gas phases, keeping the molar faction of each component in the oil phase; capillary pressure constraints are used to eliminate pressures in the water and gas phases, with the oil pressure kept; the phase saturation constraint is used to eliminate the oil saturation.

In our simulator, the components we consider can be divided into the water components, oil components, and gas components. The oil components can only exist in the oil phase and gas phase, and the water components can only exist in the water phase and gas phase. However, the gas components here denote non-condensate gas, which can only exist in the

Equations	**Index in the thesis**	**Number**
Mass conservation of N_c components	(2.3)	N_c
Energy conservation equation	(2.5)	1
Darcy's law	(2.1)	3
Phase equilibrium equations between phase o & w	(2.17)	N_c
Phase equilibrium equations between phase o & g	(2.17)	N_c
Coke conservation equation	(2.4)	1
Phase composition constraint for 3 phases	(2.15)	3
Capillary pressure constraint	(2.14)	2
Phase saturation constraint	(2.13)	1
Summation		$3N_c + 11$

Table 2.2: Equations of the mathematical model

Unknowns	**Index**	**Unknowns**	**Index**
p_o	0	Mass equation for comp 1	$1-1=0$
T	1	Mass equation for comp 2	$2-1=1$
S_w	2	$\cdots$	
S_g	3	Mass equation for comp i	$i-1$
$x_{1,o}$	$1+3=4$	$\cdots$	
$\cdots$		Mass equation for comp N_c	N_c-1
$x_{c,o}$	$c+3$	Energy equation	N_c
$\cdots$		Oil constraint equation	N_c+1
$x_{N_c-1,o}$	N_c+2	Gas constraint equation	N_c+2
C_c	N_c+3	Coke equation	N_c+3

Table 2.3: Unknowns and equations of In-Situ Combustion Simulator

gas phase. With the definition of the three kinds of components, there is no component which can exist both in the oil phase and water phase. Therefore, K value equations between the oil and water phases are no longer needed, meaning that the number of equations should decrease by N_c.

For the water components, usually we have only one water component:

$$x_{w,w} = 1,$$

which means that N_c unknowns are no longer needed. After the elimination, only $N_c + 3$ unknowns and equations are left, forming our final system of unknowns and equations.

In our simulator, one water component is mandatory. Also, coke is mandatory for the

combustion reaction. Hence, the breakdown of N_c is

$$N_c = n_{cw} + n_{co} + n_{cg} + n_{cc}, \tag{2.20}$$

We actually have

$$N_c = 2 + n_{co} + n_{cg}. \tag{2.21}$$

Here, $n_{c\alpha}$ denotes the number of the "default" components in phase α. For example, n_{cg} here denotes the number of non-condensate gas components. For all other components such as the water, light oil, and heavy oil components, they are not originally from the gas phase. In addition, a large fraction of each component is not in the gas phase under the standard conditions. The "gas components", however, will meet these two requirements. In the ISC model, a gas component mostly refers to a real component or a pseudo component such as inert gas, oxygen, carbon-monoxide, and carbon-dioxide.

For non-condensate gas components that do not dissolve into the oil phase, their molar fractions in the oil phase, x, will be zero. Therefore, x can no longer be kept as an unknown for them. The molar fraction of non-condensate gas in the gas phase will be the new unknowns for them.

For a clearer demonstration, different from the general model, the molar fraction of a component in the gas phase is denoted by y; in the oil phase, it is still denoted by x. The numbering of a component is also noted as a subscript.

For an ISC model, a well is a crucial part and well equations must be included. For each well, according to different well control conditions, well equations can be different. But

for one single well, only one unknown is required (bhp), and other parts of the equation are constants or functions of other unknowns. Therefore, with the well equations and bhp included, the system is still well-defined.

p_o can be seen as a basic pressure of a reservoir since the pressures of the gas phase and water phase can be calculated through these capillary pressure functions. Actually, the difference between the oil pressure and p_w, p_g is relatively small in some cases. So we denote p as p_o directly.

At this point, we can obtain a manipulated mathematical system which is shown in Table 2.4.

This is the final table of unknowns and equations. Specifically, the indices of the corresponding unknowns and equations are also displayed. In the programming of the simulator, these indices are used especially when building a linear system, including Jacobian matrices and right-hand side vectors in Newton iterations.

2.2 Properties and specific treatments

2.2.1 Reaction model

To illustrate our general chemical model with relative ease, we introduce a four-phase six-component model as an example. This model is originally from Crookston's model (Crookston, R. B., Culham, W. E., and Chen, W. H. (1979)). The six components involved in this model are Oxygen, Inert gas, Light oil, Heavy oil, water, and coke. There are four phases considered in our model. They are Gaseous (gas), Oleic (oil), Aqueous (water) and Solid

Index	**Unknowns**	**Equations**
0	p	Water mass equation
1	x_0	Oil mass equation for oil component 0
...	...	...
n_{co}	$x_{n_{co}-1}$	Oil mass equation for oil component $n_{co}-1$
$n_{co}+1$	y_0	Gas mass equation for gas component 0
...	...	...
$n_{co}+n_{cg}$	$y_{n_{cg}-1}$	Gas mass equation for gas component $n_{cg}-1$
$n_{co}+n_{cg}+1$	T	Energy equation
$n_{co}+n_{cg}+2$	S_w	Oil constraint equation
$n_{co}+n_{cg}+3$	S_g	Gas constraint equation
$n_{co}+n_{cg}+4$	C_c	Coke equation
$n_{co}+n_{cg}+5$	bhp	Well equation

Table 2.4: Modified mathematical system for ISC

Component		Phase (Mole Fraction)			
		Gaseous	**Oleic**	**Aqueous**	**Solid**
0	Water	$y_0 = K_0$		1	
1	Light Oil	$y_1 = K_1 x_0$	x_0		
2	Heavy Oil	$y_2 = K_2 x_1$	x_1		
3	Oxygen	y_3			
4	Inert Gas	y_4			
3	Coke				1

Table 2.5: Correlation of phase and components in a simple model

(coke).

For the gas components, only oxygen and inert gas are considered. The inert gas consists of all the non-condensate gases except the oxygen, including nitrogen, carbon monoxide, carbon dioxide and other oxides of carbon. Therefore, the inert gas component is actually a pseudo-component.

Water is assumed to be immiscible with oil. Vaporization and condensation of both water and oil are controlled by vapor-liquid equilibrium.

The situation of a component existing in different phases can be different on a case-by-case basis. This part is designed to be user input. The relationship between components and phases is shown in the following table.

For the oil component, we just make a simple breakdown of the light component and heavy component in this model. Besides the physical properties of the two components, the main difference is that there is a pyrolysis reaction for the heavy component, but not for

Index	Name	Chemical reaction equation	Reaction rate
1	Light Oil Oxidation	$LO + s_1O_2 \longrightarrow s_2CO_x + s_3H_2O$	r_1
2	Heavy Oil Oxidation	$HO + s_4O_2 \longrightarrow s_5CO_x + s_6H_2O$	r_2
3	Heavy Oil Cracking	$HO \longrightarrow s_7LO + s_8Coke + s_9CO_x$	r_3
4	Coke Oxidation	$Coke + s_{10}O_2 \longrightarrow s_{11}CO_x + s_{12}H_2O$	r_4

Table 2.6: Unknowns and equations of In-Situ Combustion Simulator

the light component. Besides oxidation reactions for the two components, there are three reactions for oil components.

As the main product of the pyrolysis reaction, there is an oxidation reaction for coke, which is the main fuel source of heat generation in this model. As shown in Table 2.6, only four reactions are included in this model.

According to the discussion, we know that the reaction sets are directly related to the division of components. These four reactions are only a chemical reaction model for this certain six-component four-phase model. If the user inputs a SARA model, with saturated, aromatic, resin and asphaltene as oil components, the reaction model will be totally different.

From these four reactions, a reactant and production, which are the basic components, can be known with certainty. Mass change of certain components is described by a kinetic reaction rate, which has a dimension of $MT^{-1}L^{-3}$. This rate is taken from the Arrhenius equation (Coats, K. H. (1980)), a formula describing the temperature dependence of reaction rates. Arrhenius' equation gives the dependence of the rate constant of a chemical reaction

on the absolute temperature, a pre-exponential factor and other constants of the reaction:

$$k(T) = Ae^{-(E_a/RT)} \tag{2.22}$$

In this equation, $k(T)$ is a reaction rate constant. It is actually a function of temperature, not a constant; the parameter of concentration is not included here. E is the activation energy of a reaction. The larger the activation energy, the slower the reaction. T is the reaction temperature. A is a pre-exponential factor, a constant for each chemical reaction that defines the rate due to the frequency of collisions in the correct orientation. In fact, this factor is a function of oil type, pressure, and other parameters.

At a low temperature, a combustion reaction is not likely to happen. Another explanation would be that oxidation is happening at a very low reaction rate, making the combustion reaction to appear not to happen. When the temperature is higher, the energy of a molecular system is easier to exceed the activation energy. Movement of particles becomes much more violent at higher temperature, resulting in more possibility of effective collisions between particles of reactants. Therefore, the reaction rate will be much higher.

Temperature plays an extremely significant role in chemical reaction rates, and an exponential function perfectly describes this behavior. A rate will be accelerated exponentially by raising the temperature of a system. Thus we have the following accurate expressions of

kinetic reaction rates:

$$
\begin{aligned}
r_1 &= A_1 e^{-\frac{E_{a,1}}{RT}}(y_3 p_g)(\phi S_o \rho_o x_0), \\
r_2 &= A_2 e^{-\frac{E_{a,2}}{RT}}(y_3 p_g)(\phi S_o \rho_o x_1), \\
r_3 &= A_3 e^{-\frac{E_{a,3}}{RT}}(\phi S_o \rho_o x_1)(1 - (\frac{C_c}{C_{c,max}})^5), \\
r_4 &= A_4 e^{-\frac{E_{a,4}}{RT}}(y_3 p_g)(C_c).
\end{aligned}
\tag{2.23}
$$

In these equations, we can see that a reaction rate depends on several physical properties of the reactants, including concentration and pressure. If a reaction occurs between the gas phase and the oil phase, the reaction rate is affected by the concentration of an oil component and partial pressure of a gas component. For a reaction occurring between the solid phase and the gas phase, the rate is proportional to the concentration of coke and partial pressure of the gas component, oxygen. Particularly, for the oil cracking reaction, the term $(1-(\frac{C_c}{C_{cmax}})^5)$ is added to control the production of coke. When the coke concentration is approaching the concentration limit of coke, this term will reduce the reaction rate, preventing the cracking reaction from producing more coke.

This model is only one fulfillment of the chemical reactions in which oxidation reactions of oil components in the gas phase are ignored, and oxidation reactions are not divided into the low-temperature and high-temperature reactions (LTO and HTO). The chemical reactions are open to users of this simulator and any scheme can be decided and input by the user, such as the SARA model (Belgrave, J. D. M. et al. (1993)).

2.2.2 Phase equilibrium

Due to the complexity of a non-isothermal model, phase equilibrium simulation is really time-consuming, so we use a K value approach to describe the phase behavior. Calculations of K-values are as follows (Chen, Z. (2007); CMG (2015)):

$$\begin{aligned} & K_W = K[0](p,T) \\ & = \left(\frac{kv1_W}{p} + kv2_W \cdot p + kv3_W\right) \exp\left(\frac{kv4_W}{T - kv5_W}\right), \\ & K_{O,i} = K[i+1](p,T) \\ & = \left(\frac{kv1_{O,i}}{p} + kv2_{O,i} \cdot p + kv3_{O,i}\right) \exp\left(\frac{kv4_{O,i}}{T - kv5_{O,i}}\right). \end{aligned} \tag{2.24}$$

However, the K value calculations above are correct when two phases co-exist: the water component exists in both the water phase and the gas phase or an oil component exists in both the oil phase and the gas phase. However, for an ISC process, temperature varies dramatically, leading to significant changes in phases.

For example, when a combustion front propagates, temperature rises rapidly. At first, water will evaporate due to its lower boiling point compared to the oil phase. With the higher temperature, the oil phase will also disappear. Since light components evaporate at high temperatures, heavy components will decrease to zero due to their participation in chemical reactions such as heavy oil cracking and heavy oil oxidation. That is to say, even if S_{or} is given in a relative permeability table when the saturation of the oil phase is low enough, mobilization is restricted and the oil phase will also disappear in the ISC process. Ideally, when the temperature is high enough, at positions near the combustion front, both

the water phase and the oil phase disappear, leading to only the gas phase and solid phase. The water component and oil components take a large portion of the gas phase. When the combustion process is over, the temperature will decrease. If water or oil components exist in the gas phase, these gasses will condense, forming the water phase and oil phase.

When such changes in phases occur, the K value formulas or the equation system will not be suitable anymore, resulting in a failure of the simulator. Therefore, proper handling of the appearance and disappearance of phases is significant. Typically, we have two approaches to deal with such a problem. First, the VS (Variable Substitution) method can be applied to the entire system, while the unknowns and equations need to be changed (Coats, K. H. (1980)). Judging the process of the method to be added to the simulator, it would result in complicated logic in the coding part and discontinuity of the system, leading to instability of the simulator.

As a result, we implement the PER (Pseudo-Equilibrium Ratios) method in our simulator (Crookston, R. B., Culham, W. E., and Chen, W. H. (1979); Abou-Kassem, J. H. and Aziz, K. (1985)). With this method, a correction term is added to the original K-value, and a pseudo K value is acquired:

$$K*_W = K^*[0](p,T) = \left(\frac{S_w}{S_w + n_{cg}}\right) K[0](p,T), \tag{2.25}$$

$$K*_{O,i} = K^*[i+1](p,T) = \left(\frac{S_o}{S_o + \epsilon}\right) K[i+1](p,T). \tag{2.26}$$

Therefore, the gas phase molar fraction for the oil components and water component are functions of p, T, S_w, S_g. The molar fraction in the gas phase for gas components are the

basic unknowns:

$$y = y(p, T, S_w, S_g). \tag{2.27}$$

In calculations of pseudo K-values, ϵ is a small number of the order of $1e-4$. When the water or oil phase exists, $S_w \gg \epsilon$, and the term $\left(\frac{S_w}{S_w+\epsilon}\right) \approx 1$. Thus the pseudo K-values are extremely close to the K-values. As the liquid phase gradually disappears, the saturation decreases and approaches ϵ, and the term $\left(\frac{S_w}{S_w+\epsilon}\right)$ approaches 0.5. When the saturation keeps decreasing, the term $\left(\frac{S_w}{S_w+\epsilon}\right)$ approaches 0.

For a typical K-value, when the liquid phase disappears, the K value approach cannot be used anymore. We take pure water as an example. When there is only a water phase, we heat the system and the temperature keeps rising. Below the boiling point, the K value is always less than 1. When the boiling point at a certain pressure is achieved, the K value equals 1. This can be used in the calculated saturated steam temperature. When the temperature of the mixture increases, the water phase disappears, leaving only the gas phase. The K value under this condition is greater than 1.

Obviously, if the water phase and gas phase co-exist, the molar fraction of the water component in the water phase and gas phase should be 1, leading to a K value of 1. In other words, a K value is only validated when it equals 1; i.e., two phases coexist. If the liquid phase and gas phase do not co-exist, the formula of a K value has no meaning. This is because the K value is defined by formula $K = \frac{y}{x}$, and the liquid phase and gas phase must appear.

In the ISC process, the disappearance of the liquid phase can be described by the PER

method. With an increase in the K value as temperature rises, the y values can be greater than 1, meaning that the molar fraction of a certain component is greater than 1. Obviously, this is wrong. With the correction term, however, pseudo K-values can be less than K-values. Therefore, although phase equilibrium no longer applies, with the pseudo K-values, the relationship $y = K^*x$ still applies.

When saturation approaches 0, both the pseudo K-values and fraction y of a certain component approach 0. This process can be described by the formulas perfectly. With the formulas, we can get the saturation by solving the equation system at each time step. The unknowns and saturation can be calculated to a non-zero value.

Although its saturation should be zero when the liquid phase disappears, it seems that the PER method introduces an obvious error deviation in this case. However, the correction term will be effective only when the liquid phase begins to disappear. At that time, the calculated saturation will be close to ϵ or even smaller. So, when we take $\epsilon = 1e-4$, the absolute error of saturation is less than $1e-4$, which is acceptable, especially when the great convenience the method brings is considered.

Besides the above error, the characteristic of the correction term can lead to a problem. As shown in Figure 2.1, when S approaches 0, the value and derivative of the function $\left(\frac{s}{s+\epsilon}\right)$ change dramatically. The behavior of the function is similar to a discontinuous function, which will lead to convergence failure in Newton iterations. Thus the pseudo K-values are extremely sensitive to saturation, and very fine Newton step sizes on phase saturation are required. Also, in Newton iterations, when S is not at the order of ϵ, the tangent slope is near zero. This causes the saturation easily converging to a negative value, which

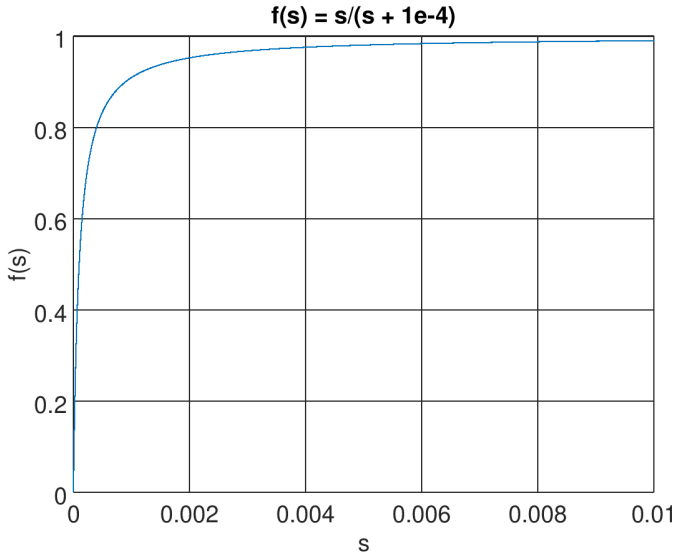

Figure 2.1: Schematic of correction term in PER method

is not acceptable. Therefore, a special treatment of saturation in each Newton step is needed(Oklany, J. S. F. (1992)):

$$S_o^{k+1} = \sqrt{\epsilon \cdot S_o^k}. \tag{2.28}$$

$$S_w^{k+1} = \sqrt{\epsilon \cdot S_w^k}. \tag{2.29}$$

or

$$S_o^{k+1} = 0.5 S_o^k. \tag{2.30}$$

$$S_w^{k+1} = 0.5 S_w^k. \tag{2.31}$$

Here, the superscript k or $k+1$ denotes the k-th or (k+1)-th Newton step. If saturation is directly set to a small value, a Newton iteration fails easily.

During the implementation, there is a modification to the PER method. Crookston, R.

B., Culham, W. E., and Chen, W. H. (1979) set a model for only two components in oil phase. A pseudo K value is only applied to heavy oil component. Oklany, J. S. F. (1992) applied pseudo K values to all components in his simulator. In our simulator, pseudo K values for the oil phase is only applied to one component, which is the heaviest component. There are two reasons for this. First of all, during the physical process, lighter components will disappear first when the oil phase is heated. This usually leaves the heaviest component to evaporate last. Molar fractions are evaluated:

$$y_{LO} = K_{LO}(p,T)x_{LO}, \tag{2.32}$$

$$y_{HO} = \left(\frac{S_o}{S_o+\epsilon}\right) K_{HO}(p,T)x_{HO}. \tag{2.33}$$

For a light component, its K value will increase earlier than a heavy component, and its molar fraction in the oil phase x will decrease earlier. After evaporation of the light component, the molar fraction in the oil phase of the heavy component will approach 1. Thus a small value for y of the light component results from a small x value. As for the heavy component, a small value for y results from a small S_o value. This realization is close to the physical process.

On the other hand, this implementation deals with potential problems in numerical solution. When the oil phase disappears ($S_o \longrightarrow 0$), the mass conservation equation of an oil component is changed to $y^{n+1} = 0$, where $n+1$ denotes the (n+1)-th time step. According

to Equation 2.32 and 2.33, the equation system becomes

$$\begin{aligned} x_{LO} &\simeq 0, \\ S_o &\simeq 0. \end{aligned} \tag{2.34}$$

However, if the correction term is added to each component, every single mass conservation equation for an oil component will be

$$S_o x_{O,i} \simeq 0. \tag{2.35}$$

Since we know that $S_o \simeq 0$, the value of $x_{oil[i]}$ cannot be determined by the system. In the solution process, it will result in linear dependence of the linear system as well as a singular Jacobian system. An accurate solution to the linear system cannot be determined. Without this modified PER method, the problem remains soluble when a restriction is applied to oil phase. When the oil phase almost disappears, both convection term and reaction term should be set to zero, which prevent the oil phase decrease to zero. The oil phase will remain an extremely small number, and the system will remain non-singular. However, this method includes new discontinuity in the whole problem. The failures in Newton iteration are more common. As a result, this modified PER method in multicomponent system showed its effectiveness in solution process.

2.2.3 Properties of fluid

Compressibility factor of real gas

Since an EOS equation is not introduced in our model for the calculation of phase equi-

librium, a compressibility factor of gas is not acquired by this EOS approach. Another approach of this physical variable is implemented.

In our model, the Redlich-Kwong EOS (Redlich, O. and Kwong, J. N. (1949)) is used to calculate the Z factor.

$$A = A(p,T) = 0.427480 \left(\frac{p}{p_{crit}}\right) \left(\frac{T_{crit}}{T}\right)^{2.5}, \tag{2.36}$$

$$B = B(p,T) = 0.086640 \left(\frac{p}{p_{crit}}\right) \left(\frac{T_{crit}}{T}\right). \tag{2.37}$$

IN addition, the following mixing method is applied:

$$a = \sum_i y_i T_{crit,i} \sqrt{\frac{T_{crit,i}}{p_{crit,i}}}, \tag{2.38}$$

$$b = \sum_i y_i \frac{T_{crit,i}}{p_{crit,i}}, \tag{2.39}$$

$$T_{crit} = \left(\frac{a^2}{b}\right)^{\frac{2}{3}}, \tag{2.40}$$

$$p_{crit} = \frac{T_{crit}}{b}. \tag{2.41}$$

Then, after we have the coefficients A and B, the compressibility factor of real gas satisfies the equation

$$Z^3 - Z^2 + (A - B - B^2)Z - AB = 0. \tag{2.42}$$

This equation is cubic. Therefore, there are three roots for the equation. Also, a root might be virtual. In this case, we choose the biggest real root. With the calculation of all the

coefficients, the Z factor is a function of p, T, x_i and y_i:

$$Z = Z(p, T, x_i, y_i). \tag{2.43}$$

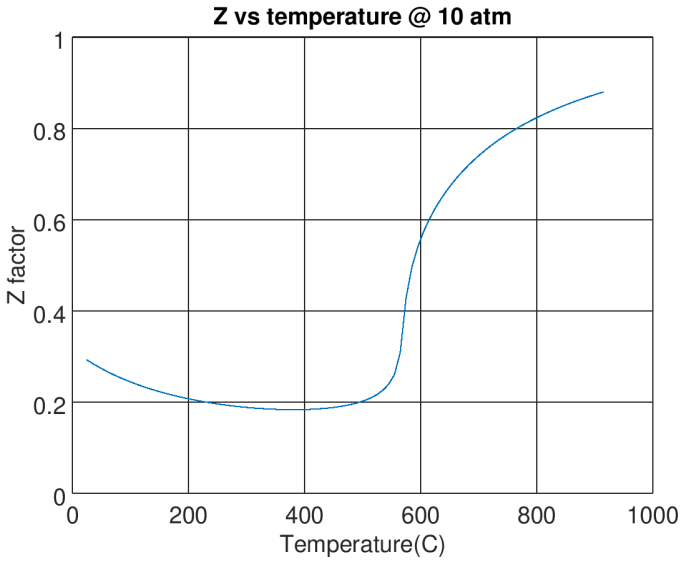

Figure 2.2: Z factor of air vs temperature (79% N_2, 21% O_2)

Critical temperatures and pressures for different components are different. This difference cannot be neglected. In the ISC process, temperature varies greatly, as well as the molar fraction of each component in the gas phase and Z factor. As shown in Figure 2.2, at the pressure of 10 atm, the Z factor of air changes a lot with temperature. Obviously, the value is different from 1. To get the accurate result, the real gas equation must apply.

2.2.3.1 Density

We introduce the molar density: the molar amount of certain material in unit volume in the simulator. This usage ensures the unit of a mass conservation equation to be [mole/t]. The calculation here are consistent with CMG (2015).

For different phases, the calculation of densities varies a lot. We must consider them separately.

First of all, for the gas phase, in 1834 Émile Clapeyron first stated the ideal gas law. Since a volume taken by the gas phase is mainly affected by molecular collisions, the volume taken by a certain amount of gas material is irrelevant to the type of the material itself. With the ideal gas law, under a condition of certain pressure and temperature, the molar density of all kinds of gas remains the same. (Clapeyron, É. (1834))

However, the ideal gas law is only an approximation to the real case. Though different EOS equations are all approximations to the real case, they perform better than the ideal gas model when comparing to results from the laboratory. For EOS equations, a compressibility factor Z is introduced as a correction term for the real gas model, while in the ideal gas model we have $Z = 1$. According to the discussion above, there is a significant difference between the compressibility factor and 1. Not only pressure and temperature also different composition of the gas phase will lead to a significant difference in the Z factor. For different components, the critical pressure and critical temperature may differ, resulting in a significant difference in gas behavior.

With a real gas equation considered, the density of the gas phase can be calculated as:

$$\rho_g = \rho_g(p, T, x_i, y_i) = \frac{p}{Z(p, T, x_i, y_i) \cdot R \cdot T}$$

The water phase only contains one water component in this model, so the calculation of

the water density is simple:

$$\rho_w = \rho_w(p,T) = \rho_{w,ref}\exp(cp_w(p-p_{ref}) - ct1_w(T-T_{ref}) \tag{2.44}$$

$$-\frac{ct2_w}{2}(T-T_{ref})^2 + cpt_w(p-p_{ref})(T-T_{ref})) \tag{2.45}$$

where $\rho_{w,ref}$ is the reference density of the water phase at the reference temperature and pressure.

For the oil phase, if only component $O[i]$ is in the oil phase, the density can be calculated similarly:

$$\rho_{O[i]} = \rho_{O[i]}(p,T) = \rho_{O[i],ref}\exp(cp_{O[i]}(p-p_{ref}) - ct1_{O[i]}(T-T_{ref}) \tag{2.46}$$

$$-\frac{ct2_{O[i]}}{2}(T-T_{ref})^2 + cpt_{O[i]}(p-p_{ref})(T-T_{ref})) \tag{2.47}$$

However, under most circumstances, more than one component exist in the oil phase. The density of the oil phase should be considered with molar fractions of oil components. If there is one gmol of material in the oil phase, then for component $O[i]$, there will be x_i gmol of it. Moreover, for one gmol of component $O[i]$ in the oil phase, it will take a volume of $\frac{1}{\rho_{O[i]}}$. Therefore, for unit gmol of material in the oil phase, it will take the volume of:

$$\frac{1}{\rho_o} = \sum_i^{n_{co}} \frac{1}{\rho_{O[i]}} x_i. \tag{2.48}$$

Therefore, using a mixing principle, the harmonic average is introduced for the oil phase:

$$\rho_o = \rho_o(p, T, x_i) = 1/\sum_i^{n_{co}} \frac{x_i}{\rho_{O[i]}(p, T)}. \tag{2.49}$$

When the water phase contains multiple components, this equation is also valid for this phase.

2.2.3.2 Viscosity

The viscosity of heavy oil is very high, making it very hard to mobilize. With this characteristic, thermal recovery is introduced. The mechanism is that, when temperature is higher, the viscosity of heavy oil will decrease dramatically. For calculation, correlation methods are adopted here (CMG (2015)).

For the oil phase, if only component $O[i]$ is in the oil phase, it will have a viscosity:

$$\mu_{O[i]} = avisc_{O[i]} \exp\left(\frac{bvisc_{O[i]}}{T}\right). \tag{2.50}$$

This viscosity can be obtained from a viscosity table as well. As a mixture of different components, the oil phase viscosity is obtained by a logarithmic mixing rule:

$$\ln(\mu_o) = \sum_i^{n_{co}} x[i] \ln(\mu_{O[i]}(T)). \tag{2.51}$$

Therefore, we see that

$$\mu_o = \mu_o(T, x_i) = \exp\left(\sum_i^{n_{co}} x[i] \ln(\mu_{O[i]}(T))\right) = \sum_i^{n_{co}} \left(\mu_{O[i]}(T)\right)^{x_i}. \tag{2.52}$$

For the water phase, since it has only one component, the viscosity can be derived directly:

$$\mu_w = \mu_w(T) = avisc_w \exp\left(\frac{bvisc_w}{T}\right). \tag{2.53}$$

For the gas phase, however, the viscosity is usually much smaller than the oil viscosity or the water viscosity, and the effect of temperature on its viscosity is not as important as that for the oil phase or the water phase. Thus, another formula for the viscosity calculation is introduced:

$$\mu_{g,c} = \mu_{g,c}(T) = avg_c \cdot T^{bvg_c}. \tag{2.54}$$

In this formula, the viscosity will not decline so fast with an increasing temperature. But under standard conditions, the viscosity of water is around 1 cp, while the viscosity of air is about 0.01 cp, much smaller than that of water. Therefore, the flow of gas will dominate.

According to a mixing rule, the molar mass of a component is included:

$$\mu_g = \mu_g(p, T, S_w, S_g, x_i, y_i) = \frac{\sum_c \mu_{g,c} \cdot y_c \sqrt{M_c}}{\sum_c y_c \sqrt{M_c}}. \tag{2.55}$$

2.2.4 Properties relative to rocks

2.2.4.1 Porosity

Porosity is the ratio of the pore volume to the bulk volume in a porous medium, describing the volume containing fluids. When pressure is high, due to the effort of fluids, pores are also enlarged. For a non-isothermal model, the porosity is also influenced by temperature.

We define a coefficient as a total compressibility of porosity (Chen, Z. (2007)):

$$ctot = ctot(p,T) = cpor(p - p_{ref}) - ctpor(T - T_{ref}) + cptpor(p - p_{ref})(T - T_{ref}). \quad (2.56)$$

This factor is a function of pressure and temperature. For the calculation of porosity, we have two approaches with this factor:

Linear:

$$\phi = \phi(p,T) = \phi_{ref} \cdot (1 + ctot(p,T)). \quad (2.57)$$

Nonlinear:

$$\phi = \phi(p,T) = \phi_{ref} \cdot e^{ctot(p,T)}. \quad (2.58)$$

For both two approaches, porosity is a function of pressure and temperature.

Especially, for the ISC process, we consider the effect of coke. In Coats' simulator, when coke is generated, the solid phase appears. The saturations of all four phases should be

considered. (Coats, K. H. (1980)) The phase constraint becomes

$$S_w + S_o + S_g + S_s = 1. \tag{2.59}$$

In other words, when the solid phase appears, it will take up a part of the pore volume, but porosity will not be affected. Nevertheless, we will take the same treatment as in CMG's STAR. When the solid phase appears, it is immobile and thus takes a volume of the fluid space. In this case, coke has a similar effect as the matrix. This process can also be treated as porosity declining. With the formula

$$\phi_f = \phi - \phi_{coke}, \tag{2.60}$$

the porosity for fluids can be calculated by the original porosity and porosity is altered by coke. The concentration of coke is an unknown, which is defined by the amount of material (gmol) in unit volume (m^3). So the porosity taken by coke ϕ_{coke} can be calculated by

$$\phi_{coke} = C_c/\rho_{coke} \tag{2.61}$$

In the solid phase, the volume can hardly be affected by pressure or temperature. Thus, the molar density of coke can be taken as a constant. As a result, with the volume of coke being considered, porosity is a function of pressure, temperature and coke concentration:

$$\phi = \phi(p, T, C_c) \tag{2.62}$$

With this treatment of coke, our ISC simulator can also handle the process like in-situ coal gasification problems.

2.2.4.2 Relative permeabilities

Just like any model containing three phases, relative permeabilities have the similar expression. Although a solid phase of coke is involved, this phase is not movable.

For the relative permeability of water k_{rw}, the relative permeability in a three-phase system equals the relative permeability in a water-oil system. Thus, from the SWT table (a water-oil relative permeability table), k_{rw} can be obtained with interpolation. k_{rw} is a function only of S_w:

$$k_{rw} = k_{rw}(S_w). \tag{2.63}$$

In addition, the relative permeability of gas k_{rg} in a three-phase system or an oil-gas system can be calculated in the same manner. Through interpolation with a SLT table (a liquid-gas relative permeability table), k_{rg} can be calculated. Also, k_{rg} is a function only of S_g:

$$k_{rg} = k_{rg}(S_g). \tag{2.64}$$

As for the relative permeability of oil k_{ro}, there are several models available (Corey, A. T. et al. (1956); Naar, J. and Wygal, R. J. (1961), 1961; Stone, H. L. (1970); Delshad, M. and Pope, G. A. (1989)). In this simulator, we only include Stone's model II (Stone, H.

L. (1973)):

$$k_{ro} = k_{ro}(S_w, S_g) \tag{2.65}$$

$$= k_{rocw}\left[\left(\frac{k_{row}(S_w)}{k_{rocw}} + k_{rw}(S_w)\right)\left(\frac{k_{rog}(S_g)}{k_{rocw}} + k_{rg}(S_g)\right) - k_{rw}(S_w) - k_{rg}(S_g)\right]. \tag{2.66}$$

where k_{rocw} is the oil-water two-phase relative permeability to oil at connate water saturation, $krog$ is the oil-gas two-phase relative permeability to oil, and $krow$ is the oil-water two-phase relative permeability to oil.

$$k_{rocw} = k_{row}(S_w = S_{wc}) = k_{rog}(S_g = 0). \tag{2.67}$$

$krow$ and $krog$ are interpolated from a SWT table and a SLT table separately.

However, for our ISC simulator, temperature will change during the entire process. The temperature will affect the properties of non-isothermal flow such as relative permeabilities. During implementation, the Stone II model needs to be adjusted. However, due to the complexity of non-isothermal flow, the effect is not considered in our simulator at present. Also, in the upstream direction of a combustion front, the temperature is very high, which has a very big effect on relative permeabilities; while only the gas phase is present, the effect on total flow is not that important. Furthermore, in the downstream direction of a combustion zone, the temperature is not that high, and thus the effect of temperature is not necessarily involved.

2.2.5 Non-isothermal description

Enthalpy is a measurement of energy in a thermodynamic system. It is the thermodynamic quantity equivalent to the total heat content of a system. It is equal to the internal energy of the system plus the product of pressure and volume. For the gas phase, if only component c exists, the gas enthalpy is calculated as follows (CMG (2015)):

$$H_{g,c} = H_{g,c}(T) = \int_{T_{ref}}^{T} \big(cpg1_c + cpg2_c \cdot t + cpg3_c \cdot t^2 + cpg4_c \cdot t^3\big) dt, \tag{2.68}$$

$cpgi_c$, $i = 1, 2, 3, 4$, are constants for component c. Here, the component c can refer to not only a gas component, but also an oil component or a water component. All the components must be able to exist in the gas phase, and the enthalpy of a component in the gas phase can generally be calculated by this formula.

Since the gas phase is a mixture, the gas enthalpy can be calculated by a weighted mean with gas molar fractions y_c:

$$H_g = H_g(p, T, S_w, S_g, x_i, y_i) = \sum_{c}^{N_c} y_c H_{g,c}. \tag{2.69}$$

For the oil and water phases, the heat of vaporization should be considered. From the gas phase to a liquid phase, steam will condense, and part of the energy is released as heat. This part of the heat is measured by the heat of vaporization, which is the amount of energy (enthalpy) that must be added to a liquid substance to transform a unit amount of that

substance into a gas. It can be calculated by:

$$H_{v,c} = H_{v,c}(T) = u(T_{crit,c} - T) \cdot hvr_c \cdot (T_{crit,c} - T)^{ev_c}. \tag{2.70}$$

Here $u(x)$ denotes the Heaviside unit step function. The component's enthalpy will decrease from the gas phase to a liquid phase when the temperature is lower than the critical temperature. This heat is also influenced by temperature.

Therefore, for a certain component c, its enthalpy in a liquid phase can be calculated as:

$$H_c = H_c(T) = H_{g,c} - H_{v,c}. \tag{2.71}$$

where $H_{g,c}$ is the enthalpy of component c in the gas phase, which can be calculated using the formula above.

As a result, for the water phase which only includes one component, the enthalpy is:

$$H_w = H_w(T) = H_{g,W} - H_{v,W}. \tag{2.72}$$

For the oil phase, as a mixture, the enthalpy is:

$$H_o = H_o(p, T, x_i) = \sum_{i}^{n_{co}} x_i (H_{g,O[i]} - H_{v,O[i]}). \tag{2.73}$$

Now, we have enthalpy for all three fluid phases, which are used to measure energy in heat convection. Including the mechanical work, it is the energy carried by the material of a phase.

As heat accumulates, the internal energy must be used, because the pushing work is not necessarily included. For a liquid phase, its volume is set and cannot be changed much, leading to zero pushing work. Thus, the pushing work will only have a significant effect on the gas phase. Hence we have the formulas for three fluid phases (CMG (2015)):

$$U_w = U_w(T) = H_w - p/\rho_w, \tag{2.74}$$

$$U_o = U_o(p, T, x_i) = H_o - p/\rho_o, \tag{2.75}$$

$$U_g = U_g(p, T, S_w, S_g, x_i, y_i) = H_g - p/\rho_g. \tag{2.76}$$

For our simulator, since the solid phase and rock have no mobility, the related properties are not included, such as density. Both have internal energy, and for heat conduction, this part of the energy will also play an important role in the whole process.

For rock, a similar formula is used:

$$U_r = U_r(T) = cp1_r(T - T_{ref}) + \frac{cp2_r}{2}(T^2 - T_{ref}^2). \tag{2.77}$$

One thing to notice is that the internal energy for rock has a unit of energy per unit volume, while others have energy per unit amount of material.

As for the solid phase, if coke has a heat capacity, which is the only component in the solid phase, then the enthalpy can be calculated as:

$$U_c = U_c(T) = cp_c(T - T_{ref}). \tag{2.78}$$

2.2.6 Well terms

A Peaceman's model is adopted for well modeling. A well may have many perforations. For each perforation at a grid cell, its well rate for phase α, $Q_\alpha = Vq_\alpha$, is calculated by the following formula (Peaceman, D. W. (1978)):

$$Q_{\alpha,well} = WI\frac{\rho_\alpha k_{r\alpha}}{\mu_\alpha}\left(bhp - p_\alpha - \gamma_\alpha g(z_{bh} - z)\right), \tag{2.79}$$

where WI is the well index. For a vertical well, it can be defined as:

$$WI = \frac{2\pi h_3\sqrt{k_{11}k_{22}}}{\ln(\frac{r_e}{r_w}) + s}. \tag{2.80}$$

A well index defines the relationship among a well bottom hole pressure, a flow rate and a grid block pressure. bhp is the bottom hole pressure defined at the reference depth z, z_bh is the depth of the perforation in grid cell, and p_α is the phase pressure in grid block m.

From the mass conservation equation of component c, the well term can be obtained:

$$Q_{c,well} = Vq_{c,well} = \sum_{\alpha}^{N_\alpha} Q_{\alpha,well}\cdot x_{c,\alpha}. \tag{2.81}$$

From the energy conservation equation, the well term can be obtained:

$$Q_{well} = \sum_{\alpha}^{N_\alpha} Q_{\alpha,well}\cdot H_\alpha. \tag{2.82}$$

If a well is added to the reservoir model, the bottom hole pressure of a well becomes an unknown. Due to various operation constraints, a fixed bottom hole pressure, a fixed oil

rate, a fixed water rate or a fixed liquid rate may be applied to a well at different time stages. When the fixed bottom hole pressure condition is applied to a well, the well equation is

$$bhp = c. \tag{2.83}$$

Here c is a constant pressure.

The fixed rate operation for phase α is described by the following equation:

$$\sum_{perf} \left(Q_{\alpha,well}\right)_{perf} = c, \tag{2.84}$$

where c is a constant rate and known.

The total fixed rate operation is described by the following equation:

$$\sum_{perf} \left(Q_{w,well} + Q_{o,well} + Q_{g,well}\right)_{perf} = c. \tag{2.85}$$

Different from other equations, the well equations are handled by the last processor when running a parallel simulation. As discussed before, a given grid will be divided into different subdomains, stored on different nodes. All the cells and related properties are stored together. Thus, for one single processor, all the equations in cells in this part are handled by this processor.

In one scenario, cells with perforations of a certain well should not be divided into different parts. Otherwise, for a certain processor, when assembling well equations, communications between different processors are required, which will increase the overhead of the total assembling time and solution time. However, the loading balance solution we use makes it

kind of difficult to consider the integrity of perforation cells of one well. On the other side, comparing both effectiveness and quality of a loading balance method, the overhead resulting from well perforation cells is acceptable.

Therefore, to have the best performance and quality of domain partitioning, wells are not considered in separate processors. All the well equations will be handled only by the last processor. With message passing from/to other processors, the well equations can be assembled.

Chapter 3

Numerical Methods

The partial differential equations and constraints are discretized based on structured or unstructured grids. The main purposes of a numerical discretization method include providing geometrical information and computing residuals and derivatives on each element or control volume. Detailed information will be introduced in the following sections.

3.1 Discretization scheme

Typically, different time discretization (time marching) methods can be applied. Explicit (or forward difference, forward Euler) and implicit (or backward difference, backward Euler) schemes are the two most popular methods and are both first-order accurate in time. The Crank-Nicolson scheme is second-order accurate.

An explicit scheme is the simplest method in which all fluxes and sources are evaluated by using their previous time step values. It is stable provided that a time step is sufficiently small (conditionally stable). Therefore, if a time step is large, a stable constraint must be

determined to restrict and modify the time step. Under a given set of conditions, a time step cannot exceed this constraint, which will prevent the time step from increasing and the total number of iterations from deceasing. Therefore, for an explicit scheme, its performance may not be very good with a large number of time iterations.

The Crank-Nicolson scheme is second-order accurate and is particularly useful when time accuracy is of importance. Although this scheme is more accurate, it requires almost the same computational effort as other implicit schemes. This scheme is quite popular since it is unconditionally stable, but oscillatory solutions are possible when using large time steps. For this reason, it is not suitable for our simulator either.

For an implicit scheme, its fluxes and sources are evaluated in terms of the unknown variable values at the new time step. So both the current state of a system and the later one are involved. Although an implicit scheme requires more memory (a linear system of algebraic equations needs to be stored and solved), the use of the implicit Euler scheme allows for large time steps to be taken, increasing the overall performance of a simulator. However, a time step may be limited due to the severely nonlinear nature of reservoir dynamics created by multi-phase, multi-component mass and heat flow with chemical reactions through heterogeneous porous media.

As a result, the time discretization is carried out with an implicit scheme in our simulator. As for spatial discretization, typically, the Finite Difference Method (FDM), Finite Element Method (FEM), and Finite Volume Method (FVM) can be applied in the ISC process. In our approach, we adopt the FDM due to its good performance with simplicity of parallelization and development.

Before discretization, let us illustrate some concepts and quantities. First, we have the following relationship in time steps:

$$t^{n+1} = t^n + \Delta t^n.$$

Here, superscript n denotes the previous time step and $n+1$ the current time.

Second, recalling the mass conservation equations (2.3) and energy conservation equation (2.5), we introduce the potential for each fluid phase, defined as

$$\Phi_w = p_w - \gamma_w z$$

$$\Phi_o = p_o - \gamma_o z$$

$$\Phi_g = p_g - \gamma_g z$$

Therefore, Darcy's flow equations can be rewritten in the form of

$$\vec{u}_w = -k\frac{k_{rw}}{\mu_w}\nabla\Phi_w,$$
$$\vec{u}_o = -k\frac{k_{ro}}{\mu_o}\nabla\Phi_o,$$
$$\vec{u}_g = -k\frac{k_{rg}}{\mu_g}\nabla\Phi_g.$$

Third, we introduce the difference sign Δ and take a potential as an example:

$$\Delta\Phi_{\alpha,face} = (\Phi_\alpha)_c - (\Phi_\alpha)_{face}.$$

Here, a face is a direction in space. Typically, it is defined by a Cartesian coordinate system.

Along with the x, y, and z directions and three opposite directions, we have six directions defined. Each direction is denoted by an integer from 0 to 5. A face can denote the meaning of front, back, left, right, up or down. c here denotes a center cell. Thus, a face here actually refers to the neighboring cell in the direction of the "face".

Then we have the discretization of the governing equations. Here, we omit the superscript $n+1$, and n is preserved. Also, if not noted by a "face", all properties and equations are regarding the center cell c. Otherwise, they are regarding the neighboring cell in the "face" direction.

First, we obtain mass conservation equations in new forms. For the water component W:

$$\begin{aligned} F_W = &- \sum_{face=0}^{5} \left(\frac{Ak}{h}\right)_{avg} \left(\frac{k_{rw}\rho_w}{\mu_w}\right)_{avg} \Delta\Phi_{w,face} \\ &+ \frac{V}{\Delta t}\left[\phi\rho_w S_w - (\phi\rho_w S_w)^n\right] \\ &- \sum_{face=0}^{5} \left(\frac{Ak}{h}\right)_{avg} \left(\frac{k_{rg}\rho_g}{\mu_g} y_W\right)_{avg} \Delta\Phi_{g,face} \\ &+ \frac{V}{\Delta t}\left[\phi\rho_g S_g y_W - (\phi\rho_g S_g y_W)^n\right] \\ &- V q_{W,well} - V q_{W,reac}. \end{aligned}$$

Here, V denotes the volume of a center cell, A denotes a cross-sectional area, and h denotes the length of an edge of the cell. The water phase and the gas phase have mass transfer so these two phases need to be involved in this equation.

Similarly, for the oil component $O[i]$:

$$\begin{aligned} F_{O[i]} = &- \sum_{face=0}^{5} \left(\frac{Ak}{h}\right)_{avg} \left(\frac{k_{ro}\rho_o}{\mu_o} x_i\right)_{avg} \Delta\Phi_{o,face} \\ &+ \frac{V}{\Delta t}\left[\phi\rho_o S_o x_i - (\phi\rho_o S_o x_i)^n\right] \\ &- \sum_{face=0}^{5} \left(\frac{Ak}{h}\right)_{avg} \left(\frac{k_{rg}\rho_g}{\mu_g} y_{O[i]}\right)_{avg} \Delta\Phi_{g,face} \\ &+ \frac{V}{\Delta t}\left[\phi\rho_g S_g y_{O[i]} - (\phi\rho_g S_g y_{O[i]})^n\right] \\ &- V q_{O[i],well} - V q_{O[i],reac}. \end{aligned}$$

But for the gas component $G[i]$, it only exists in the gas phase so that

$$\begin{aligned} F_{G[i]} = &- \sum_{face=0}^{5} \left(\frac{Ak}{h}\right)_{avg} \left(\frac{k_{rg}\rho_g}{\mu_g} y_{G[i]}\right)_{avg} \Delta\Phi_{g,face} \\ &+ \frac{V}{\Delta t}\left[\phi\rho_g S_g y_{G[i]} - (\phi\rho_g S_g y_{G[i]})^n\right] \\ &- V q_{G[i],well} - V q_{G[i],reac}. \end{aligned}$$

For the coke equation, we have:

$$F_{Coke} = \frac{V}{\Delta t}\left[C_c - (C_c)^n\right] - V q_{c,reac}.$$

Similarly, for the energy conservation equation, we have the discrete form:

$$
\begin{aligned}
F_E = &- \sum_{face=0}^{5} \left(\frac{Ak}{h}\right)_{avg} \left(\frac{k_{rw}\rho_w}{\mu_w} H_w\right)_{avg} \Delta\Phi_{w,face} \\
&- \sum_{face=0}^{5} \left(\frac{Ak}{h}\right)_{avg} \left(\frac{k_{ro}\rho_o}{\mu_o} H_o\right)_{avg} \Delta\Phi_{o,face} - \sum_{face=0}^{5} \left(\frac{Ak}{h}\right)_{avg} \left(\frac{k_{rg}\rho_g}{\mu_g} H_g\right)_{avg} \Delta\Phi_{g,face} \\
&+ \frac{V}{\Delta t}\left[\phi\rho_w S_w U_w - (\phi\rho_w S_w U_w)^n\right] + \frac{V}{\Delta t}\left[\phi\rho_o S_o U_o - (\phi\rho_o S_o U_o)^n\right] + \frac{V}{\Delta t}\left[\phi\rho_g S_g U_g - (\phi\rho_g S_g U_g)^n\right] \\
&+ \frac{V}{\Delta t}\left[(1-\phi)U_r - ((1-\phi)U_r)^n\right] + \frac{V}{\Delta t}\left[C_c U_c - (C_c U_c)^n\right] \\
&- \sum_{face=0}^{5} \left(\frac{A}{h}\right)_{avg} (K_T)_{avg} \Delta T_{face} \\
&- VQ_{well} - VQ_{reac} + VQ_{loss}.
\end{aligned}
$$

For the gas and oil composition constraints, we also have the discrete form:

$$
\begin{aligned}
F_{GS} &= \sum_{c} y_c - 1, \\
F_{OS} &= \sum_{i}^{nco} x_i - 1.
\end{aligned}
$$

Therefore, all the discretization is finished for all cell equations.

The properties of a grid, rock, and fluids, such as thickness, permeability, and porosity, can significantly differ from one block to another. The properties are often given only at the centers of cells, but the transmissibilities are computed at the boundaries of cells. For transmissibilities at grid block boundaries, proper average techniques between two adjacent blocks should be taken into consideration.

For transmissibility in a "face" direction, a different method of calculating averages should

be applied. For the first term, grid and rock properties are depicted, and a harmonic average should be used:

$$\left(\frac{Ak}{h}\right)_{avg} = \frac{2\left(\frac{Ak}{h}\right)_c\left(\frac{Ak}{h}\right)_{face}}{\left(\frac{Ak}{h}\right)_c + \left(\frac{Ak}{h}\right)_{face}}$$

The fluid properties are described in the second term in transmissibility. Especially for multi-phase flow, upstream weighting should be applied to calculate rock/fluid properties. This prevents a physically incorrect solution. With a fluid potential defined,

$$\left(\frac{k_{r\alpha}\rho_\alpha}{\mu_\alpha}\right)_{avg} = \begin{cases} \left(\frac{k_{r\alpha}\rho_\alpha}{\mu_\alpha}\right)_c, & \text{if}(\Phi_\alpha)_c > (\Phi_\alpha)_{face} \\ \left(\frac{k_{r\alpha}\rho_\alpha}{\mu_\alpha}\right)_{face}, & \text{if}(\Phi_\alpha)_c < (\Phi_\alpha)_{face} \end{cases}$$

If $(\Phi_\alpha)_c > (\Phi_\alpha)_{face}$, we call cell c the upstream cell, and the cell face is the downstream block. We should take the properties of the upstream cell to be the term at the boundary (Ertekin, T., Abou-Kassem, J. H., and King, G. R. (2001); Chen, Z. (2007)).

In the in-situ combustion process, the gas phase will not disappear for non-condensable gas injected and produced during chemical reactions. Such gases include oxygen, carbon oxides, and nitrogen. A saturated state for a thermal model always exists. With the correction terms of K-values, the liquid phases also exist. Therefore, the structure of this system does not change during calculations.

3.2 Linearization of nonlinear system

For one cell, there are several equations and unknowns in this cell. If we denote

$$\vec{F}_c = \left(F_W, F_{O[0]}, \cdots, F_{O[nco-1]}, F_{G[0]}, \cdots, F_{G[ncg-1]}, F_Coke, F_E, F_{OC}, F_{GC}\right)^T,$$

$$\vec{u}_c = \left(p, T, S_w, S_g, x_0, \cdots, x_{nco-1}, y_{G[0]}, \cdots, y_{G[ncg-1]}, C_c\right)^T,$$

combining these vectors cell by cell and adding well equations and bottom hole pressure, we can get a global equation vector $\vec{F}$ and a global unknown vector $vecu$. With these two vectors, we can get the nonlinear system at time step $n+1$:

$$F^{n+1}(u^{n+1}) = 0.$$

In other words, at each time step $n+1$, we need to solve this system to find a u^{n+1}. Most of the time, this system is nonlinear. So, to solve this system, we must linearize it. Usually, iteration methods can be applied to deal with this problem. We can choose the Newton method or the Inexact Newton Method.

Typically, if there is an equation

$$f(x) = 0,$$

where f is a nonlinear function, a root of this equation x_R needs to be obtained. Then we consider the Taylor expansion at $x = x*$:

$$f(x) = f'(x_R)(x - x_R) + o(x - x_R).$$

If the high-order infinitesimal is omitted, we obtain

$$x* \approx x - f(x)/f'(x_R).$$

If there is an initial value of x_0, since $f'(x_R)$ is unknown, $f'(x_0)$ is taken instead. Then we

can get

$$x^1 = x^0 - f(x^0)/f'(x^0).$$

Here, x_1 is closer to $x*$ than x_0. If we repeat this procedure:

$$x^{k+1} = x^k - f(x^k)/f'(x^k),$$

and if function f satisfies certain properties, x^{k+1} will approach x_R. This method is called the Newton method, which is shown in Figure 3.1.

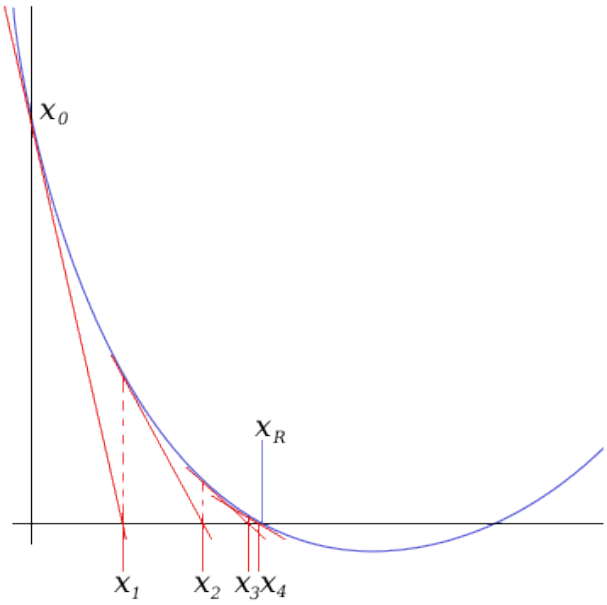

Figure 3.1: Schematic of the Newton method

The Newton method can be extended to multivariate functions. We use the Taylor expansion

$$\vec{f}(\vec{x} + \delta\vec{x}) = \vec{f}(\vec{x}) + J(\vec{x})\delta\vec{x} + \delta\vec{x}^T H(\vec{x})\delta\vec{x} + \cdots .$$

Here, $J(\vec{x})$ is the Jacobian matrix and $H(\vec{x})$ is the Hessen matrix. With this expansion, the

Newton method can be modified to:

$$J(\vec{x}^k)\delta\vec{x}^{k+1} = -f(\vec{x}^k),$$

$$\delta\vec{x}^k = \vec{x}^{k+1} - \vec{x}^k.$$

Therefore, calculations of the Jacobian matrix are crucial in the Newton method to solve a nonlinear system.

$$J = \frac{\partial \vec{f}}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial \vec{f_1}}{\partial \vec{x_1}} & \cdots & \frac{\partial \vec{f_1}}{\partial \vec{x_n}} \\ \cdots & \cdots & \cdots \\ \frac{\partial \vec{f_n}}{\partial \vec{x_1}} & \cdots & \frac{\partial \vec{f_n}}{\partial \vec{x_n}} \end{bmatrix}.$$

However, linear systems from reservoir simulations are usually hard to solve, and the time spent on their solution may take up more than 70% of the total simulation time. To achieve high accuracy, the stopping tolerance of the standard Newton method is usually set to be low, exacerbating time consumption. Actually, in the first few Newton iterations, a solution may be far from the real solution. Therefore, it is not necessary to solve the linear systems very accurately, and the Inexact Newton Method is considered in the simulator (Dembo, R. S., Eisenstat, S. C., and Steihaug, T. (1982); Toselli, A. and Widlund, O. (2006); Chen, Z., Huan, G., and Ma, Y. (2006); Chen, T. et al. (2009); Zhong, H. et al. (2016)).

The inexact Newton algorithm is shown in Algorithm 1.

In the Inexact Newton method, solving the linear system $J\vec{u} = -\vec{F}$ accurately is replaced by finding an approximate solution $\vec{u}^*$ of the linear system.

Algorithm 1 Inexact Newton method

Set $\vec{u}^{l=0} = \vec{u}^n$, $J^{l=0} = \frac{\partial \vec{F}^n}{\partial \vec{u}^n}$, $\vec{F}^{l=0} = \vec{F}^n$, $l = 0$
while Newton iteration step $l < l_{max}$ **do**
Determine η_l
Find a $\delta\vec{u}^l$ such that

$$||J^l \delta\vec{u}^l + \vec{F}^l||_2 \leq \eta_l ||\vec{F}^l||_2$$

Update the unknown vector

$$\vec{u}^{l+1} = \vec{u}^l + \delta\vec{u}^l$$

Update the residual vector

$$\vec{F}^{l+1} = \vec{F}(\vec{u}^{l+1})$$

Update the Jacobian matrix

$$J^{l+1} = \frac{\partial \vec{F}^{l+1}}{\partial \vec{u}^{l+1}}$$

if $||\vec{F}^{l+1}||_2 \leq \epsilon$ **then**
Break
end if
Set $l = l + 1$
end while

In the inequality above, the forcing term η_l forces the residual to be low enough, thus assuring fast local convergence and avoiding over-solving of the linear system. One can choose the forcing term in the following ways:

$$\eta_l = \begin{cases} \frac{||\vec{F}(\vec{u}^l) - \vec{r}^{l-1}||_2}{||\vec{F}(\vec{u}^{l-1})||_2}, \\ \frac{||\vec{F}(\vec{u}^l)||_2 - ||\vec{r}^{l-1}||_2}{||\vec{F}(\vec{u}^{l-1})||_2}, \\ \gamma \left(\frac{||\vec{F}(\vec{u}^l)||_2}{||\vec{F}(\vec{u}^{l-1})||_2} \right)^\beta. \end{cases}$$

where R is the residual of the l-th iteration $r^l = \vec{F}(\vec{u}^{l-1}) - J(\vec{u}^{l-1})\vec{u}^l$. And β, γ are constants, and $\eta_l \in [0.01, 0, 1]$.

3.3 Jacobian matrix

As discussed in the last subsection, the calculation of a Jacobian matrix is crucial in the Newton method. For the calculation of a matrix, we have two choices: analytical and numerical.

For the analytical choice, a matrix consists of partial derivatives, and it can be obtained through the chain rule. The chain rule is a formula for computing the derivative of the composition of two or more functions.

For example, for functions f and g, if we define the function F by $F = f(g(x))$, then the derivative of F is

$$F'(x) = f'(g(x)) \cdot g'(x).$$

Also, for a multivariate function $\vec{f} = \vec{f}(\vec{x})$,

$$\frac{\partial \vec{f}}{\partial x_i} = \sum_{l=1}^{m} \frac{\partial \vec{f}}{\partial u_l} \frac{\partial \vec{u_l}}{\partial x_i}.$$

Then, with the chain rule, we can calculate a Jacobian matrix one entry by one entry to obtain its analytical Jacobian matrix.

Also, there is a choice of a numerical Jacobian matrix. If a partial derivative is not easily obtained, a difference can be used to approach a derivative. For the definition of a partial derivative:

$$\frac{\partial f_i}{\partial x_j} = \lim_{\Delta x_j \to 0} \frac{f_i(x_0, \cdots, x_j + \Delta x_j, \cdots, x_n) - f_i(x_0, \cdots, x_j, \cdots, x_n)}{\Delta x_j},$$

when Δx_j is small enough, we will have

$$\frac{\partial f_i}{\partial x_j} \approx \frac{f_i(x_0, \cdots, x_j + \Delta x_j, \cdots, x_n) - f_i(x_0, \cdots, x_j, \cdots, x_n)}{\Delta x_j}.$$

With this formula, as a set of partial derivatives, a Jacobian matrix can be calculated approximately.

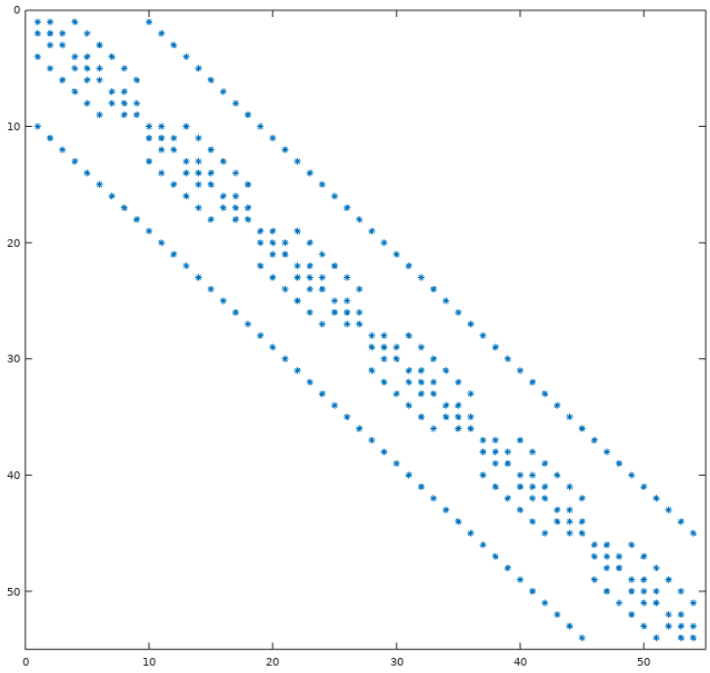

Figure 3.2: Pattern of Jacobian matrix

As shown in Figure 3.2, the Jacobian matrix has a pattern of the 7-block-diagonal matrix. The grid has a structure of $3 \times 4 \times 5$ cells. Here the well equation part is omitted in this figure. In the figure, each asterisk sign denotes a small matrix. This small matrix has a size of a number of cell equations. For a small matrix, the condition number can be very big for an ISC simulator. Also, if there is a large number of grid cells in the model, the condition number can be very big for a large scale.

For the condition number of matrix A, which is not singular, it has a definition:

$$cond(A) = ||A^{-1}|| \cdot ||A||.$$

Here $||A||$ denotes a norm of matrix A. And it can be any norm of a matrix, such as 1-norm, 2-norm or ∞-norm. Also, for the definition, we have

$$cond(A) \geq ||A^{-1} \cdot A|| = 1.$$

Therefore, the best case for a matrix is that its condition number equals 1.

We consider a linear system $A\vec{x} = \vec{b}$, and hypothesize that there is a disturbance $\delta\vec{b}$, and the solution to the disturbed system is $\vec{x} + \delta\vec{x}$, which satisfies $A(\vec{x} + \delta\vec{x}) = \vec{b} + \delta\vec{b}$. Then there is an estimation that

$$\frac{||\delta\vec{x}||}{||\vec{x}||} \leq cond(A)\frac{||\delta\vec{b}||}{||\vec{b}||} \tag{3.1}$$

Moreover, if there is a disturbance δA, the solution to the disturbed system is $\vec{x} + \delta\vec{x}$, satisfying $(A + \delta A)(\vec{x} + \delta\vec{x}) = \vec{b} + \delta\vec{b}$. Then there is an estimation that

$$\frac{||\delta\vec{x}||}{||\vec{x}||} \leq \frac{cond(A)\frac{||\delta A||}{||A||}}{1 - cond(A)\frac{||\delta A||}{||A||}}. \tag{3.2}$$

When $||\delta A||$ is small enough, the right-hand side approximately equals $cond(A)\frac{||\delta A||}{||A||}$ (Belsley, D. A., Kuh, E., and Welsch, R. E. (2005)).

As a result, according to Eq. (3.1) and Eq. (3.2), if there is a disturbance of the system

with a big condition number, even if a solution is obtained, this solution can be extremely inaccurate.

Also, if the condition number is big for a matrix when an iterative method is applied, the convergence speed can be very slow with this ill-conditioned problem. If the nature of the matrix is better, the performance of the Krylov subspace method will typically be better. We take GMRES as an example, if the matrix is not too far from normality, fast convergence will occur. On the other hand, fast convergence will occur when all eigenvalues of a matrix are clustered away from the origin. But if the condition number is big, the matrix is similar to a singular matrix and there will be at least one eigenvalue close to the origin. Therefore, a large condition number prevents a Krylov subspace method from fast convergence.

3.4 Decoupling and solution

Typically, a big condition number results from two aspects: coupling in a cell matrix and a large scale of grid cells.

First, for the matrix in a single cell, when the number of components is large, the coupling phenomenon can be extremely obvious, making its condition number extremely big. If the condition number gets bigger, it indicates that the matrix is closer to a singular matrix, which makes a linear system very hard to solve.

According to the nature of a system of the equations, there are three main reasons for a large condition number for a small matrix. First, for the in-situ combustion method, the number of components of a single model can be very large. However, the mass conservation equations and energy conservation equation are similar. If the number of similar equations is

large in the system, the linear correlation of rows in this matrix can be very strong. We know that for an orthogonal matrix, one row is orthogonal to another one, the linear correlation of rows is the least in this case. Also, the condition number of this matrix is one, which is the best case.

Second, for the adoption of pseudo K-values as discussed in the sub-section of K-values, the derivatives of these pseudo K-values with respect to saturation can be extremely large when the saturation of water or oil is very small. As shown in Figure 2.1, the tangent at $s = 0$ is nearly parallel to the y-axis. For a correlation term, if we take function the

$$f(s) = \frac{s}{s + \epsilon},$$

then

$$f'(s) = \frac{\epsilon}{(\epsilon + s)^2}.$$

Therefore,

$$f'(0) = \frac{1}{\epsilon},$$

indicating that when the oil phase or water phase is about to disappear, the entries in the Jacobian matrices $\frac{\partial F_i}{\partial S_w}$ and $\frac{\partial F_i}{\partial S_g}$ can be extremely big compared to other entries. This phenomenon also makes a matrix ill-conditioned. In the implementation, we set the crk_mod to be not too small. According to the literature and our tests, a number around 1e-4 would lead to a good balance between the condition number and relative error of mass conservation. However, with an application of the PER method, the matrix is still not well-conditioned. In this case, a preconditioner is required.

Third, the primary variable selection and variable alignment are considered in many reservoir simulators with the aim to make the nature of a matrix better. One example is to make a matrix diagonally dominant or positive definite. But in a process of ISC, changes among phases are violent. The correspondence between equations and unknowns set at the beginning may preserve the good nature of a matrix, but it may not be suitable for a new state of equations as time goes. Therefore, a good arrangement may be damaged. Under this circumstance, the matrix may not be so well-conditioned.

Therefore, we adopt a decoupling method, especially for a small matrix. For full decoupling, a Schur complement can be used. In practical problems, however, this is impossible to implement. Many approximate decoupling methods are proposed, such as the quasi-IMPES method and classical IMPES method. These decoupling methods can remove the pressure-saturation coupling in a diagonal block in a Jacobian matrix (Gries, S. (2015); Jiang, Y. (2008); Lacroix, S., Vassilevski, Y. V., and Wheeler, M. F. (2000)). Furthermore, some methods do not only focus on the pressure-saturation coupling but also some general inter-equation couplings of a linear system, such as the ABF (Alternate Block Factorization) method and local QR decomposition for decoupling (Klie, H., Rame, M., and Wheeler, M. (1996); Bank, R. E. et al. (1989); Lacroix, S., Vassilevski, Y. V., and Wheeler, M. F. (2000)).

The coupling problem is caused not only by the pressure-saturation coupling but also caused by multi-components. Therefore, a general inter-equation couplings exists in a system, which ABF can handle.

The ABF method is achieved by scaling a Jacobian matrix J such that its diagonal blocks

become the unit block. For the linear system $J\delta\vec{u} = -\vec{f}$, $J = (J_{i,j})_{ncell\times ncell}$, where $J_{i,j}$ is the main cell matrix of the first cell. Then we define

$$C_L = \begin{bmatrix} (J_{1,1})^{-1} & & \\ & \ddots & \\ & & (J_{ncell,ncell})^{-1} \end{bmatrix}.$$

A new linear system can be obtained:

$$C_L J\delta\vec{u} = -C_L\vec{f}.$$

With this decoupling, the diagonal blocks of this Jacobian matrix will change to the unit block.

When implementing the ABF decoupling method, the Gauss-Jordan transformation is employed. Different form multiplied by an inverse matrix directly, each diagonal block in the Jacobian matrix is transformed in less steps than traditional ABF method. The complexity of computations can be reduced from $ncell \cdot O(nequ^3)$ to $ncell \cdot O(nequ^2)$.

In the process of Gauss-Jordan transformation, three types of elementary row operations are employed to achieve this goal. The main difference between this method and ABF is that multiplications between blocks are avoided. This method is called the Gauss-Jordan elimination through pivoting. The detailed algorithm of the transformation is listed as follows:

Algorithm 2 Gauss-Jordan transformation

Initialize a flag array d with zero.
for $c = 0 : ncell - 1$ **do**
  for $k = 0 : ncell - 1$ **do**
    $a_{max} = 0$
    for $i = c : c + nequ - 1$ **do**
      if $d[i] == 1$ **then**
        continue
      end if
      for $j = c : c + nequ - 1$ **do**
        if $d[j] == 1$ **then**
          continue
        end if
        if $|a_{ij}| > |a_{max}|$ **then**
          Set $r = i$, $l = j$
          $a_{max} = a_{ij}$
        end if
      end for
    end for
    if $r \neq l$ **then**
      Exchange row $A_{r,:}$ and $A_{l,:}$
      Exchange entry b_r and b_l
    end if
    for $j = c : c + nequ - 1$ **do**
      $a_{lj} = a_{lj}/a_{max}$
    end for
    for $i = c : c + nequ - 1$ **do**
      if $i == l$ **then**
        continue
      end if
      for $j = 0 : ncell \cdot nequ$ **do**
        $a_{ij} = a_{ij} - a_{il}a_{lj}$
        $b_i = b_i - a_{il}b_l$
      end for
    end for
    $d[i] == 1$
  end for
end for

With this decoupling method, the condition number of a diagonal block will reduce to one directly. In testing, without the Gauss-Jordan transformation, the simulator cannot execute successfully. The main problem occurs in the solution of the linear system. The iterative solution can hardly reach a convergence without the proper decoupling method. Another common decoupling method, DRS, also failed in the simulator. ABF succeed in the execution, while the performance is still not as good as Gauss-Jordan transformation. These performances indicate the effectiveness of this novel decoupling method.

On the other hand, if the condition number of a diagonal block is small enough and the scale of the problem is enormous, the problem will still be ill-conditioned. Under this circumstance, Krylov subspace methods are adopted to solve this problem (Van der Vorst, H. A. (2003); Greenbaum, A. (1997); Meurant, G. (1999); Hackbusch, W. (1994); Elman, H. C., Silvester, D. J., and Wathen, A. J. (2014); Liesen, J. and Strakos, Z. (2013); Olshanskii, M. A. and Tyrtshnikov, E. E. (2014)). For the effectiveness and reliability of these methods, preconditioning must be adopted. Trefethen and Bau III once said in their book, "Nothing will be more central to computational science in the next century than the art of transforming a problem that appears intractable into another whose solution can be approximated rapidly. For Krylov subspace matrix iterations, this is preconditioning" (Trefethen, L. N. and Bau III, D. (1997)).

Many general preconditioners are proposed: ILU(k), ILUT, domain decomposition and AMG (algebraic multi-grid). In addition, special preconditioners including the classical CPR (pressure constrained residual) preconditioner and other CPR-like preconditioners are contained (Wallis, J. R., Kendall, R. P., and Little, T. E. (1985); Cai, X. C. and Sarkis,

M. (1999); Falgout, R. D. and Yang, U. M. (2002); Cao, H. et al. (2005); Cao, H. (2002); Liu, H., Wang, K., and Chen, Z. (2016)).

Also, in our simulator, commonly used Krylov subspace solvers are implemented, including the restarted GMRES solver and BiCGSTAB solver (Saad, Y. and Schultz, M. H. (1986); Saad, Y. (2003); Van der Vorst, H. A. (1992); Sleijpen, G. L. and Fokkema, D. R. (1993)).

AMG solvers are also an effective iterative solver. But only when a matrix is positive definite, the solution process will be efficient. For Krylov subspace solvers, these solvers are suitable for arbitrary linear systems. Linear systems from the ISC simulator are neither symmetric nor positive definite, so the Krylov subspace solvers are more suitable for our simulator. (Huang, C. K. (2009))

3.5 Overall structure

For a simulator, simulation results are required in a period of $[t_{start}, t_{end}]$ so a PDE system can be solved numerically with a sequence of time steps:

$$t_{start} = t^0 < t^1 < \cdots < t^n < t^{n+1} < \cdots = t_{end}.$$

At the beginning, an initial condition is needed, which can be used at the first time step. When the first step is calculated, the time step size is directly given by user setting. But for the following steps, time steps can be adjusted to minimize the total number of time steps, and, therefore, decrease the total iterations for a solution. However, if the time step Δt^n is too big, there will be a big truncation error, which is the error of a mathematical model from

simple linear behavior in time. Therefore, the time steps should be controlled and adjusted at every time step.

There are some desired changes of unknowns according to user-specified parameters. The user can set max changes for some variables, which are regarded as the best case to minimize a truncation error. The aim of a time step selection is to try to approach the max changes of the desired changes of unknowns at each time step. Therefore, the change in each unknowns should be noted. Also, the size from the previous time step should be included as well (Grabowski, J. W. et al. (1979); Sammon, P. H. and Rubin, B. (1986); Todd, M. R., O'dell, P. M., and Hirasaki, G. J. (1972); Chappelear, J. E. and Rogers, W. L. (1974); Jensen, O. K. (1980); Mehra, R., Hadjitofi, M., and Donnelly, J. K. (1982)).

We introduce a time step size ratio ϑ:

$$\vartheta = \frac{\Delta t^{n+1}}{\Delta t^n} = \min_{u_i} \min_{cell} \left(\frac{(1+\eta)\Delta u_i}{\delta u_i^n + \eta \Delta u_i} \right)_{cell}.$$

Here, $(u_i)cell$ is the i-th unknown in a cell, and Δu_i is the desired change of $(u_i)cell$. η is a tuning factor, which is between 0 and 1.

Then, at each time step, a nonlinear system will be solved with the Newton method. If the Newton method fails, it indicates that the convergence condition does not meet within the number of iterations. Then the time step will be cut to a smaller one, and the iteration will be redone. If the iteration remains a failure until a limit of time step cuts (NCUTS) is reached, the simulator will stop with an error. However, if convergence is reached during iteration, the unknowns $\vec{u}$ will be updated.

The Newton iteration will continue for the next time step. The flow chart of the simulator

is illustrated in Figure 3.3.

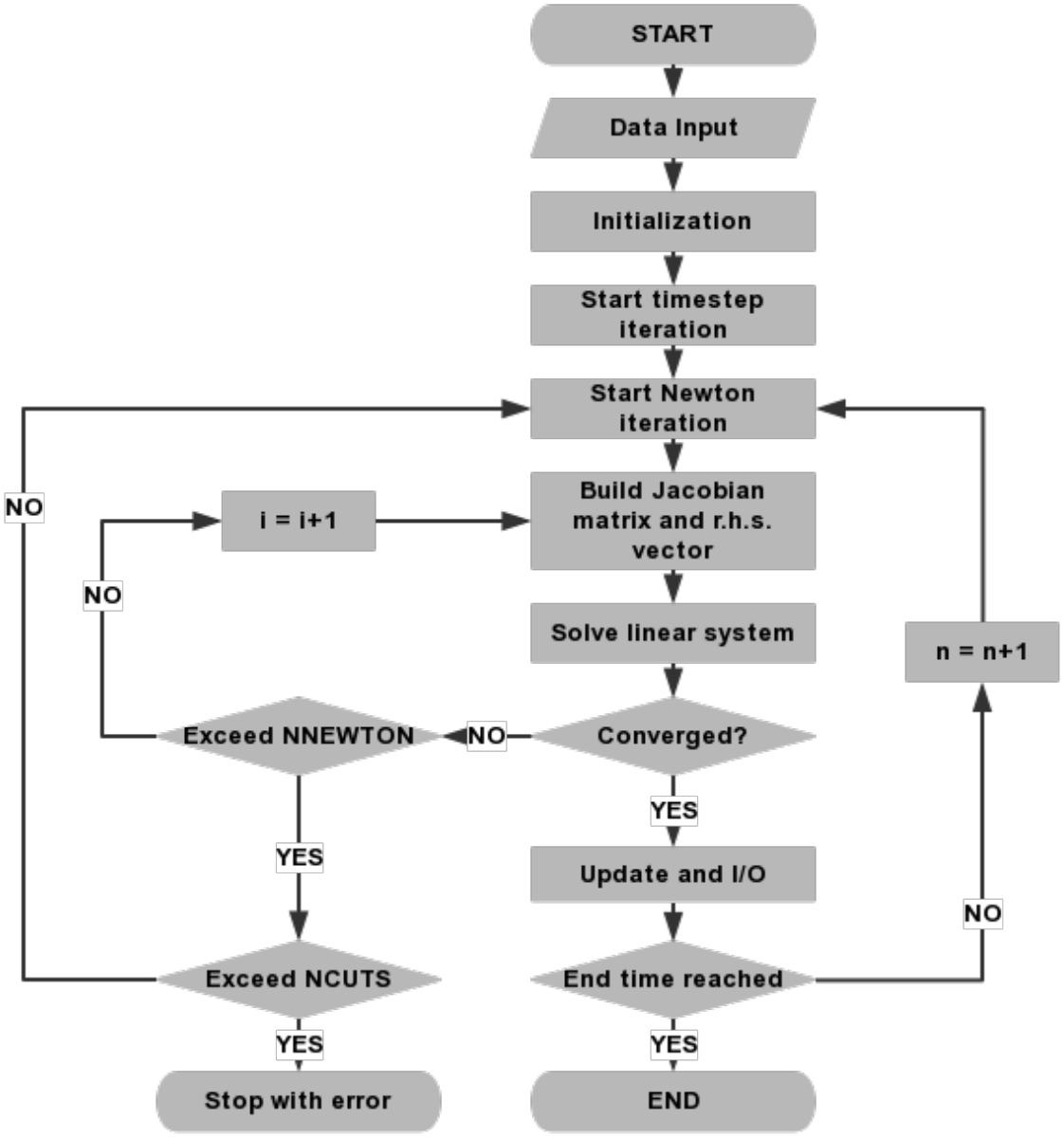

Figure 3.3: Flow Chart of In-Situ Combustion simulator.

Chapter 4

Parallel Computing

4.1 Why parallelization?

Parallel computing is a type of computation in which multiple computing units work simultaneously on one single computation problem. For a serial computing, one computing unit can solve one problem solely (as shown in Figure 4.1), but if a problem is computationally intensive, its computation will be extremely time-consuming.

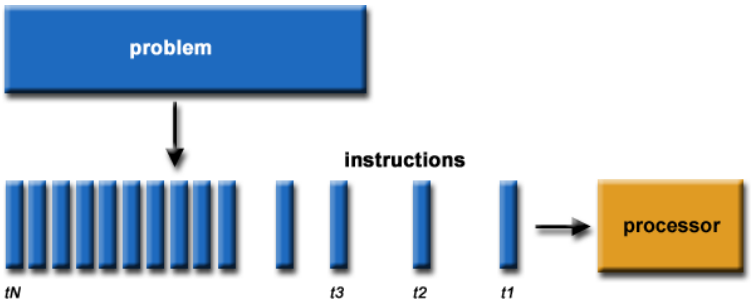

Figure 4.1: Schematic of serial computing (Barney, B. (2007)).

Traditionally, programs have been written for serial computing. For hardware, most of these programs will be run on a single computer with one single Central Processing Unit

(CPU). Nowadays, computers with multi-processors are common in our daily life, from PCs to clusters. With all processor units working together, computationally intensive engineering problems can be solved by multiple computing units so that the solution time will be reduced significantly. Decades ago, parallel computing has been considered as "the high end of computing" and has been adopted for various scientific and engineering problems by scientists and engineers.

As shown in Figure 4.2, a large-scale problem can be divided into small parts that can be solved concurrently, and each part will be assigned to one processor. Each part can be further divided into a series of instructions, which will simultaneously execute on different processors.

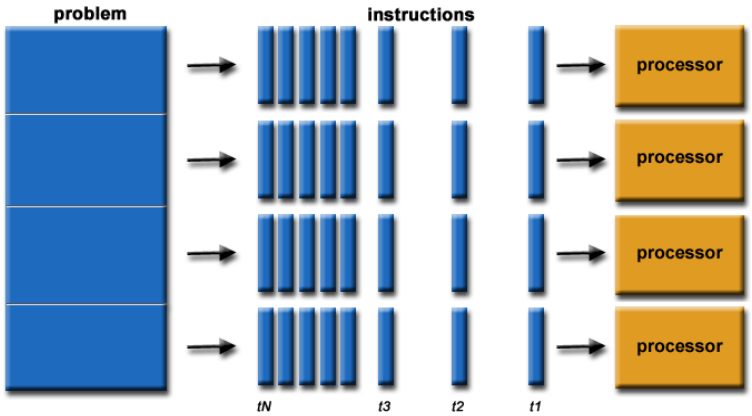

Figure 4.2: Schematic of parallel computing (Barney, B. (2007)).

There are three main advantages with parallel computing. First of all, the solution time of problems is decreased significantly. When more computing resources are used, more computing power will be obtained, shortening the solution time. Here, the computing resources refer to both numbers of multiple processors/cores in a single computer or the number of such computers connected by a network.

Besides, the capability of computing will be improved. This refers to a distributed-

memory parallel system, especially for clusters. Multiple memories are combined, thus large-scale problems with a large amount of data can be stored separately, and the limitation of one computer with one memory can be overcome.

Furthermore, a low price and performance ratio can be achieved. For an advanced single processor computer, its cost will increase even faster than increasing computing power due to physical limits, such as the effectiveness of heat dissipation. So parallel computing is economical. For the ISC process, the rapid-changing temperature and a thin moving combustion front make a time step small.

With a thin combustion front, extremely fine grids are applied. Therefore, comparing to other reservoir simulators, a problem to be solved in an in-situ combustion simulator will be more complex. Thus, running a simulation is even more time-consuming. To deal with this problem, a dynamic gridding technique is used to separately deal with the region of a combustion front and other regions (Sammon, P. H. (2003); CMG (2015)). Another approach is to adopt a new upscaling method with non-Arrhenius kinetics (Zhu, Z. (2011a)). But with parallel computing, methods to simplify a model are not necessary. A computationally intensive reservoir model can be simulated in the acceptable time.

From the discussion above, parallel computing is extremely beneficial for an in-situ combustion model, but is an ISC simulator suitable for implementation of parallelization? The answer is yes.

For the process of parallel computing, a computational problem can be broken down into small discrete pieces. These pieces must be solved simultaneously. For a reservoir simulation model, all the properties, unknowns, and equations are based on a grid. For a whole grid, it

can be partitioned into different sub-domains.

Besides, the time spent can be reduced if a problem is solved with multiple compute resources than with a single compute resource, as mentioned above. As a matter of fact, 70–90% of the computational time is spent on the assembly and solution of linear systems of algebraic equations for any simulator. In addition, for Krylov subspace solvers, a parallel algorithm can be applied. Also, assembling of a system can be finished by a processor separately. Therefore, the computational time of any main part can be shortened by the cooperation of processors.

With parallelization, the performance of a simulator can be highly-improved. Parallel simulators have been developed and documented in the literature (Wheeler and Smith, 1989; Rutledge, J. M. et al. (1992); Killough, J. E. and Bhogeswara, R. (1991); Kaarstad, T. et al. (1995); Anguille, L. et al. (1995); Parashar, M. et al. (1997); Killough, J. E., Camilleri, D., et al. (1997)).

Also, in our Reservoir Simulation Group, a lot of work related to parallel simulators has been done (Liu, H., Wang, K., Chen, Z., Jordan, K. E., et al. (2015); Liu, H., Wang, K., Chen, Z., Yang, B., et al. (2016); Wang, K., Liu, H., and Chen, Z. (2015); Liu, H., Wang, K., Chen, Z., Luo, J., et al. (2015); Zhong, H. et al. (2016)).

4.2 Parallel computers

For parallel systems, supercomputers or high-performance clusters are a good choice. Compared to personal computers, supercomputers have strong computing power. Taking a laptop with an Intel Core i5 CPU as an example, its speed is 43.4467 GFLOPS. But for

Sunway TaihuLight in China, the fastest supercomputer in the world, its speed is over 125 PFLOPS. It is faster than the Intel Core i5 CPU by over 3,000,000 times, meaning that if a simulation is run for more than one month on PCs, it will only take one second on this supercomputer. Therefore, with supercomputers, a simulator will be much more effective.

Also, clusters are systems in which multiple stand-alone computers (nodes) are connected with a network. As shown in Figure 4.3, every node is connected by an interconnection network. Inside a single node, processors are connected by a crossbar. For each node, data is all stored in the memory of the node. But if data stored in another node is required, communication between different nodes is needed through an interconnection network.

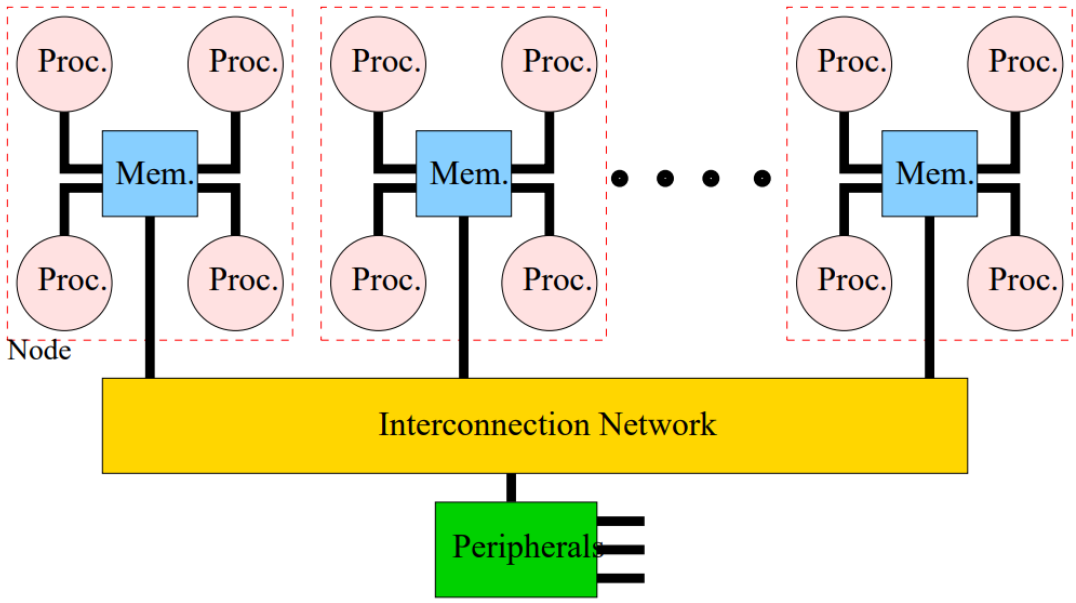

Figure 4.3: Block diagram of a cluster system with a "hybrid" network (Van der Steen, A. J. and Dongarra, J. J. (2012)).

Clusters have a hybrid shared-distributed memory. Typically, there are three types of architecture for parallel computers, including shared memory, distributed memory, and hybrid shared-distributed memory.

For a shared memory system (Figure 4.4), all processors can get access to the global address of the memory. Therefore, the memory is shared by all processors. If there is a large number of processors, with a limited bandwidth, there may be a traffic restriction on a program. Besides, cache/memory management can be a problem for a developer. Moreover, the changes in a memory location are affected by all processors, and changes are visible to all other processors. Therefore, the problem of synchronization remains to be dealt with by developers. For hardware, its scalability is limited to the number of processors.

For a distributed memory system (Figure 4.5), each processor has an independent memory, which other processors do not have the permission to get access to. The address is only available to its own processor. With this, overhead spent on maintaining global cache coherency is prevented. Additionally, save/load will not be interfered by other processors. Scalability can be easily achieved, as the size of memory will increase proportionately when the number of processors increases. Besides, if a problem requires a large amount of data, a memory limitation will not be easily reached if data is stored in a distributive manner. However, communications between processors are all remote communications with low efficiency, which will take a longer time than local memory access.

For the reasons above, the hybrid shared-distributed memory system is applied to clusters (Figure 4.6). With this structure, different small shared memory systems are connected together within a network. The advantages of the two systems are combined. The system is scalable with the number of nodes and processors. Also, the complexity of development is weakened.

With the great effort of Compute Canada and Westgrid, we can get access to their

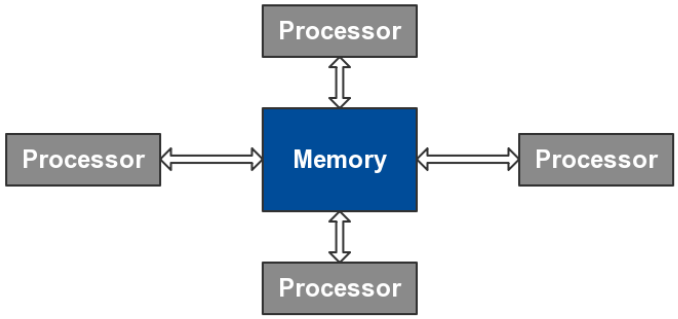

Figure 4.4: Schematic of shared memory architecture.

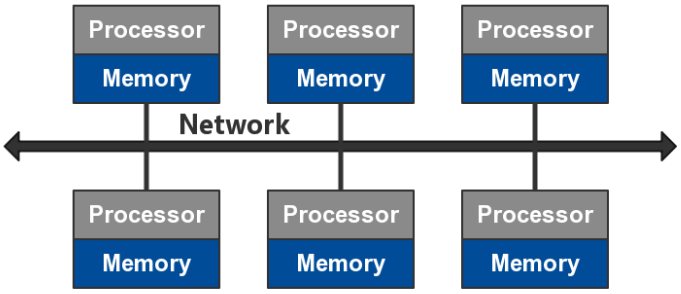

Figure 4.5: Schematic of distributed memory architecture.

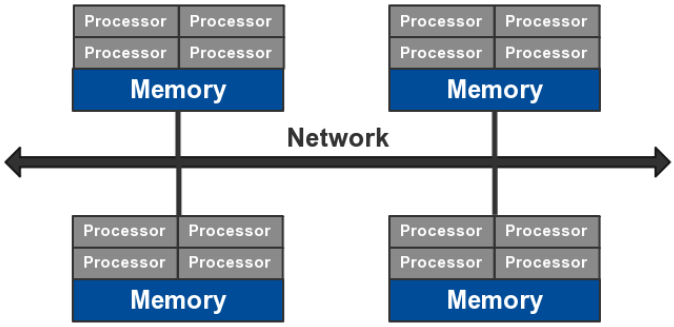

Figure 4.6: Schematic of hybrid shared-distributed memory architecture.

clusters, such as Niagara. Therefore, our parallel simulator can be developed successfully.

At present, CMG STARS only support parallelization on shared-memory structure computers (CMG (2015)). Therefore, the main difference between our simulator and CMG

STARS is the architecture of parallel computers. With the distributed memory architecture, the scalability will achieve a better result when the number of cores is about 100 or even more.

4.3 Data communication

In the illustration of the distributed memory architecture, we discussed processor communications. Communications are the processes to exchange data across different computing units. In most of the cases, a parallel program requires data exchange on a processor. Network latency is introduced in total computation time. Resulting from a delay on hardware, latency is the time spent on sending a message from one point to another, constituting an important part in the total overhead. An inefficient process of communications will cause a significant overhead. Thus investigation of communications is of great importance.

For clusters, there are two different APIs (application programming interfaces) that can be used to fulfill all communications, Message Passing Interface (MPI) and Open Multi-Processing (OpenMP). The role of two APIs is shown in Figure 4.7.

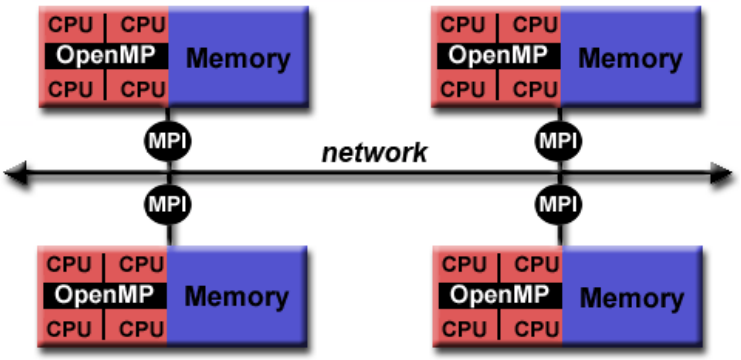

Figure 4.7: Role of MPI and OpenMP on a cluster (Barney, B. (2007)).

Communications occurring between processors on different nodes use MPI, which is ac-

complished over a network. Communications occurring between local processors on a computational node use OpenMP, which is based on a shared memory. Thus a cluster is actually a combination of a message passing model and a thread model.

Communications on a node are much faster than communications between nodes. Thus latency is mainly resulting from message passing. In 1994, Message Passing Interface (MPI), the industry standard for message passing, was first released. Thus we also adopt MPI for the standard of message passing.

Message passing is required when one processor cannot get access to the data which is stored on another node. There are several types of operations of message passing defined by MPI. Generally, the operations can be divided into two parts.

The first part is a point-to-point communication. For this kind of communication, the action is finished by a pair of processors. In one communication, the message is passed from one processor to another. During this procedure, $MPI\_Send()$ is called by the message sender and $MPI\_Recv()$ is called by the receiver. Both functions are from the MPI library (Figure 4.8 (1)).

But in some programs, there may be many communication pairs. Thus the identification between a desired sender and a desired receiver is necessary. In other words, in one single message passing between one sender and one receiver, a sender needs to select a section of memory, copy it, and send it. Nevertheless, as a receiver, many difficulties may arise.

First, in a program, a large number of messages are not only passed by a sender, but also by other processors as well. To deal with this problem, a rank is introduced. Each processor has a unique rank as an index among all processors. In the operation of $MPI\_Send()$

and $MPI\_Recv()$, the ranks of both a sender and a receiver need to be specified in both functions as a parameter. This can prevent the receiver from capturing a message from other processors.

Second, with a confirmed processor pair, the problem of a message passing order remains. For example, there may be two different messages (A and B) passing between two processors. Message sending of A is earlier than message sending of B. However, due to the latency of a network, there is a possibility that message A arrives later than message B. If the receiver captures the first-arrived message first, it may receive the wrong message, causing a failure in execution. Therefore, a tag is used to identify the message passing between the same processor pair. The tag is used as a parameter when calling a function.

Third, what is coming next is the deadlock phenomenon. Taking a sender as an example, if a message is sent, there are two options. One is blocking a communication: waiting until this message is received by the receiver. While waiting, no instructions are executed by the processor. Another is a non-blocking communication: after sending the message, the data is put into the buffer and the processor keeps executing the following instructions.

For the blocking option, there is a case in which different processors send messages at the same time and receiving is not accomplished by any processor. Then all the senders will have to wait until receiving is finished and returned. If no receiving is returned, all processors will continue waiting, resulting in a deadlock.

As for the non-blocking option, although the deadlock of this type can be prevented, there is also a disadvantage. That is, exception control is weak for non-blocking communications. Additionally, the implementation of non-blocking introduces low maintainability and high

coding complexity. Nevertheless, the deadlock will still occur for an inappropriately coded structure. For example, if there are two processors and all communications only occur between them, each processor first call the $MPI\_Send()$ function followed by $MPI\_Recv()$. In this case, a deadlock also happens as no message is sent and receiving is impossible.

Therefore, deadlocks can only be prevented by setting up a proper structure for parallelization.

For all the difficulties introduced by point-to-point communications, precise development is required in coding. This includes structural design and instruction implementation, which are all fulfilled in our simulator.

In addition to the point-to-point communications, there are also collective communications, which refers to message passing among multiple processors. There are three types of communications: one-to-all, all-to-one and all-to-all communications.

For one-to-all communications, data on one processor ("root" processor) is passed to multiple processors. If the data is a whole trunk, it is a broadcasting process ($MPI\_Bcast()$) (Figure 4.8 (2)). If the data is segmented and passed separately, it is a scattering process ($MPI\_Scatter()$) (Figure 4.8 (4)).

For all-to-one communications, data on multiple processors are passed to one processor ("root" processor). If the data is collected, calculated, and stored as one single variable, it is a reducing process ($MPI\_Reduce()$) (Figure 4.8 (3)). If the data is collected and stored separately, it is a gathering process ($MPI\_Gather()$) (Figure 4.8 (5)).

If messages are sent from multiple processors and received by multiple processors, it is an all-to-all communication, which can be implemented by $MPI\_Alltoall()$, $MPI\_Allgather()$

and $MPI\_Allreduce()$.

All of these communications are called in our in-situ combustion simulator, making parallel computing feasible. In general, it is a better choice to apply collective communications than the combination of point-to-point communications, since optimization has been achieved in collective communication functions to overcome difficulties such as deadlocks. If the implementation is optimized based on data flow and suitable for network topology, the implementation is a good MPI implementation (Hager, G. and Wellein, G. (2010)). This optimization of implementation is achieved mainly through collective communications.

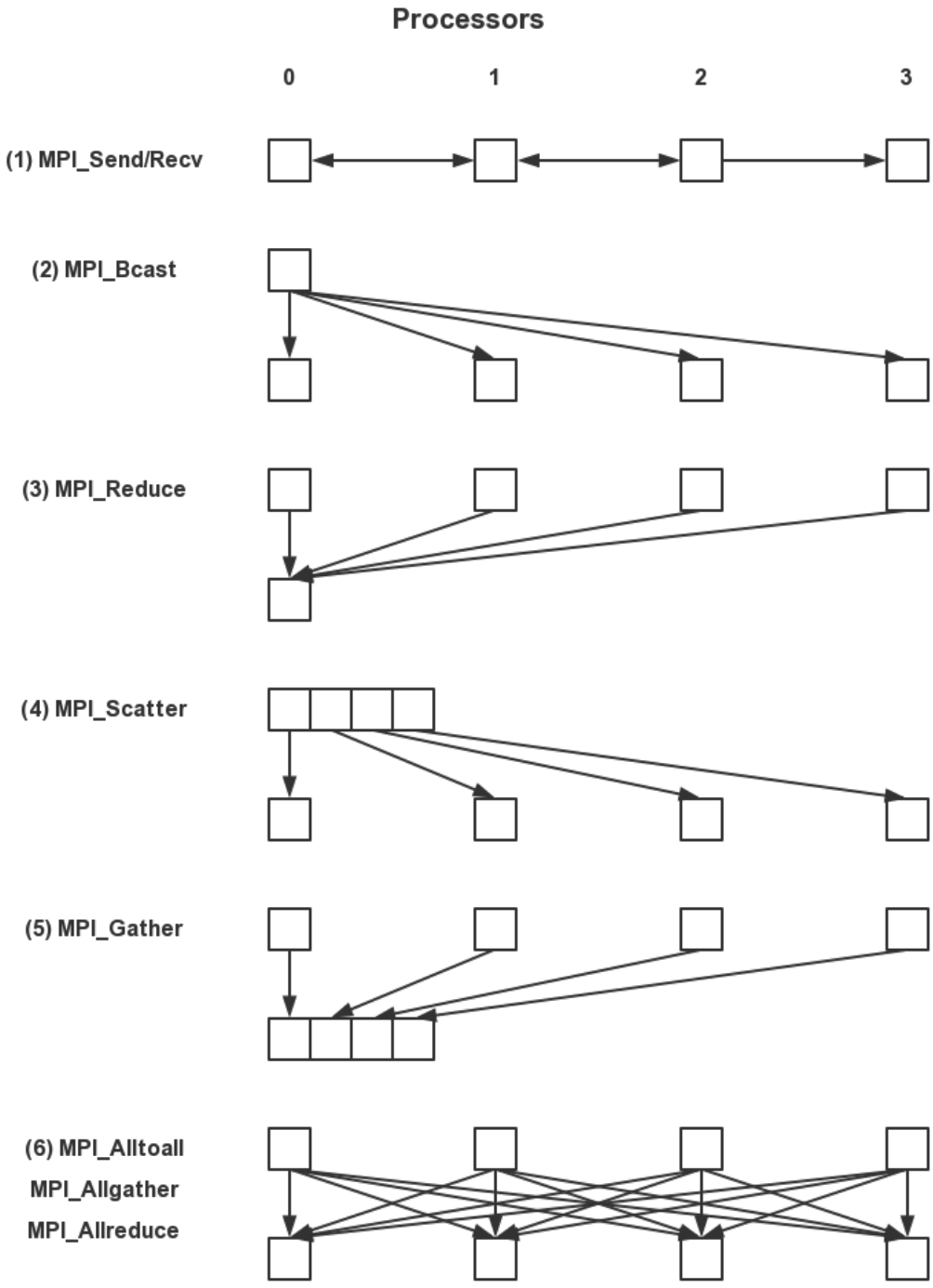

Figure 4.8: Typical types of communications in MPI.

4.4 Quality of parallelization

The goal of parallelization of the ISC simulator is to accelerate simulation. Speed is the main factor for us to evaluate the quality of parallelization.

Ideally, with an increasing number of processors, a proportionate increase in parallel speed is expected. However, overhead prevents the program from achieving the ideal scenario. With a good performance of increased speed, it is also an indicator of good scalability.

Speedup is commonly used as a criterion. For a model with a fixed size, it presents the efficiency of parallelization implementation of a program. It is defined as:

$$speedup = \frac{speedwithmoreprocessors}{originalspeed} = \frac{executiontime}{executiontimewithmoreprocessors}.$$

Ideally, a proportionate increase in speedup with the number of processors is expected. In fact, a smaller speedup is common in a parallel program, as additional time is wasted in execution. This waste in time is the parallel overhead.

Overhead is the additionally-required time in parallel execution, which is mainly influenced by data communication and synchronization.

As discussed before, communication overhead results from factors such as latency. A direct thought to reduce the communication overhead is to decrease the total volume of communications. This part will be illustrated in grid partitioning.

As for synchronization, overhead mainly comes from load imbalance. Here, the terminology of load balance is introduced: all the processors are kept busy at all times. On the contrary, load imbalance can be explained in Figure 4.9. The workload of different tasks is

not equal. If all tasks are assigned to one processor and the speed of all processors are all the same, some processors will finish their work earlier while others are still busy executing all the instructions assigned. Before synchronization, the processors which finish earlier will have to wait until all processors finish their work. In the figure, as an example, the workload of task 1 is heavier than any other tasks, causing all other processors to wait. This is a waste of computing power. If the workload is equally assigned, all processors will finish at about the same time, and the computing resource is utilized at its maximum.

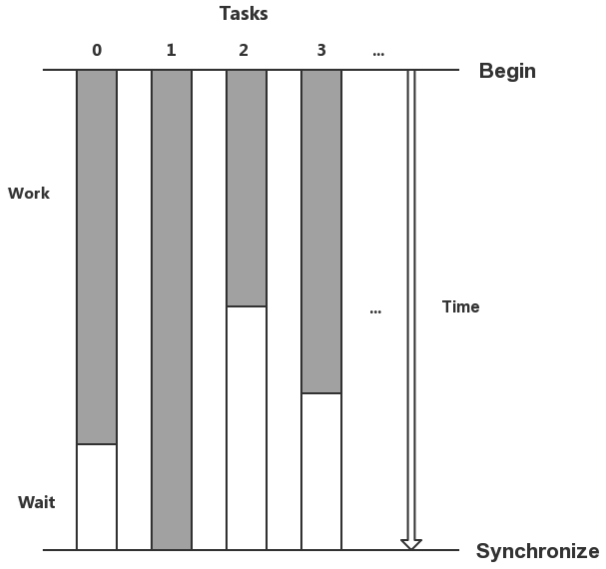

Figure 4.9: Schematic of load balancing.

Therefore, a smaller volume of communications and better load balancing will improve the quality of the parallel implementation of the in-situ combustion simulator.

4.5 Grid partitioning

A regular reservoir can be described as a hexahedral domain with hexahedral cells. A grid can be uniform or non-uniform. Structured grids support finite difference methods, finite volume methods, and finite element methods. They have been widely applied in most commercial reservoir simulators. A grid and coordination are shown in Figure 4.10.

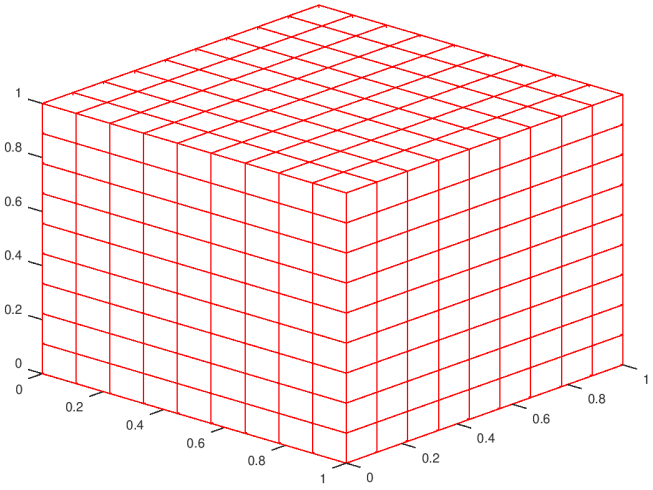

Figure 4.10: Structured grid.

A cell-centered grid is used by our simulator. For this type of grid, cells are the elements in the whole grid. All data is stored within the unit of a cell. For all cells constituting the whole grid, all cell data, referring to properties most of the time, constitutes all the data in a model.

First of all, each cell has a unique integer coordinate (i, j, k) $(i \geq 0, j \geq 0, k \geq 0)$. Here, i is the x coordinate component along the x axis, j is the y coordinate component along the y axis and k is the z coordinate component along the z axis. All coordinate components are numbered from the negative direction to the positive direction along each axis. In addition,

an index of a cell is calculated by:

$$index = nx \cdot ny \cdot k + nx \cdot j + i,$$

where nx and ny are the numbers of cells in the x-direction and y-direction, respetcively. This numbering is the global index, since a partitioning of the grid is not yet applied.

With global indexes of cells, vectors can be adopted to store all properties in a model. The index of a element in a vector directly corresponds to the global index of a cell.

As discussed before, the main bottleneck for adopting the shared-memory system is a large-scale model, with a large amount of data to be stored in the memory. With clusters, the large amount of data can be first divided into different parts and then stored in each node separately. Since data is all based on cells, we consider decomposing the whole grid into sub-grids of cells. For each sub-grid, we call its cells the local cells, as data is stored in the form of vectors. In fact, when we say that data is stored separately, it is different parts of vectors that are stored separately.

For a part of a vector, with the aim to save memory resource, only elements corresponding to the local cells are stored in the form of local vectors. As a local vector, the length is shortened, and there is a natural index of this local vector. With one-to-one correspondence between vector elements and cells, there is also a local index besides the global index for a cell. A map is introduced to express the relationship between the global index and local index of each cell.

For grid partitioning, we denote a grid at $\mathbb{G}$, which is a union of all the cells in the grid. For N_p processors, the grid is partitioned into N_p sub-grids, and each sub-grid consists of

a number of cells. For a sub-grid $\mathbb{G}_i$, it is assigned to the i-th processor. Moreover, for all sub-grids, we see that

$$\begin{cases} \mathbb{G}_i \neq \emptyset, & i = 1, \cdots, N_p, \\ \mathbb{G}_i \cap \mathbb{G}_j = \emptyset, & i \neq j, \\ \cup \mathbb{G}_i = \mathbb{G} & i = 1, \cdots, N_p. \end{cases}$$

For a single cell in the sub-grid $\mathbb{G}_i$, it is an interior cell for $\mathbb{G}_i$. Also, this cell has six neighboring cells in the space. For a neighboring cell, if it is located in another sub-grid $\mathbb{G}_j$, $i \neq j$, then it is a remote neighbor. An interior cell itself is a local neighbor.

Once a global grid is set up, the approach to partition the global grid and build the relationship between the global and local grids is the first step to implement parallel computing.

First of all, for grid partitioning approaches, the main principle is to improve the quality of parallelization as much as possible. Thus parallel overheads need to be minimized in the partitioning. Therefore, a reduction in the communication volume and improvement of load balance are required for a partitioning algorithm.

For load balance, we assume that calculations of any cell are similar in a grid. The best case for load balance is that every processor is assigned with a sub-grid with the even number.

However, the assumption does not always hold. For example, if a water phase or oil phase disappears in one cell, the nature of a small cell matrix gets worse, and then more iterations will be required in a linear solver to get changes of unknowns. If a large number of these cells exist in one sub-grid on one processor, the linear solution converges in fewer

iterations with a shorter time, and other processors will have to wait until the solution on this processor converges. Therefore, a typical load imbalance occurs.

To deal with this problem, a dynamic load balancing grid partitioning method is required. With this method, the solution time on each processor is monitored, and a modification of partitioning is done periodically. With the historical workload, optimization of partitioning is fulfilled dynamically. So, this method is called the dynamic load balancing grid partitioning method. This kind of dynamic load balancing grid partitioning method is fulfilled in ParMETIS, Zoltan and HSFC (Karypis, G., Schloegel, K., and Kumar, V. (1997); Boman, E. et al. (2007); Liu, H. (2008); Liu, H., Wang, K., Yang, B., et al. (2017); Liu, H., Zhang, P., et al. (2016)).

As discussed before, in terms of a communication volume, data communication is required when data is owned by other processors. It is needed as data are all corresponded with a cell. If a cell is in other sub-grid, the corresponding data is owned by another processor, leading to communications taking place between cells in different sub-grids. Based on the knowledge in the previous chapter, we know that for a single equation in one cell, data of all neighboring cells is required. Reflected in a numerical solution, the assembly of a Jacobian system and a linear solver requires data in the neighboring cells.

Therefore, data communications are required by the neighboring cells in different sub-grids. In other words, between a local neighboring cell and a remote neighbor cell, messages are passed. Therefore, to minimize the communication volume, the number of pairs of local neighboring cells and remote neighbor cells need to be minimized.

To sum up, the main objective for a successful grid partitioning is to create partitions

with the equal number of cells and minimize the number of partition interfaces.

With ParMETIS, the package of graph partitioning Karypis, G. and Kumar, V. (1998), a grid can be partitioned topologically, with the two objectives achieved.

4.6 Parallel algorithm

As for the connection relations, a dual graph $\mathbb{G} = (V, E)$ is employed, which can be presented by CSR (Compressed Sparse Row). Thus the adjacency structure can be stored.

As a part of initialization, grid partitioning is finished. Recalling the entire process of the ISC simulator, in Figure 3.3, two processes still remain to be dealt with explicitly.

The first one is assembling a Jacobian matrix. In this process, recalling the equations in the previous chapter, data communications are required between processors. In this process, as a processor, data of remote neighbor cells is collected/received from other processors at one synchronization. All the operations adopted are all listed in the section of data communications. The data is stored as a stand-alone vector corresponding to "remote neighbor cells", and different from local vector entries is all related to local indexes of remote neighbor cells. Here, for remote neighbor cells, although they are not local cells, they have local indexes. So all these cells can be accessed by the local indexes.

Afterward, a Jacobian matrix on a processor will be finished with the data of remote neighbor cells. This data will be cleaned by this processor for the sake of memory management. Once the Jacobian matrix is assembled, all the unknowns will be updated in a Newton iteration, and all the properties will also be updated. Therefore, the updated data will be passed to help assemble the Jacobian matrix for the next iteration.

With the Jacobian system assembled, the system will be passed as a parameter to a linear solver. Also, this procedure is paralleled with the parallel iterative solver. Typically, the linear solver only solves the whole linear system of $A\vec{x} = \vec{b}$. Here A is a square matrix of size $n \times n$. However, as a parallel solver, the system is only

$$[A_{local} \quad A_{remote}] \quad \vec{x}_{local} = \vec{b}_{local},$$

which is not a typical linear system. A_{local} refers to the square block of the matrix, and A_{remote} denotes the remaining block of the whole local block on a processor. This remaining part represents the interactions between cells in a local sub-grid and other sub-grids. Since the number of neighboring cells is limited, the number of local cells without a remote neighbor is much bigger than the number of local cells with a remote neighbor. Therefore, this block A_{remote} is much sparser than A_{local}.

To take an example, we assume that 1-th subgrid is only adjacent to 2-th subgrid. Therefore, with the respect to the 1-th processor and the corresponding linear system, the interaction only occurs between the 1-th subgrid and the 2-th subgrid. So the linear system assigned to the 1-th processor is

$$[A_{11} \quad A_{12}] \quad \vec{x}_1 = \vec{b}_1,$$

where A_{11} is the local block for the 1-th sub-grid, and A_{12} is the interaction block with the effect from the 2-th sub-grid to the 1-th subgrid. For a processor, its aim is actually to solve a linear system in the following form:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \end{bmatrix} = \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \end{bmatrix}$$

With block Gaussian elimination, the linear system changes to

$$A_{11}\vec{x}_1 = \vec{b}_1 - A_{12}\vec{x}_2.$$

With this transformation, a local processor will only be required to solve this local linear system. Furthermore, the data of $\vec{x}_2$ is passed from the 2-th processor.

Chapter 5

Results and discussions

In this chapter, some results of different experiments will be presented and some discussions will be given. Also, validations will be shown to examine the legitimacy of the numerical model. Mainly, results from our simulator RSGISC will be compared with CMG STARS.

5.1 Combustion tube experiment

First of all, a combustion tube experiment is presented as a 1-D case. All the data was constructed from CMG case stdrm001.dat.

In this case, the validity is shown first by the physical properties profile and second by the comparison between the simulator and CMG STARS.

Initial condition	
$k_{x,y,z}$ (md)	12700, 12700, 12700
ϕ	0.4142
p (psi)	2014.7
T $(^\circ F)$	100
$S_{w,o,g}$	0.178, 0.654, 0.168
$x_{LO,HO}$	0.744 0.256
$y_{W,LO,HO,O2,IR}$	0.0, 0.0, 0.0, 0.21, 0.79

Table 5.1: Input data for tube model.

5.1.1 Model description

The reservoir is 0.1602ft long, 0.1602ft wide, and 2.6458ft high, and performed as a one-dimensional model, which is partitioned into 12 cells vertically. Air is injected from the top into the tube and fluids are produced at the bottom. Oil sample in this combustion tube test is light oil sample (24.26 $^\circ API$ at $^\circ F$). The viscosity is 22.6 cp, which is not very high. Therefore, this sample is suitable for Crookston's model, where the combustion is mainly for light oil. LO and HO denote the light oil component and heavy oil component. IR denotes the inert gas pseudo-component, where nitrogen and carbon oxides exist. The set of data is listed below:

Numerical control	
Newton	inexact
Solver	BiCGSTAB
Preconditioner	RAS
Jacobian	numerical
Tolerance of solver	1e-3
Max solver iteration	100
Tolerance of newton	1e-2
Max Newton iteration	15
Max p change (psi)	40
Max S_w change	0.1
Max S_g change	0.1
Max T change ($^\circ F$)	50
Max molar fraction change	0.1

Table 5.2: Input data for tube model (cont'd).

Properties	**H2O**	**LO**	**HO**	**O2**	**IR**	**Coke**
M $(lb/lbmole)$	18	156.7	675	32	40.8	13
p_{crit} (psi)	3155	305.7	120	730	500	
T_{crit} $(^\circ F)$	705.7	651.7	1138	-181	-232	
$\rho_{ref}(lbmole/ft^3)$	3.466	0.3195	0.0914			57.2
cp $(1/psi)$	3e-6	5e-6	5e-6			
$ct1$ $(1/^\circ F)$	1.2e-4	2.839e-4	1.496e-4			4.06
$cpg1$ $(Btu/(^\circ F \cdot lbmol))$	7.613	-1.89	-8.14	6.713	7.44	
$cpg2$ $(Btu/(^\circ F^2 \cdot lbmol))$	8.616e-4	0.1275	0.549	-4.883e-7	-0.0018	
$cpg3$ $(Btu/(^\circ F^3 \cdot lbmol))$	0	-3.9e-5	-1.68e-4	1.287e-6	1.975e-6	
$cpg4$ $(Btu/(^\circ F^4 \cdot lbmol))$	0	4.6e-9	1.98e-8	-4.36e-10	-4.78e-10	
hvr $(Btu/(^\circ F^{ev} \cdot lbmol))$	1657	1917	12198			
ev	0.38	0.38	0.38			
avg $(cp/^\circ F)$	8.822e-6	2.166e-6	3.926e-6	2.196e-4	2.127e-4	
bvg	1.116	0.943	1.102	0.721	0.702	
$avisc$ (cp)	4.7352e-3	4.02e-4	4.02e-4			
$bvisc$ $(^\circ F)$	2728.2	6121.6	6121.6			
$kv1$ (psi)	1.7202e6	1.4546e5	2.7454e5	$kv3$ and $kv4$ are all 0		
$kv4$ $(^\circ F)$	-6869.59	-4458.73	-8424.83			
$kv5$ $(^\circ F)$	-376.64	-387.78	-205.69			

Table 5.3: Input data for tube model (cont'd).

SLT table			
Sl	Krg	Krog	Pcg
0.21	0.784	0	0
0.32	0.448	0.01	0
0.4	0.288	0.024	0
0.472	0.184	0.052	0
0.58	0.086	0.152	0
0.68	0.024	0.272	0
0.832	0.006	0.448	0
0.872	0	0.9	0
SWT table			
Sw	Krw	Krow	Pcw
0.1	0	0.9	0
0.25	0.004	0.6	0
0.44	0.024	0.28	0
0.56	0.072	0.144	0
0.672	0.168	0.048	0
0.752	0.256	0	0

Table 5.4: Input data for tube model (cont'd).

Reaction data		
$LO + 14.06O_2 \longrightarrow 11.96IR + 6.58H_2O$		r_1
$HO + 60.55O_2 \longrightarrow 51.53IR + 28.34H_2O$		r_2
$HO \longrightarrow 2.154LO + 25.96Coke$		r_3
$Coke + 1.18O_2 \longrightarrow 1IR + 0.55H_2O$		r_4
A_R $(1/(psi \cdot day))$		7.248e11, 7.248e11, 1.00008e7, 1.00008e4
E_{aR} $(Btu/lbmol)$		59450, 59450, 27000, 25200
H_R $(Btu/lbmol)$		2.9075e6, 1.2525e7, 4.0e4, 2.25e5
Well conditions		
Injector	air rate (O2:N2 = 0.21:0.79) (ft^3/hr)	0.554
	wi $(ft \cdot md)$	5.54
	pinjw (psi)	10000.0
	tinjw $(^\circ F)$	70
	heat rate (Btu/day)	4800
	heat stop time (day)	0.02083333333
Producer	bhp (psi)	2014.7

Table 5.5: Input data for tube model (cont'd).

5.1.2 Physical behavior

Figures 5.1, 5.2, 5.3, 5.4, 5.5, and 5.6 present the time variations of temperature, oil pressure, water saturation, oil saturation, a light oil molar fraction and coke concentration at cell #0, 2, 4, 6, 8, and 10, respectively.

First of all, for the temperature profile, we can find that at a single cell, the temperature will go up at first and reach the highest point, which indicates that the combustion front reaches this position. Then the temperature will decrease due to the heat conduction and low-temperature air flow from the injector.

Also, for the pressure of each cell, the profile is similar to the behavior of the bottom-hole pressure of the injector. Since the heater is open in the first period, the temperature will increase rapidly in the first cell. Then the water and oil will evaporate in a very short time. Next the pressure of the first cell will rapidly increase. To inject gas with a constant rate, the bottom-hole pressure will also perform a correspondence of the raised pressure. Due to a small size of the model, the change of pressure in the first cell will quickly be conducted on all the cells.

Now, for the water saturation, in the first cell, heated by the well, water evaporation is observed easily. But in the following cells, the water saturation will first rise and then decline. It is mainly about the formation of the steam zone. In the front of the steam zone, the steam will condense for the low temperature. When steam condenses, the total amount of the water component in the water phase will be more than the original amount in the cell. Therefore, the volume of the water phase will increase. But as the propagation of the combustion front goes, the steam zone will also move forward. The water in the cell will all

become steam and then be pushed to the producer at the end.

From the very beginning, the oil saturation will increase a little and then gradually decline. The peaks are because of the early build-up of the pressure, which pushes fluids to flow. For the high transmissibility of the gas phase, the gas phase is much easier to move in the direction of the producer. The oil phase is least likely to flow for the high viscosity with low temperature.

For the molar fraction of the light oil, it will rise up quickly in the first period. It results from the condensation of light oil vapor, just like the rising of water saturation. Since the light oil component is more likely to vaporize than the heavy component, the heavy component will quickly condense near the high-temperature zone. The vapor of the light component will move further. Therefore, at a cell far from the combustion front, the oil vapor is mainly light oil vapor. Moreover, condensation oil is mainly light oil. When the high-temperature zone keeps moving forward, the light component in the cell will also vaporize since the temperature is higher. The residual is the heavy component, which will vaporize with a higher temperature than light oil. Also, in a single cell, the heavy oil in the gas or oil phase will arrive at the cell eventually, and there is a small peak on the oil saturation for this reason. But as the temperature is high enough, the residual of the oil will finally approach 0 for both evaporation and pyrolysis reaction.

The coke concentration profile indicates the pyrolysis reaction. First of all, we can observe a rapid increasing coke concentration when the temperature is higher. In the calculation of a reaction rate, the temperature is an exponential parameter, so the reaction becomes violent in a short time. But when the temperature is about to achieve the ignition temperature,

the coke concentration slows down and reaches its peak very soon. It indicates that the coke oxidation reaction is much faster than the heavy oil pyrolysis reaction at a very high temperature. Also, prior to this point, the total concentration of heavy oil approaches 0, which slows down the reaction rate of pyrolysis and help the rate of coke oxidation surplus and the rate of pyrolysis in a much shorter time. Moreover, we can find that when the coke concentration approaches 0, the temperature of this cell will also reach its peak. When the coke concentration reaches 0, there is no fuel in the cell at all, and, therefore, the temperature will gradually decline.

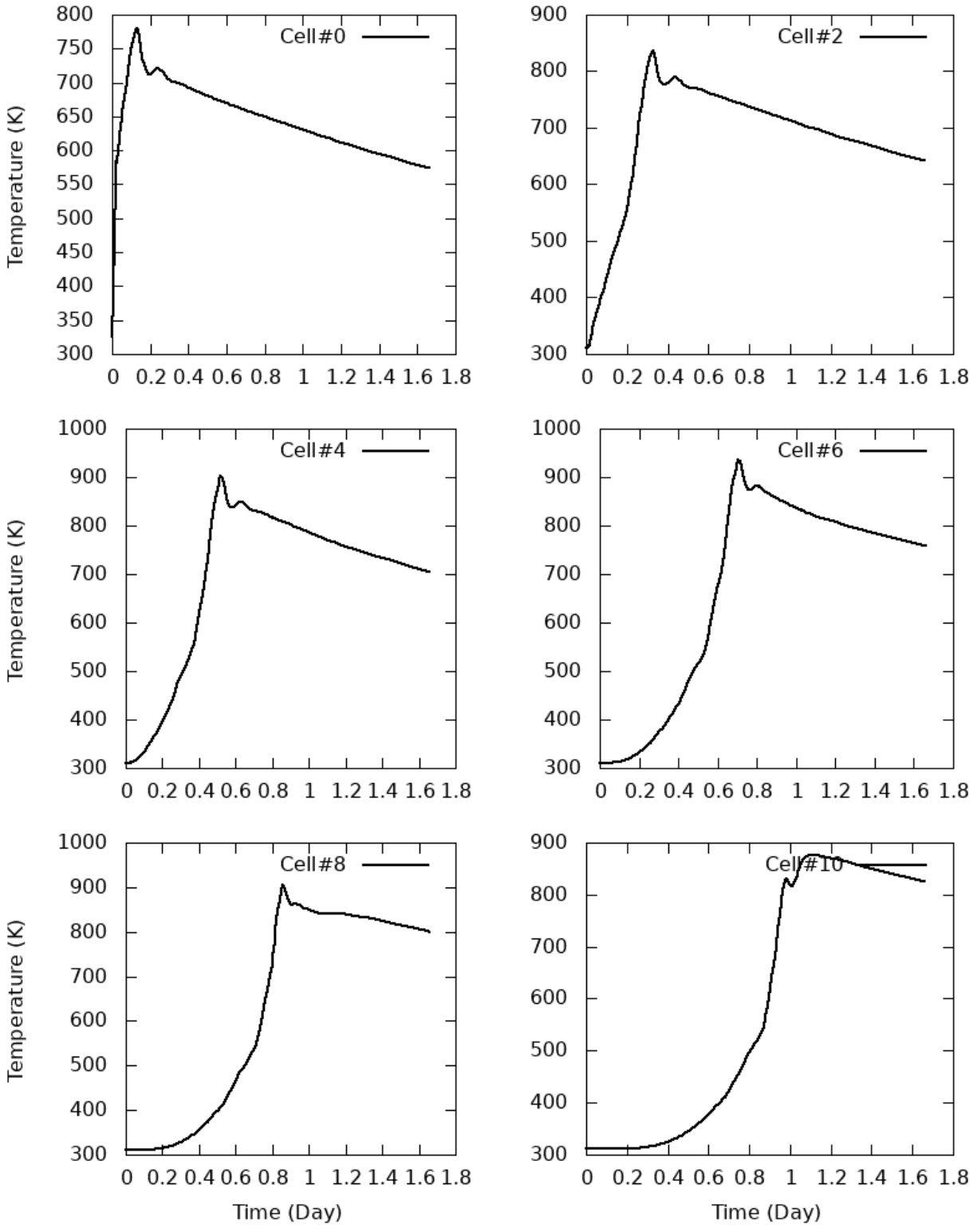

Figure 5.1: Profile of temperature.

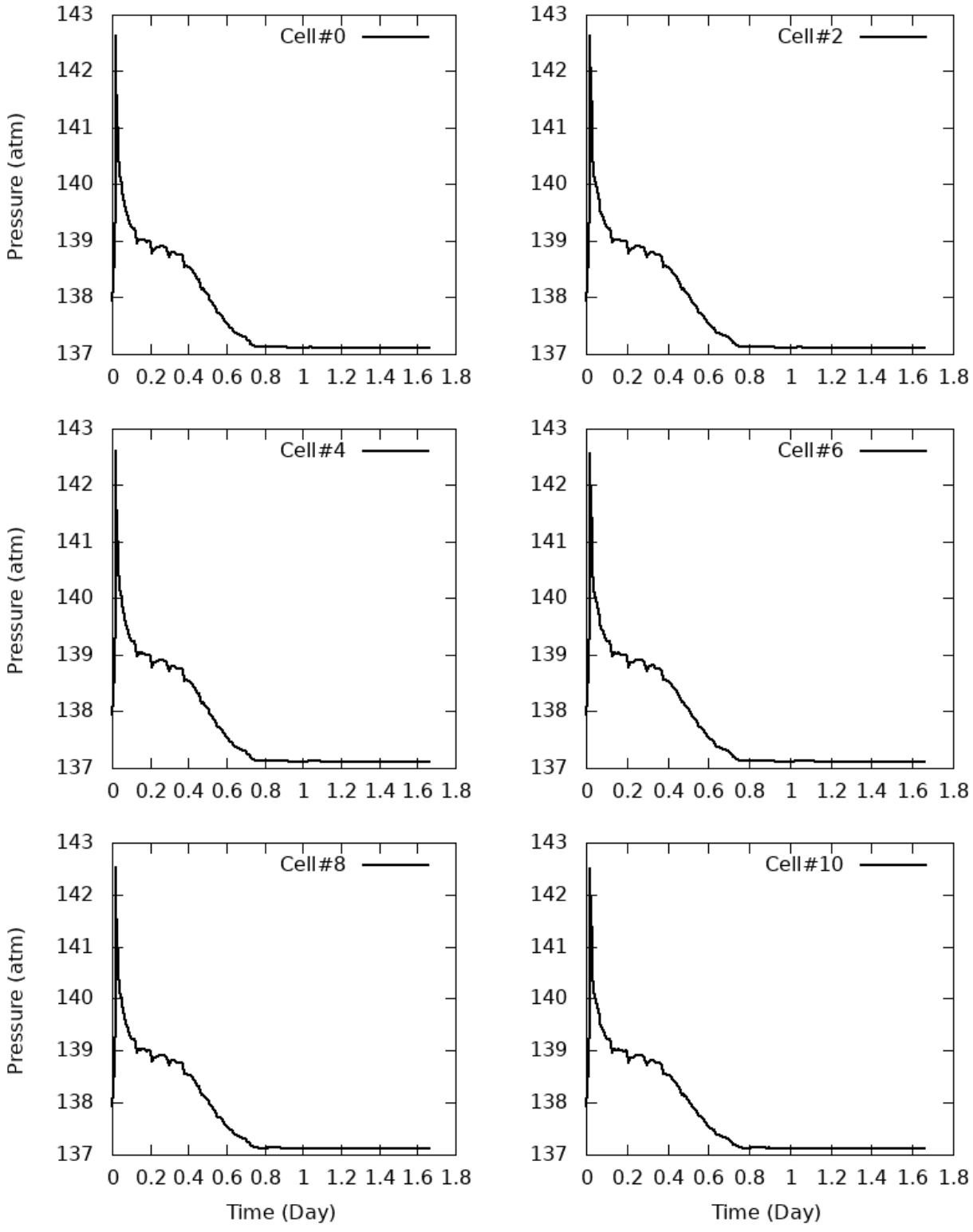

Figure 5.2: Profile of pressure.

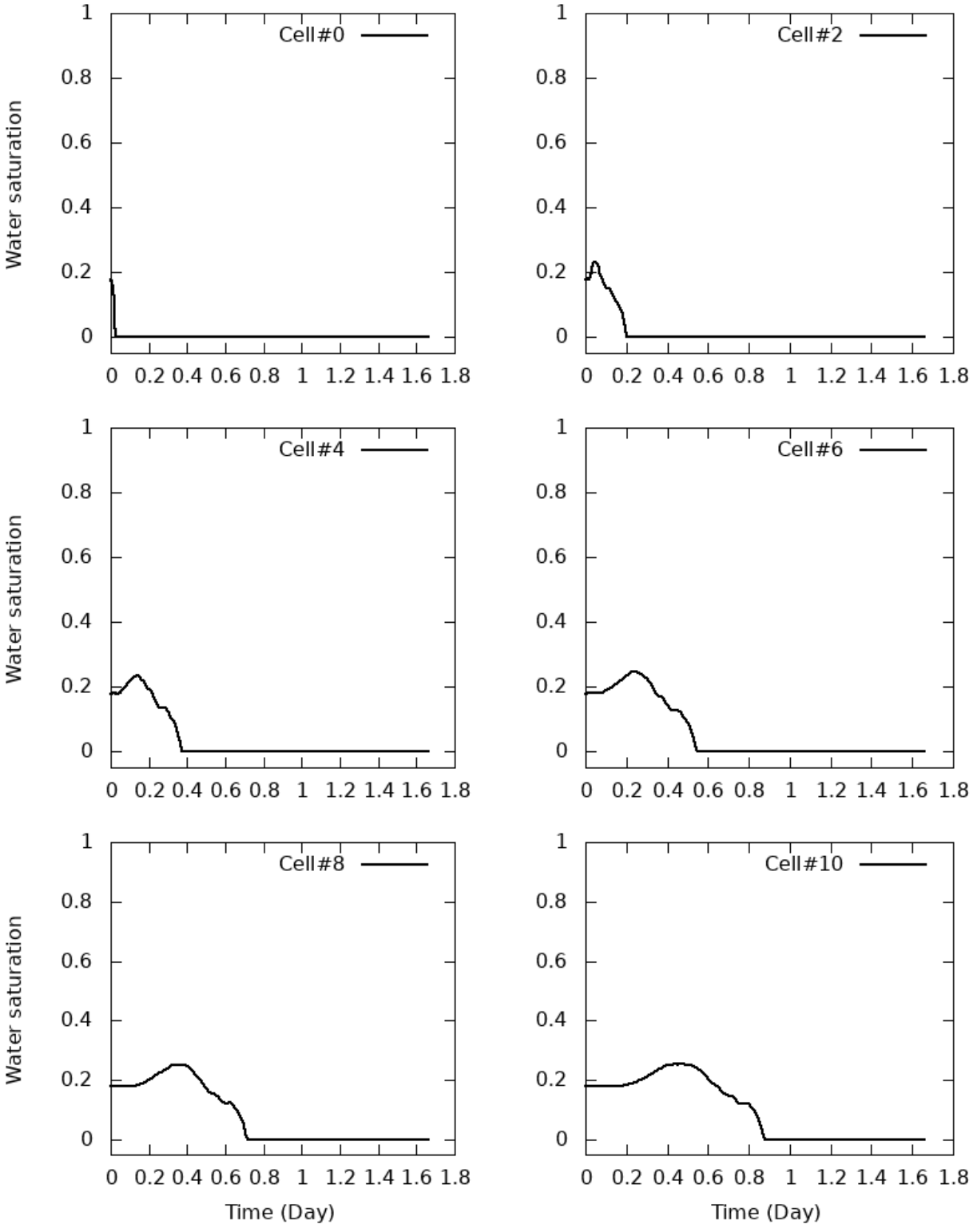

Figure 5.3: Profile of water saturation.

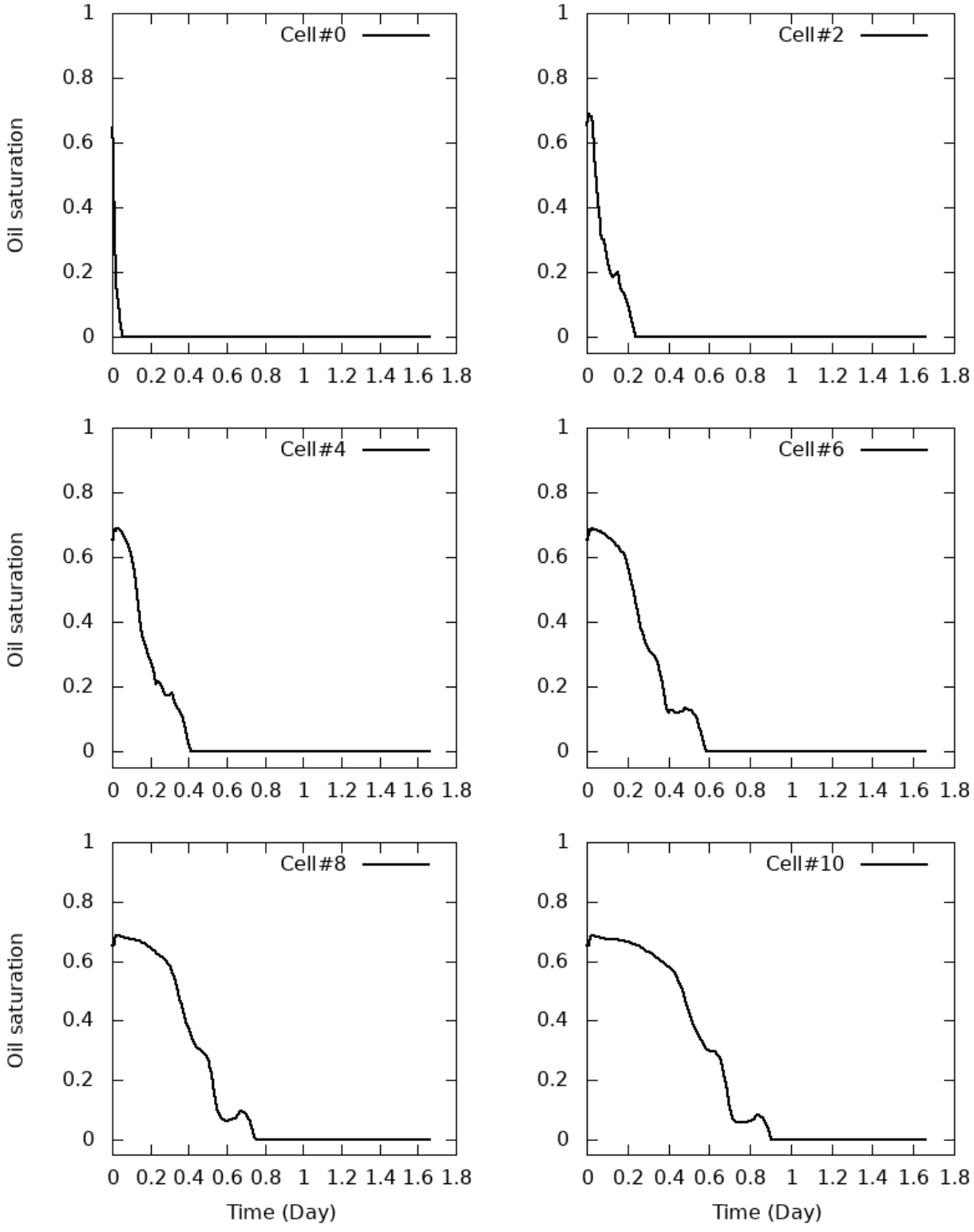

Figure 5.4: Profile of oil saturation.

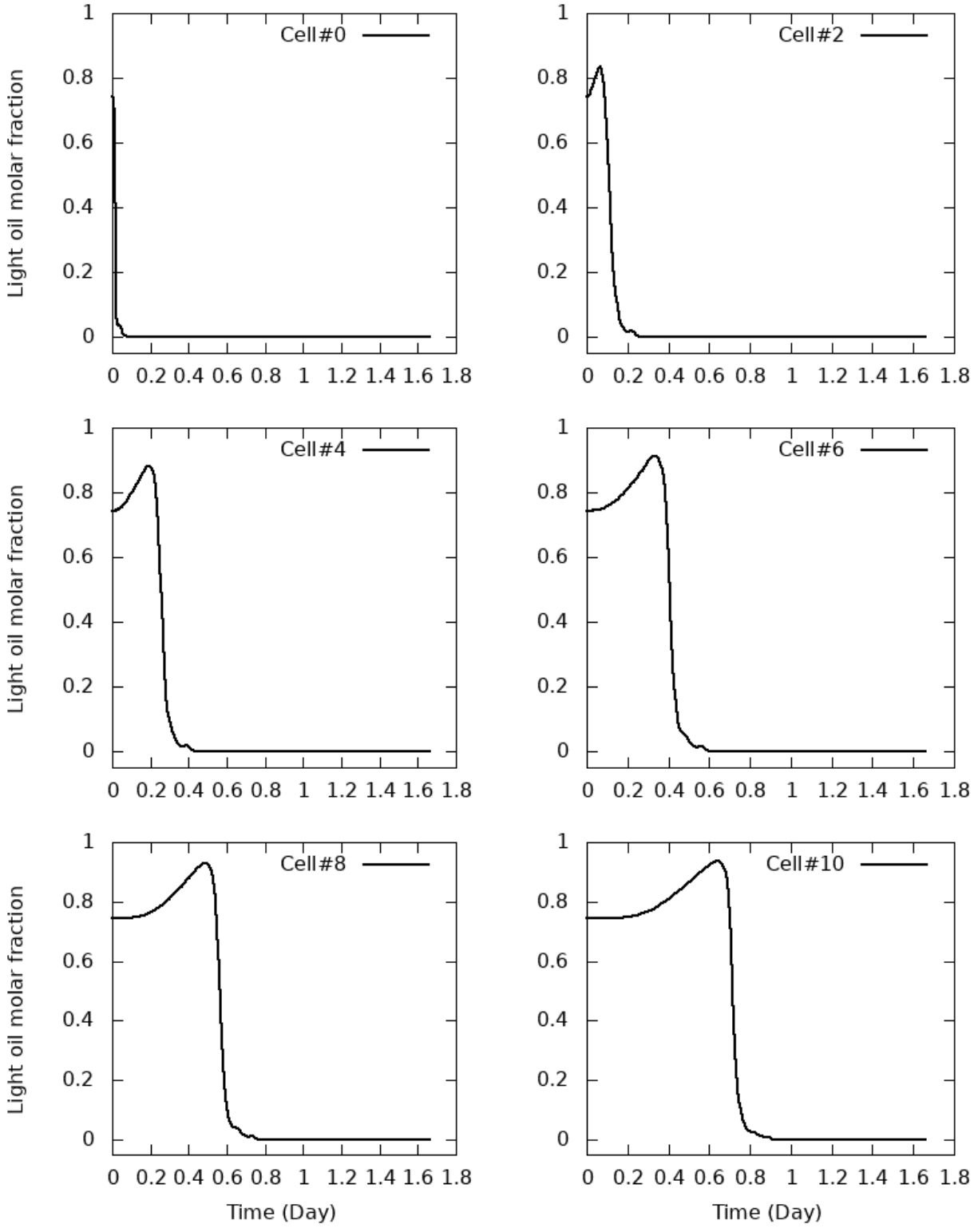

Figure 5.5: Profile of light oil molar fraction.

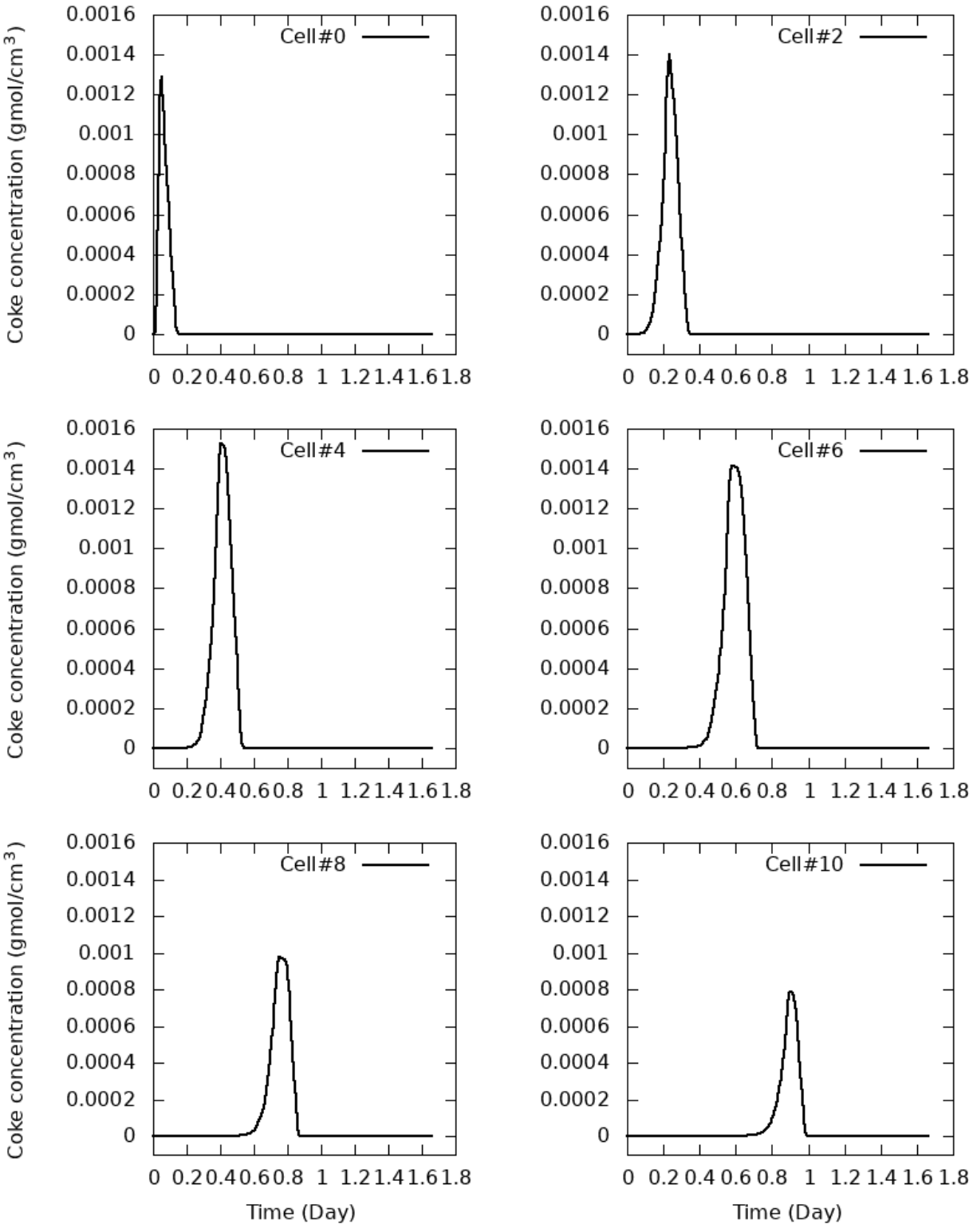

Figure 5.6: Profile of coke concentration.

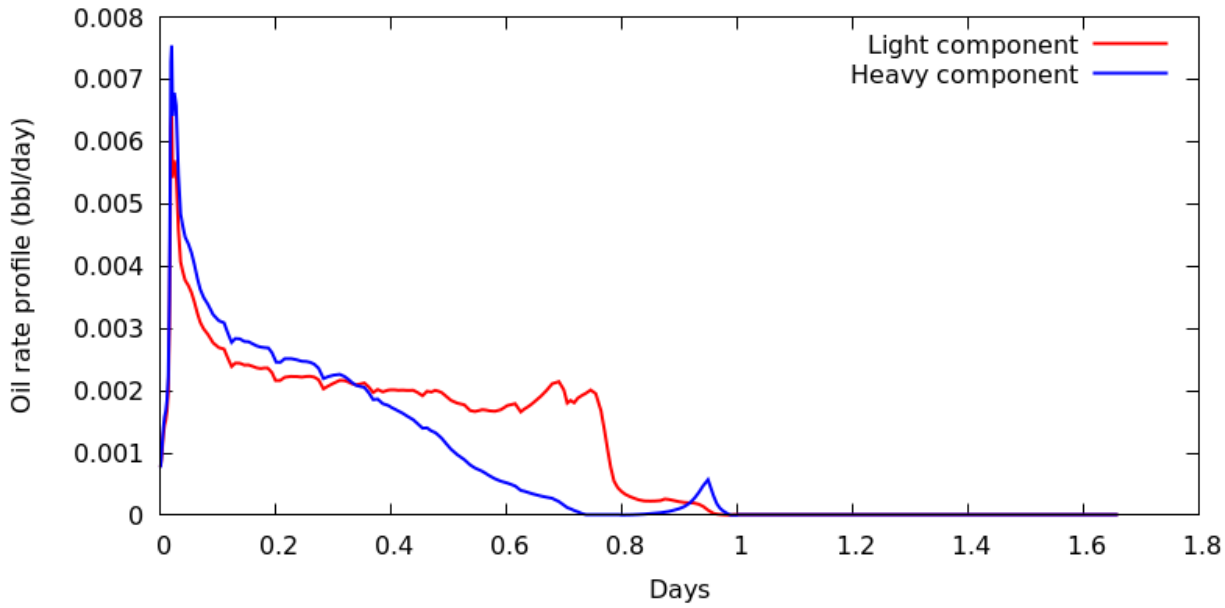

Figure 5.7: Oil rate profile.

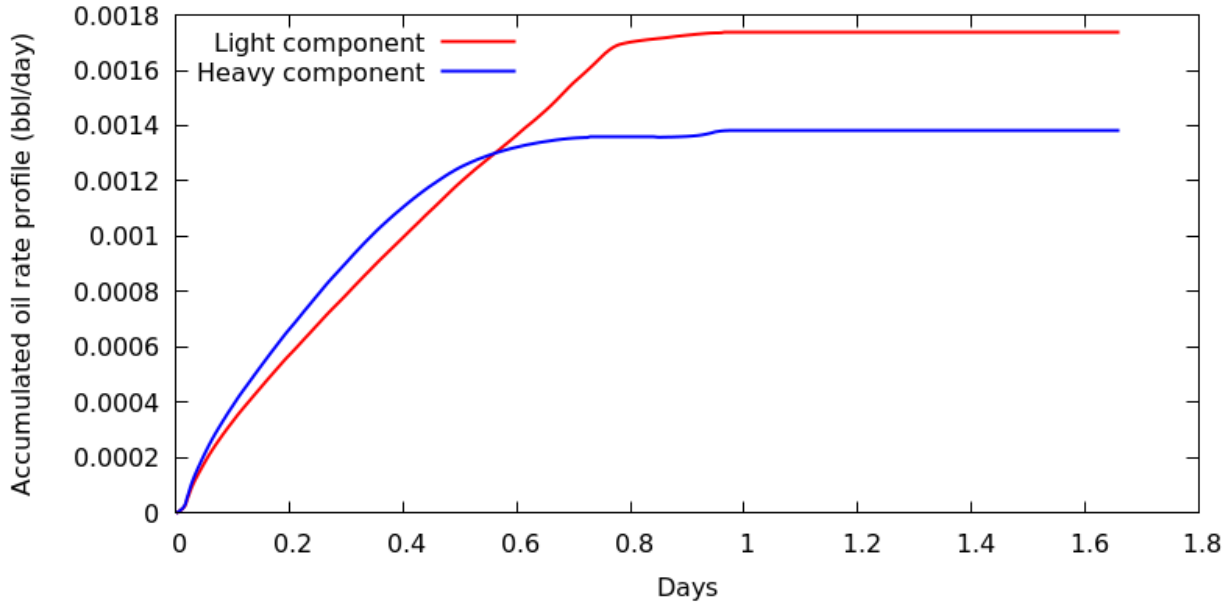

Figure 5.8: Accumulated oil rate profile.

According to the temperature profile versus time over the whole model in Figure 5.9, the area around the temperature peak is the combustion zone. Since the cells in the model are discrete, the peak in the figure and location of the combustion front are not exactly the same. In the direction of downstream of the combustion front, the temperature is raised for heat conduction and convection. With this profile, the propagation procedure of the combustion front is observed.

Also, in front of the combustion zone, there is a cracking (pyrolysis) zone/vaporization zone. The area around the peak represents the cracking zone, since coke is one of the main products in the pyrolysis reaction. Coke is accumulated in this zone. In Figure 5.10, taking t = 0.5 day as an example, the peak of the temperature is around 30 cm and the coke concentration at this position is about 0.0004 gmol/cm$\hat{3}$, which indicates that as time goes, the coke will continue to be oxidized and thus the temperature will increase until a peak is reached. Thus the combustion front will arrive at this point between 0.5 day and 0.7 day.

Also, in this cracking zone, oil and water will evaporate. The vapor will be pushed downstream and form a water bank and an oil bank downstream. This steam zone can be observed mainly from Figures 5.11 and 5.12. Also, the heavy oil component is harder to evaporate than the water component. This phenomenon can be observed in the profile at t = 0.3 day. In the area around 30 cm, the water saturation is 0, but there is still some oil. Moreover, from Figure 5.13, it indicates that at this point, in the oil phase, there is mostly the heavy component, and most of the light component has changed to vapor already and be pushed to downstream. Also, in Figure 5.14, it shows that a large amount of steam is generated in this zone, which takes about one-third of the total gas phase on average. This

is the reason that this zone is called the steam zone.

In front of the steam zone, the temperature and the saturation of oil and water remain the same as the original reservoir. So the effect of combustion has not influenced this part yet. However, in the gas phase, the gas molar fraction of inert gas is very high (Figure 5.16) and there is almost no oxygen in this area, since oxygen is used up in the combustion zone (Figure 5.15). Also, in the profiles of the gas molar fraction of inert gas and oxygen, y_{IR} and y_{O2} increase at the same time. However, the gas molar fraction of inert gas will be even less than its original gas molar fraction in the air. From Figure 5.14, it indicates that it is because there is a large amount of water vaporized at this position, which takes a large portion of the total gas phase.

In Figure 5.17, the characteristics of each zone can be observed more clearly. The most obvious zone is the combustion zone, in which the temperature is higher than in any other positions. Also, in this zone, among the four phases, only the solid phase - coke exists. In front of the combustion zone, the oil phase and water phase are vaporized, forming a vaporization zone. Moreover, the cracking reaction of the heavy component mostly occurs here, and the zone is also called the cracking zone. In front of the cracking zone, the steam takes a large portion of the gas phase, and some vapor of water and oil condenses, forming a water/oil bank. This zone is the steam zone.

Behind the zones we mentioned above, there is a burnt zone, in which only the gas phase exists, and the gas molar fractions are the same as the injected gas. Also, the temperature is gradually lowered for both the injected gas and heat conduction. In the opposite, at the very front of the reservoir, the properties have not been changed. Furthermore, in the gas

phase, mainly inert gas exists.

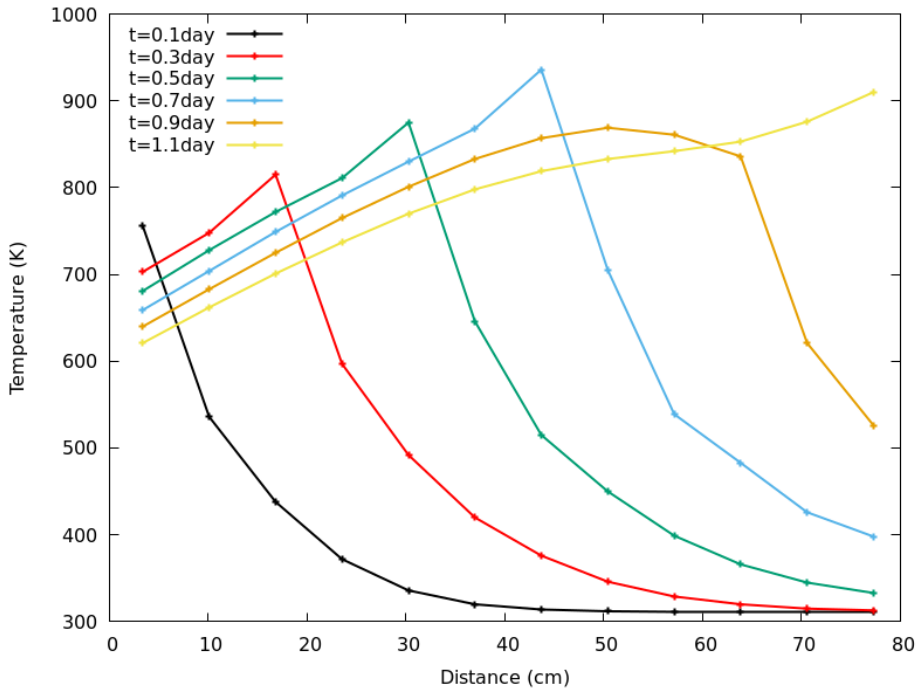

Figure 5.9: Temperature profile over the model at various times.

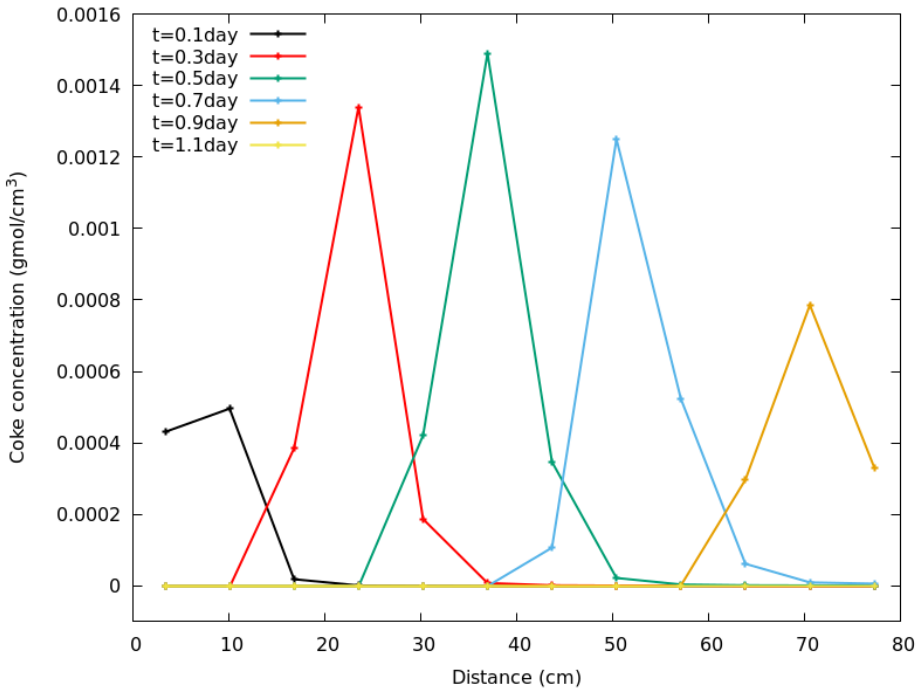

Figure 5.10: Coke concentration profile over the model at various times.

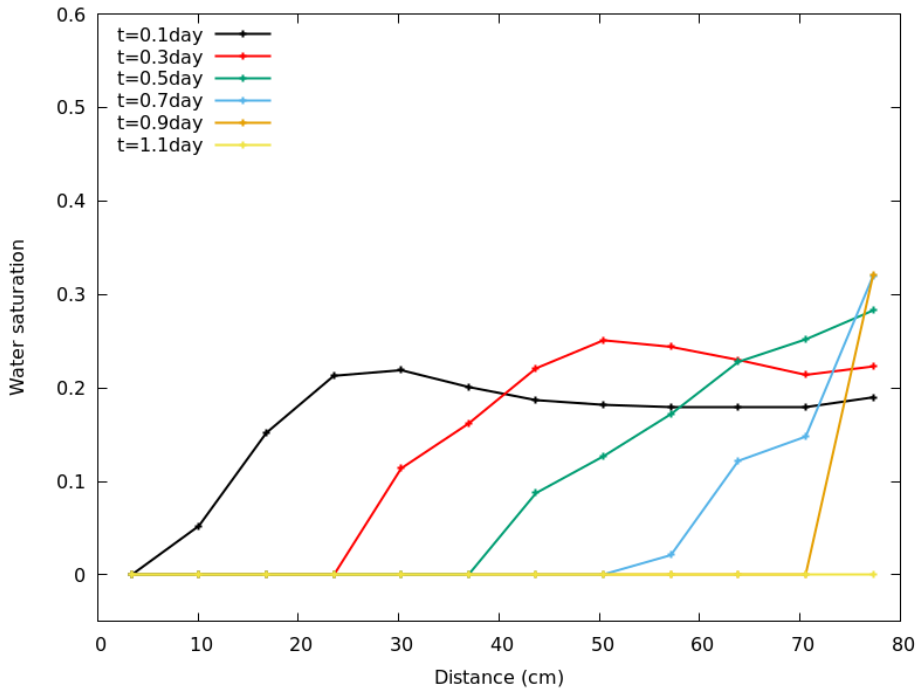

Figure 5.11: Water saturation profile over the model at various times.

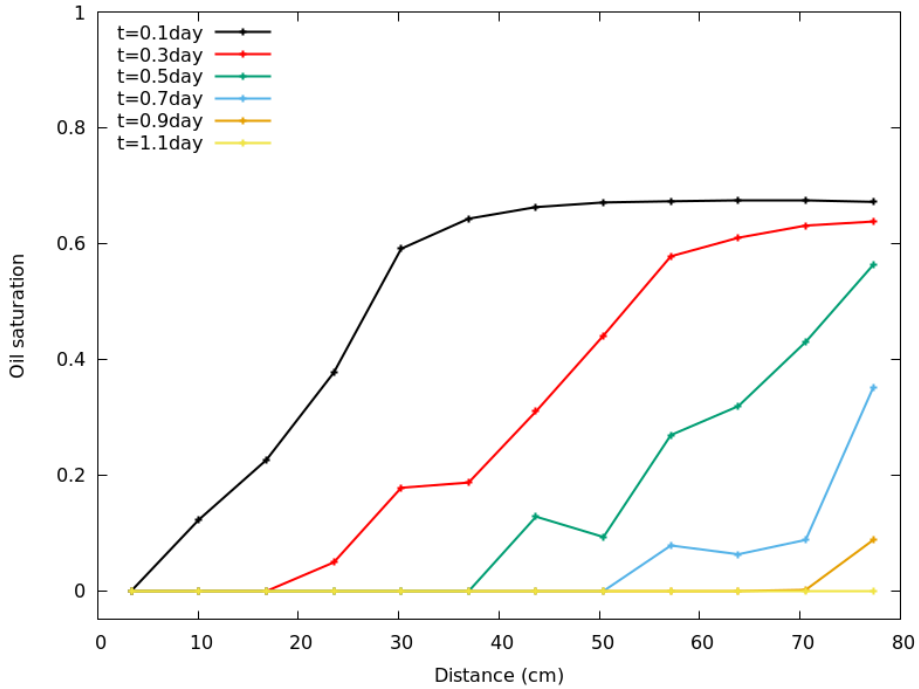

Figure 5.12: Oil saturation profile over the model at various times.

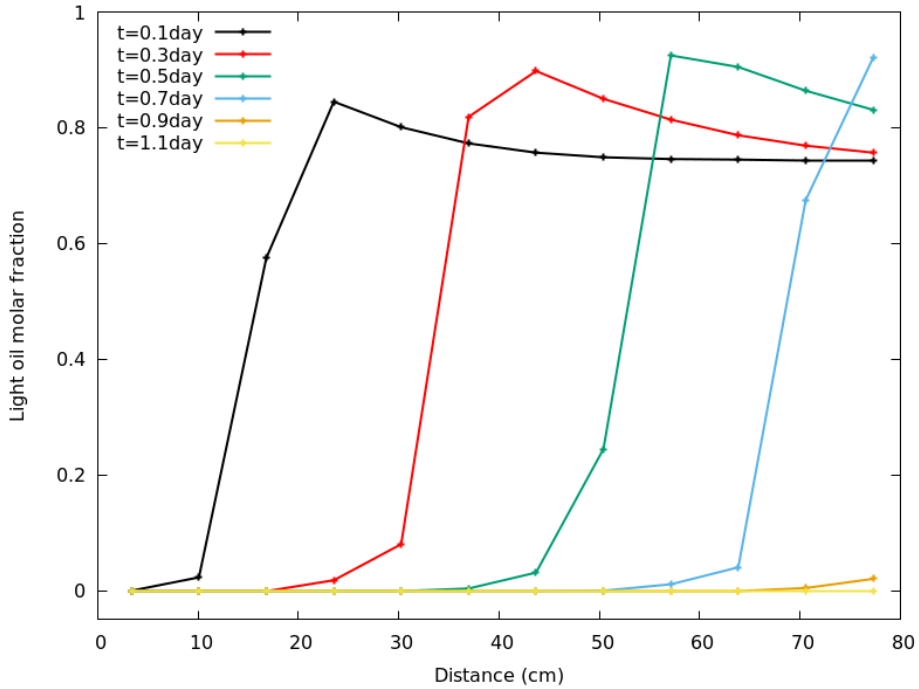

Figure 5.13: Oil molar fraction of light oil profile over the model at various times.

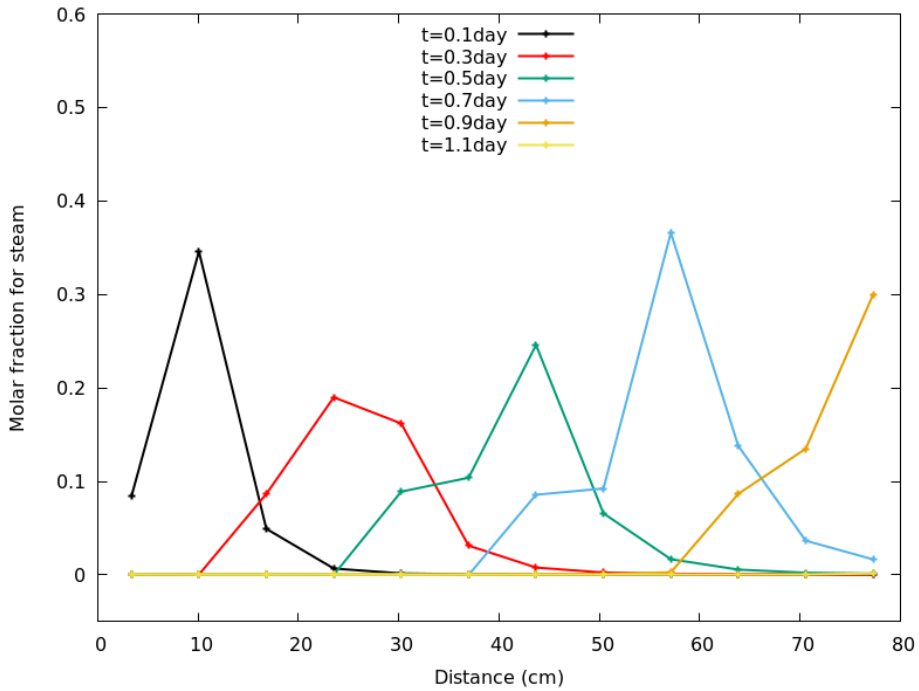

Figure 5.14: Gas molar fraction of steam profile over the model at various times.

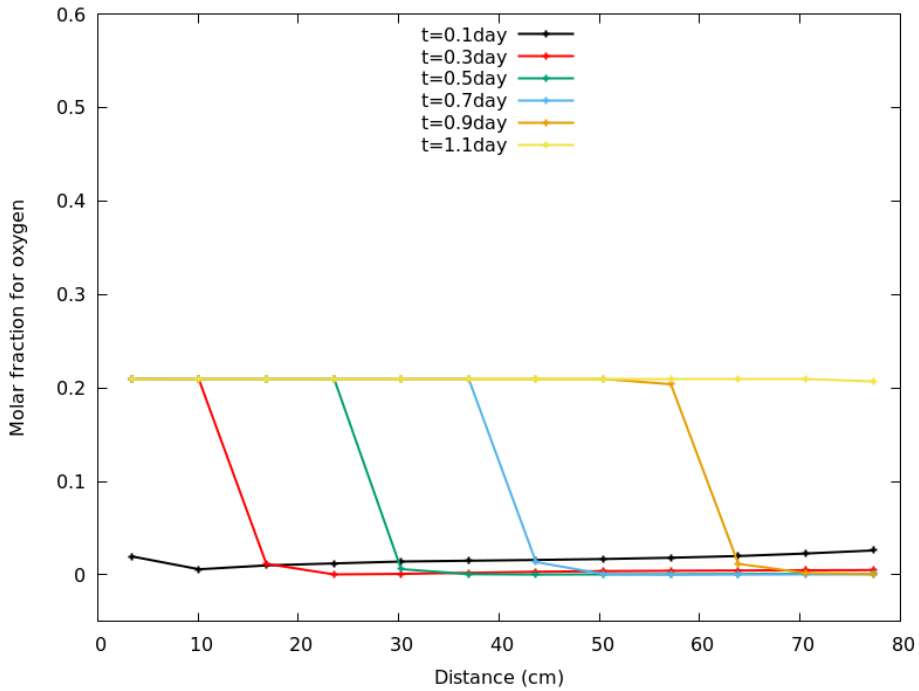

Figure 5.15: Gas molar fraction of oxygen profile over the model at various times.

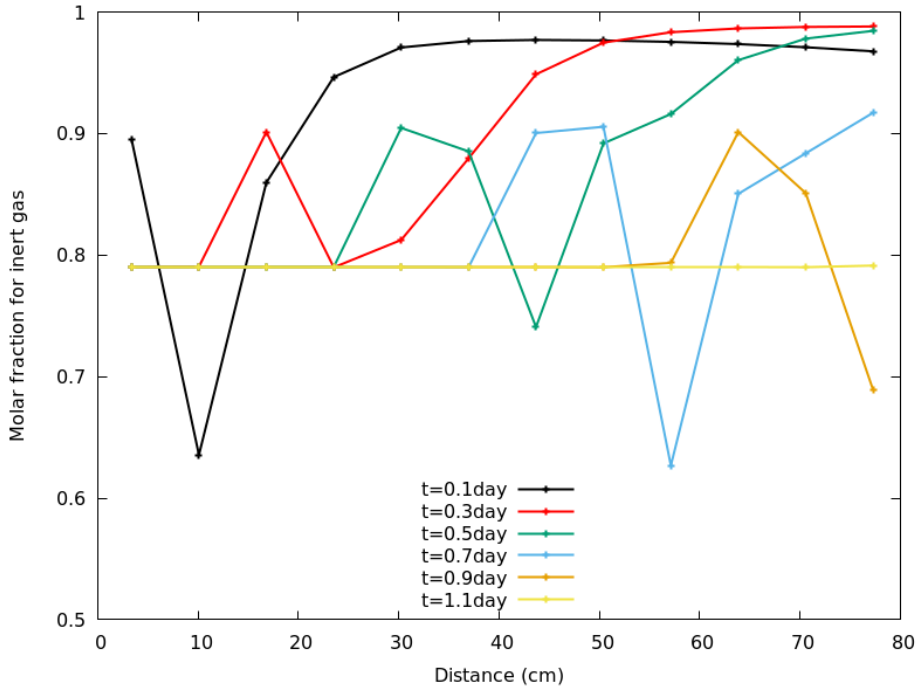

Figure 5.16: Gas molar fraction of inert gas profile over the model at various times.

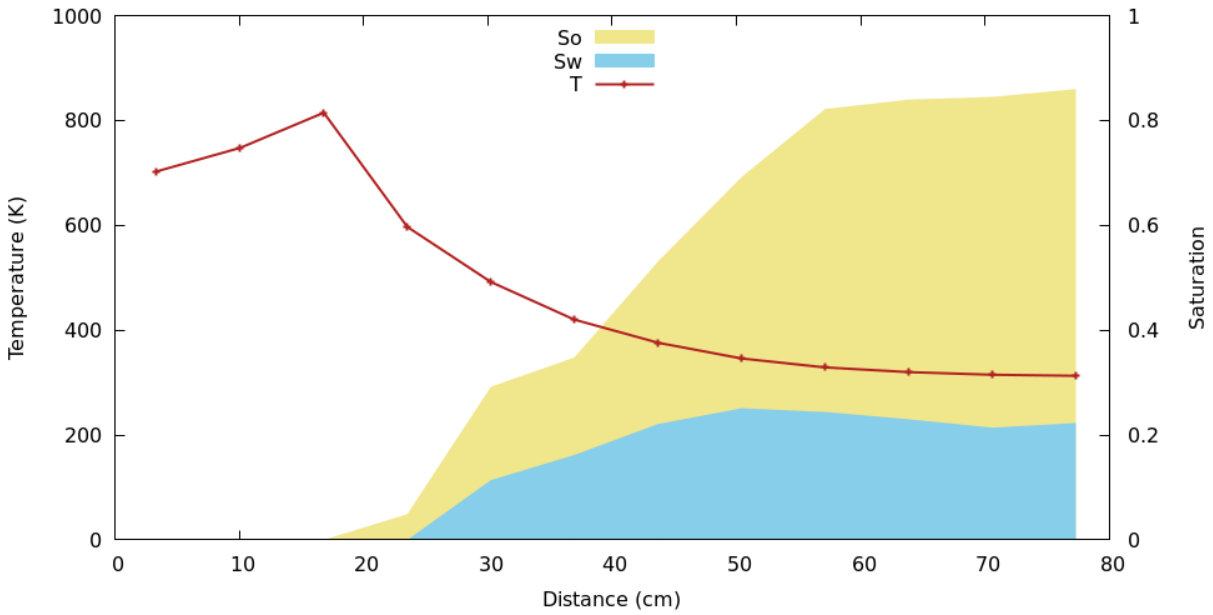

Figure 5.17: Temperature and saturation profile over the model after t = 0.3 days.

5.1.3 Validation with CMG STARS

Before adopting a simulator for practical purposes, the establishment of confidence in its predictive capability is important. To achieve this goal, its simulation results are compared with those by the CMG STARS.

CMG, founded in 1978, is a world-class software company focusing on reservoir modeling and is known for its expertise in heavy oil reservoir simulation. As a key module, released in 1983, STARS is the standard of thermal recovery modeling in the petroleum industry. As mentioned before, the biggest difference between CMG STARS and the RSGISC simulator is that RSGISC is specially designed for parallel computing on distributed-memory clusters. However, STARS currently supports the shared-memory system, and the distributed-memory system has not been supported yet (CMG STARS (2018); CMG (2015)).

From the following figures, it indicates that the results from our simulator RSGISC match very well with those from CMG STARS.

Generally, some properties of three cells (#0, 6, 11) are selected to compare. From Figures 5.18 to 5.25, the profiles of cell properties are presented, including temperature, water saturation, oil saturation, the oil molar fraction of light oil, the gas molar fraction of steam, oxygen and inert gas. Also, well bottom hole pressures are compared with those from STARS.

The properties mentioned above are all the primary unknowns in our simulator, and if these unknowns are validated, all other properties can be obtained by these unknowns. Hence the validated unknowns ensure the validity of other properties.

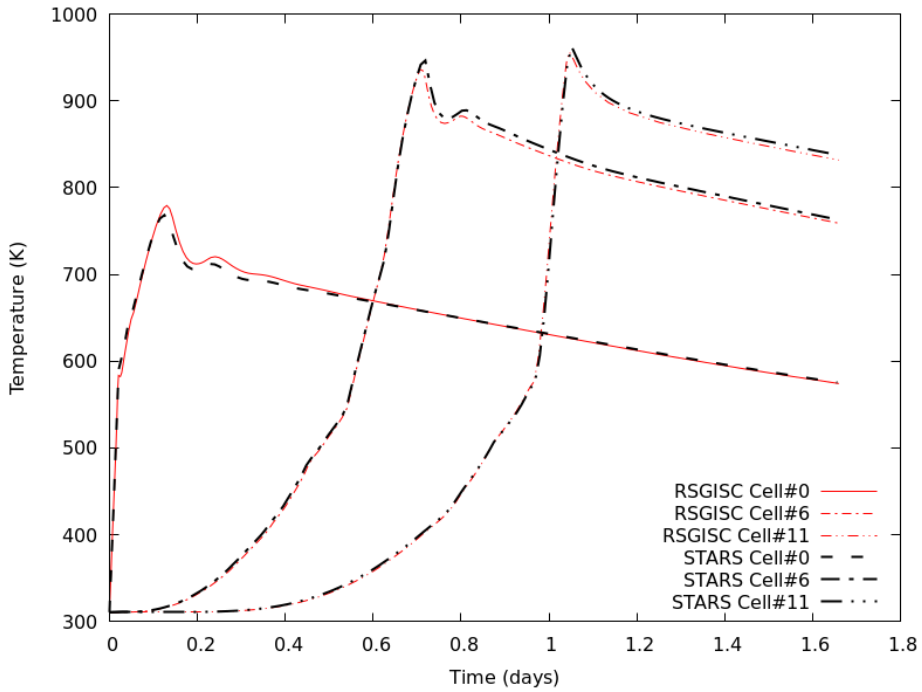

Figure 5.18: Temperature profile with results from CMG STARS.

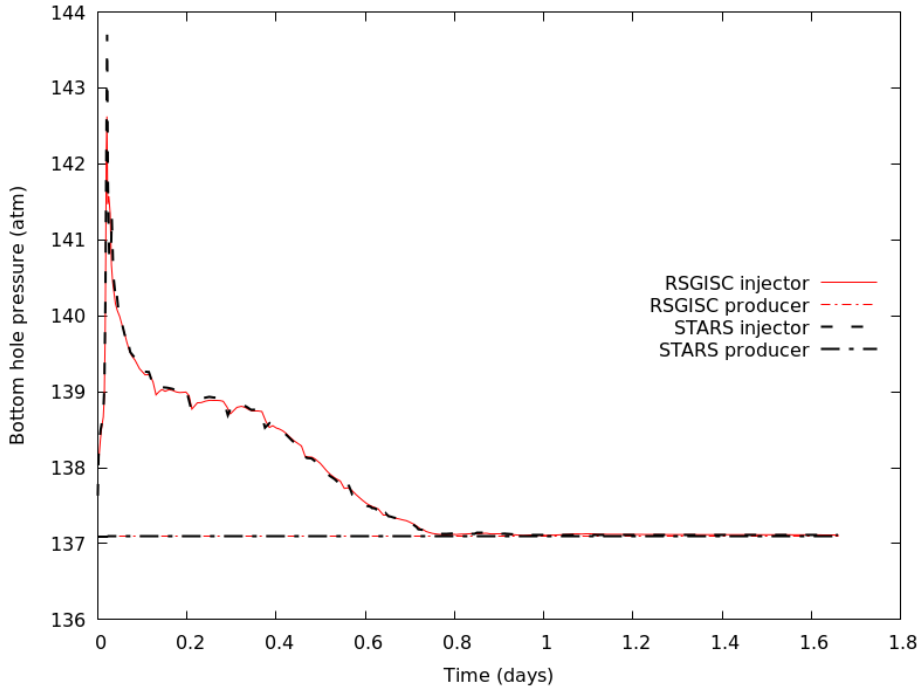

Figure 5.19: Bottom hole pressure profile with results from CMG STARS.

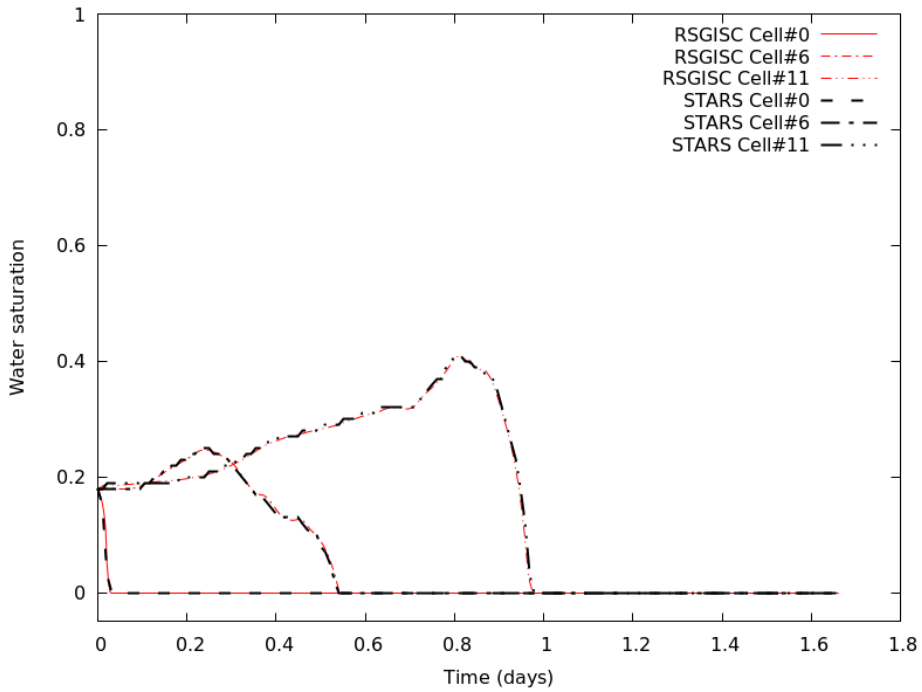

Figure 5.20: Water saturation profile with results from CMG STARS.

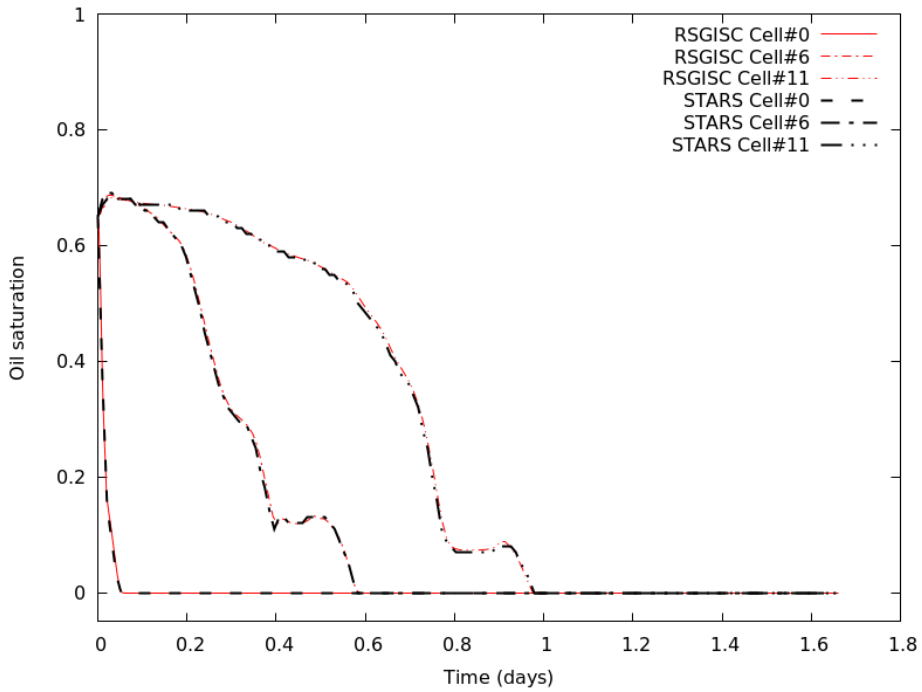

Figure 5.21: Oil saturation profile with results from CMG STARS.

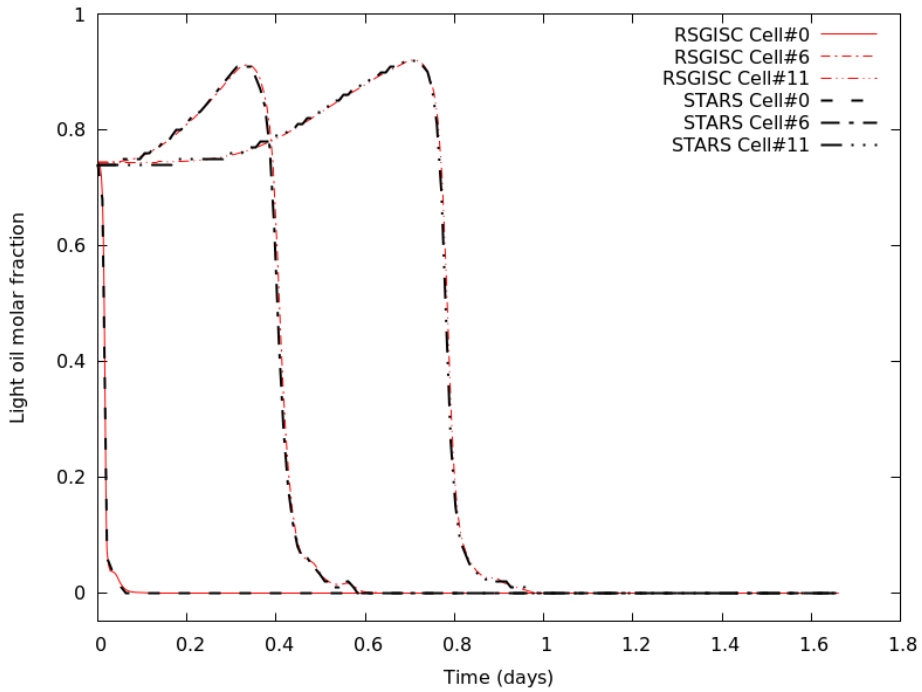

Figure 5.22: Oil molar fraction of light oil profile with results from CMG STARS.

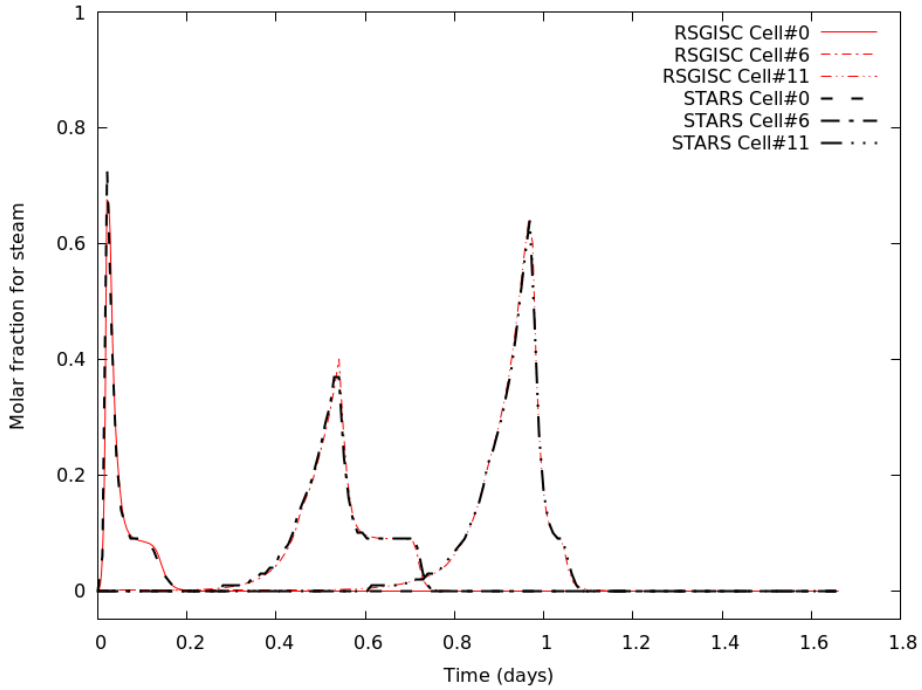

Figure 5.23: Gas molar fraction of steam profile with results from CMG STARS.

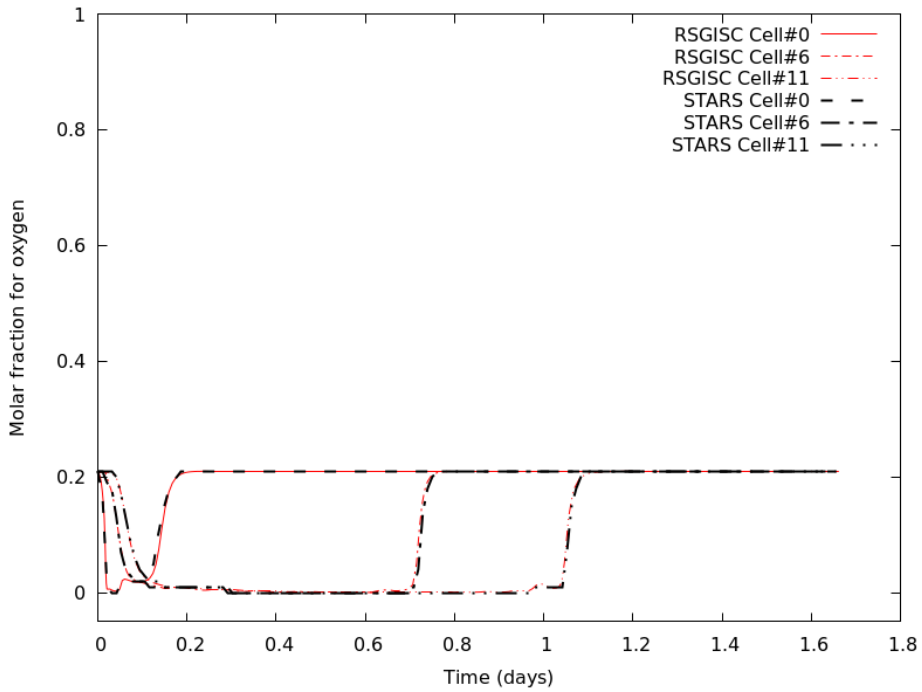

Figure 5.24: Gas molar fraction of oxygen profile with results from CMG STARS.

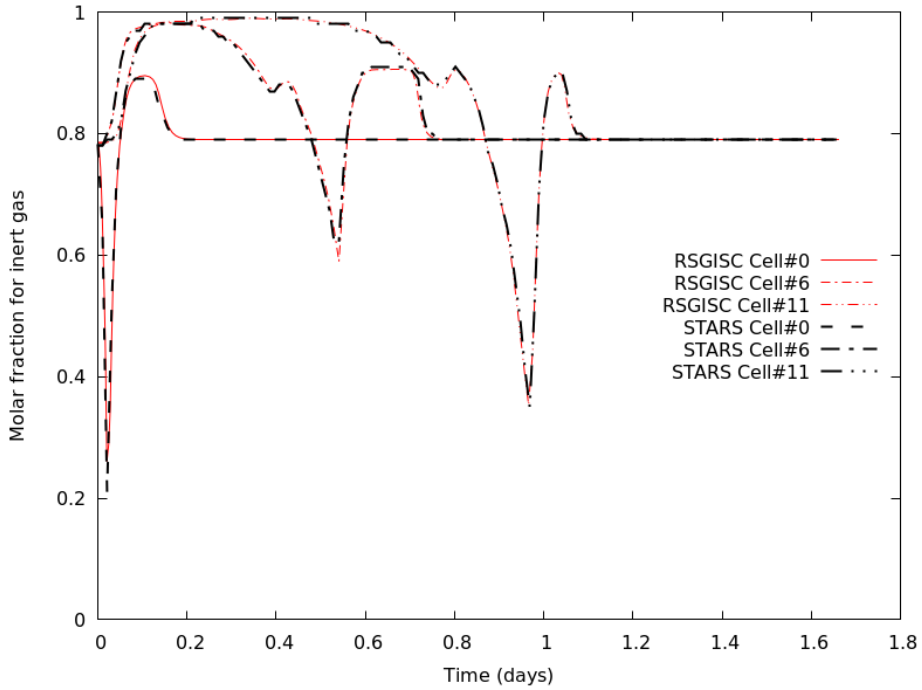

Figure 5.25: Gas molar fraction of inert gas profile with results from CMG STARS.

5.2 Field-scale in-situ combustion simulation

After a combustion tube experiment, a field-scale model is the task of the next step. We adopt the data from another set of data from different papers (Crookston, R. B., Culham, W. E., and Chen, W. H. (1979); Grabowski, J. W. et al. (1979); Vaughn, P. and Wyoming, L. (1986)). Historically used as a benchmark, this model is used to explain the validation of our simulator as well.

5.2.1 Model description

The reservoir itself is 164ft long, 115ft wide and 21ft high, in which the behavior of in-situ combustion in about one year is considered. In this model, pure oxygen is injected as the oxidizer. The oil has a density of 76.5 API at standard condition. The viscosity is 138.837 cp at initial condition. The components and corresponding reactions are different from the combustion tube model. All properties, reactions, and operations are listed as follows. All denotations remain the same meanings as the previous model.

Initial condition	
$k_{x,y,z}(md)$	4000, 4000, 4000
ϕ	0.38
$p(psi)$	65
$T(^\circ F)$	200
$S_{w,o,g}$	0.2, 0.5, 0.3
$x_{LO,HO}$	0.0 1.0
$y_{W,LO,HO,O2,IR}$	0.0, 0.0, 0.0, 1.0, 0.0
C_c	0

Table 5.6: Input data for field-scale model.

Properties	**H2O**	**LO**	**HO**	**O2**	**IR**	**Coke**
M $(lb/lbmole)$	18	44	170	32	44	13
p_{crit} (psi)	3206.2	615.9	264.6	730	1073	
T_{crit} $(^\circ F)$	705.73	205.93	725.23	-181.77	88.03	
$\rho_{ref}(lbmole/ft^3)$	3.47	0.4545	0.25			80.002
cp $(1/psi)$	1.00e-5	7.69e-4	3.80e-4			
$ct1$ $(1/^\circ F)$	3.80e-4	2.20e-4	1.00e-5			3.9
$cpg1$ $(Btu/(^\circ F \cdot lbmol))$	7.613	-52.0192	57.8	7.68	11	
$cpg2$ $(Btu/(^\circ F^2 \cdot lbmol))$	8.616e-4	1.519e-1	6.018e-2	0	0	
hvr $(Btu/(^\circ F^{ev} \cdot lbmol))$	1657	1917	12198			
ev	0.38	0.38	0.38			
avg $(cp/^\circ F)$	8.822e-6	2.166e-6	3.926e-6	2.196e-4	2.127e-4	
bvg	1.116	0.943	1.102	0.721	0.702	
$avisc$ (cp)	4.7352e-3	4.02e-4	4.02e-4			
$bvisc$ $(^\circ F)$	2728.2	6121.6	6121.6			
$kv1$ (psi)	1.7202e6	1.4546e5	2.7454e5			
$kv2$ $(1/psi)$	0	0	0			
$kv3$	0	0	0			
$kv4$ $(^\circ F)$	-6869.59	-4458.73	-8424.83			
$kv5$ $(^\circ F)$	-376.64	-387.78	-205.69			

Table 5.7: Input data for field-scale model (cont'd).

SLT table			
Sl	Krg	Krog	Pcg
0.29000	0.700000000	0.000000000	0.0
0.33125	0.656250000	0.000244141	0.0
0.37250	0.612500000	0.001953130	0.0
0.41375	0.568750000	0.006591800	0.0
0.45500	0.525000000	0.015625000	0.0
0.49625	0.481250000	0.030517600	0.0
0.53750	0.437500000	0.052734400	0.0
0.57875	0.393750000	0.083740200	0.0
0.62000	0.350000000	0.125000000	0.0
0.66125	0.306250000	0.177979000	0.0
0.70250	0.262500000	0.244141000	0.0
0.74375	0.218750000	0.324951000	0.0
0.78500	0.175000000	0.421875000	0.0
0.82625	0.131250000	0.536377000	0.0
0.86750	0.087500000	0.669922000	0.0
0.90875	0.043750000	0.823975000	0.0
0.95000	0.000000000	1.000000000	0.0

Table 5.8: Input data for field-scale model (cont'd).

SLT table			
SWT table			
Sw	Krw	Krow	Pcw
0.20000	0.000000000	1.000000000	0.0
0.23125	0.0000610352	0.823975000	0.0
0.26250	0.000488281	0.669922000	0.0
0.29375	0.001647950	0.536377000	0.0
0.32500	0.003906250	0.421875000	0.0
0.35625	0.007629390	0.324951000	0.0
0.38750	0.013183600	0.244141000	0.0
0.41875	0.020935100	0.177979000	0.0
0.45000	0.031250000	0.125000000	0.0
0.48125	0.044494600	0.083740200	0.0
0.51250	0.061035200	0.052734400	0.0
0.54375	0.081237800	0.030517600	0.0
0.57500	0.105469000	0.015625000	0.0
0.60625	0.134094000	0.006591800	0.0
0.63750	0.167480000	0.001953130	0.0
0.66875	0.205994000	0.000244141	0.0
0.70000	0.250000000	0.000000000	0.0

Table 5.9: Input data for field-scale model (cont'd).

Reaction data		
$LO + 5O_2 \longrightarrow 3IR + 4H_2O$		r_1
$HO + 18O_2 \longrightarrow 11.64IR + 13H_2O$		r_2
$HO \longrightarrow 2LO + 4.67Coke + 0.484IR$		r_3
$Coke + 1.25O_2 \longrightarrow 1IR + 0.5H_2O$		r_4
A_R $(1/(psi \cdot day))$		1e6, 1e6, 0.3e6, 1e6
E_{aR} $(Btu/lbmol)$		3.33e4, 3.33e4, 2.88e4, 2.34e4
H_R $(Btu/lbmol)$		9.48e5, 3.49e6, 2.0e4, 2.25e5
Well conditions		
Injector	gas rate (O2:N2 = 1:0) (ft^3/hr)	115000
	rc (ft)	0.5
	pinjw (psi)	10000
	tinjw $(^\circ F)$	200
Producer	bhp (psi)	65

Table 5.10: Input data for field-scale model (cont'd).

5.2.2 Physical behavior and validation

From the combustion tube to a field-scale model, here we focus on the differences in physical behavior.

Different from the combustion tube model, there is an obvious period of the water phase to evaporate. When we focus on Figure 5.35, we can observe that the temperature is stable around 425 K, which is about the boiling point of the water phase. At this time, the water phase kept evaporating, which is shown in Figure 5.30. However, in this model, only the heavy oil component exists in the oil phase originally, which is harder to evaporate than the water component. Thus, when the water phase changes to steam, at a temperature lower than its boiling point, the vapor of the heavy oil component, which flows from the upstream, kept condensing, resulting in an increasing oil saturation, which can be seen in Figure 5.29. Also, steam takes the most portion in the gas phase, and coke is produced at a certain rate under a steady temperature (Figure 5.31, 5.32).

When the water phase disappears, everything changes dramatically. First of all, the heat conducted from the upstream no longer needs to support the heat of vaporization of water, and the temperature in a certain cell begins to increase rapidly. Thus the rate of all chemical reactions increases exponentially. When the ignition point of coke is reached, the coke is consumed rapidly, resulting in a rapid coke burning reaction and leading to an exponentially-increasing temperature (Figure 5.27, 5.30, 5.32). Therefore, the oil phase evaporates instantly at the high temperature, and the coke concentration reached the highest peak for the accelerated cracking reaction. For extremely high reaction rate, as the main product of all reactions, the molar fraction of steam even reached a small peak (Figure 5.29,

5.31, 5.32).

After oil also evaporates, only gas and coke take the pore volume in the reservoir. After coke is consumed, there is no longer any fuel in the cells, the violent reactions end and temperature begins to decrease for the low-temperature injected gas and heat loss.

This procedure continues to occur cell by cell, resulting in the propagation of the combustion zone. In addition, for the violent physical and chemical process, the change in pressure at each cell is very violent. For the conduction by the gas phase, several peaks of pressure can be observed in Figure 5.28.

In Figure 5.26, similar to last section, all zones in around the combustion front are easy to identify.

Also, the validations of some physical properties with CMG STARS are displayed in Figures 5.35 to 5.40.

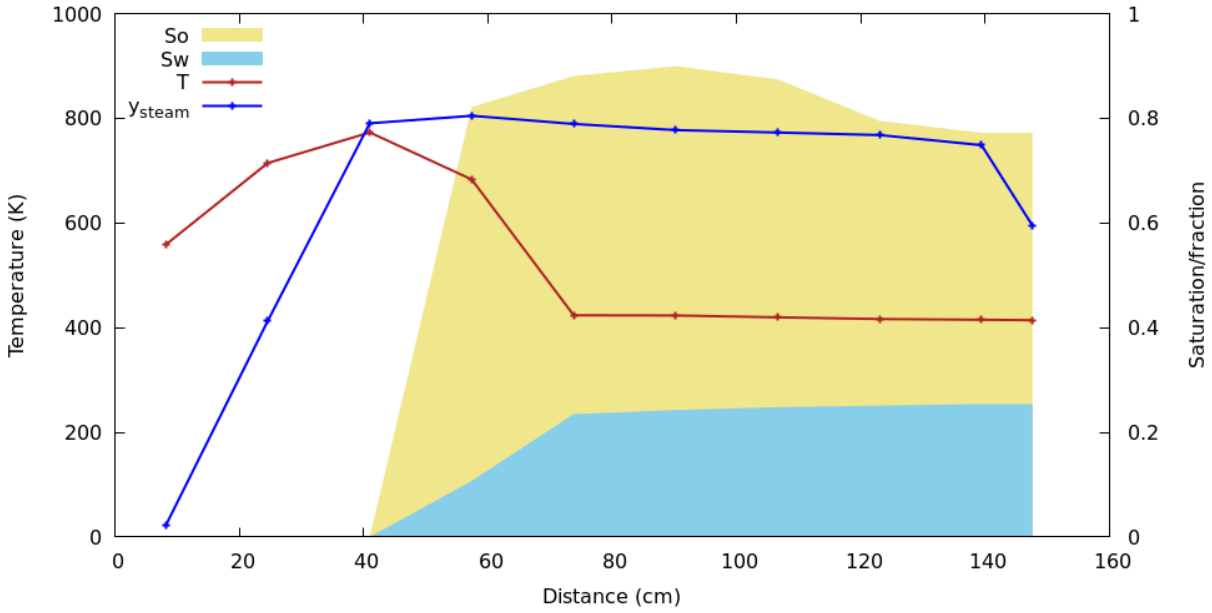

Figure 5.26: Temperature, saturation, steam fraction profile over the model after t = 30 days.

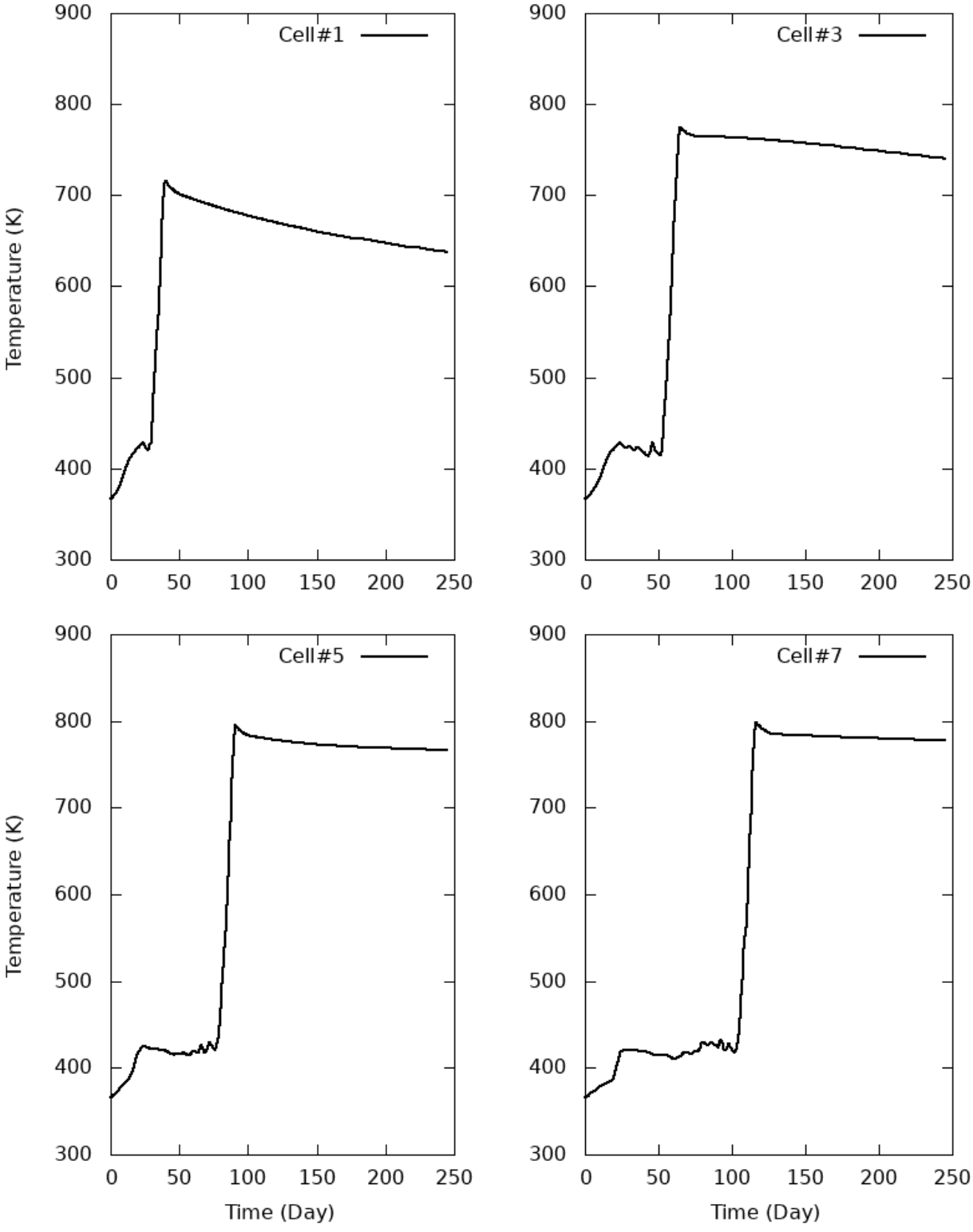

Figure 5.27: Temperature profile for field-scale model.

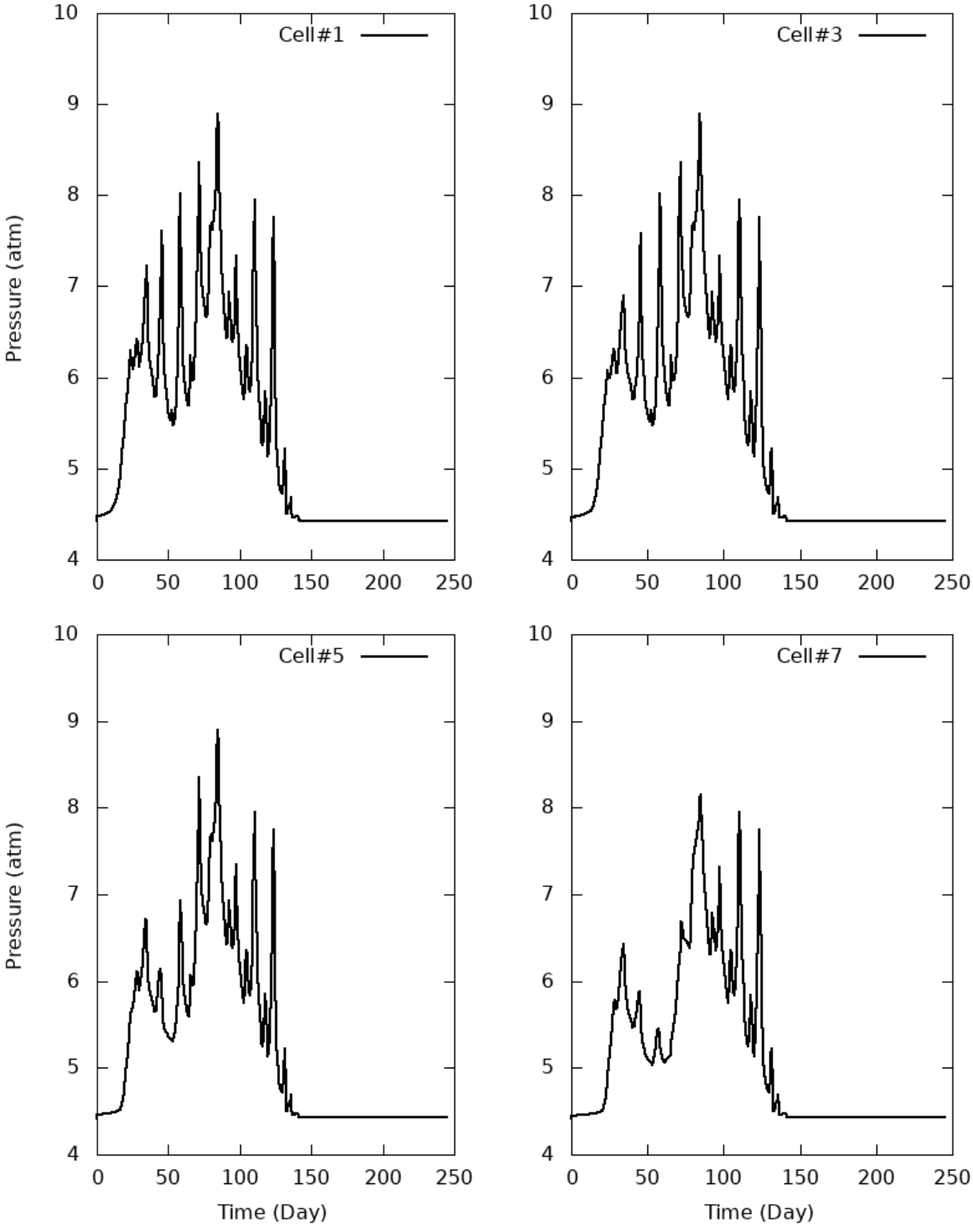

Figure 5.28: Pressure profile for field-scale model.

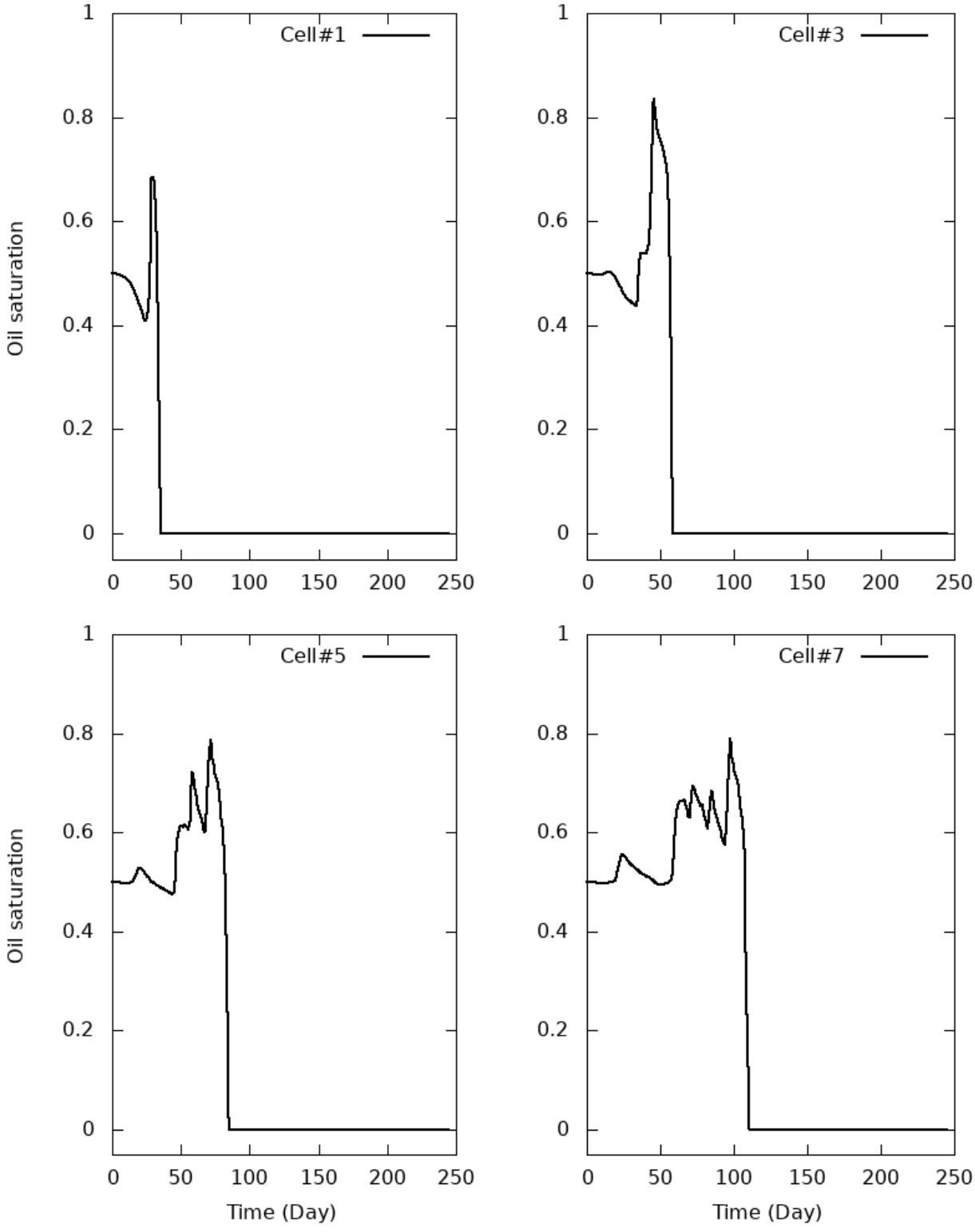

Figure 5.29: Oil saturation profile for field-scale model.

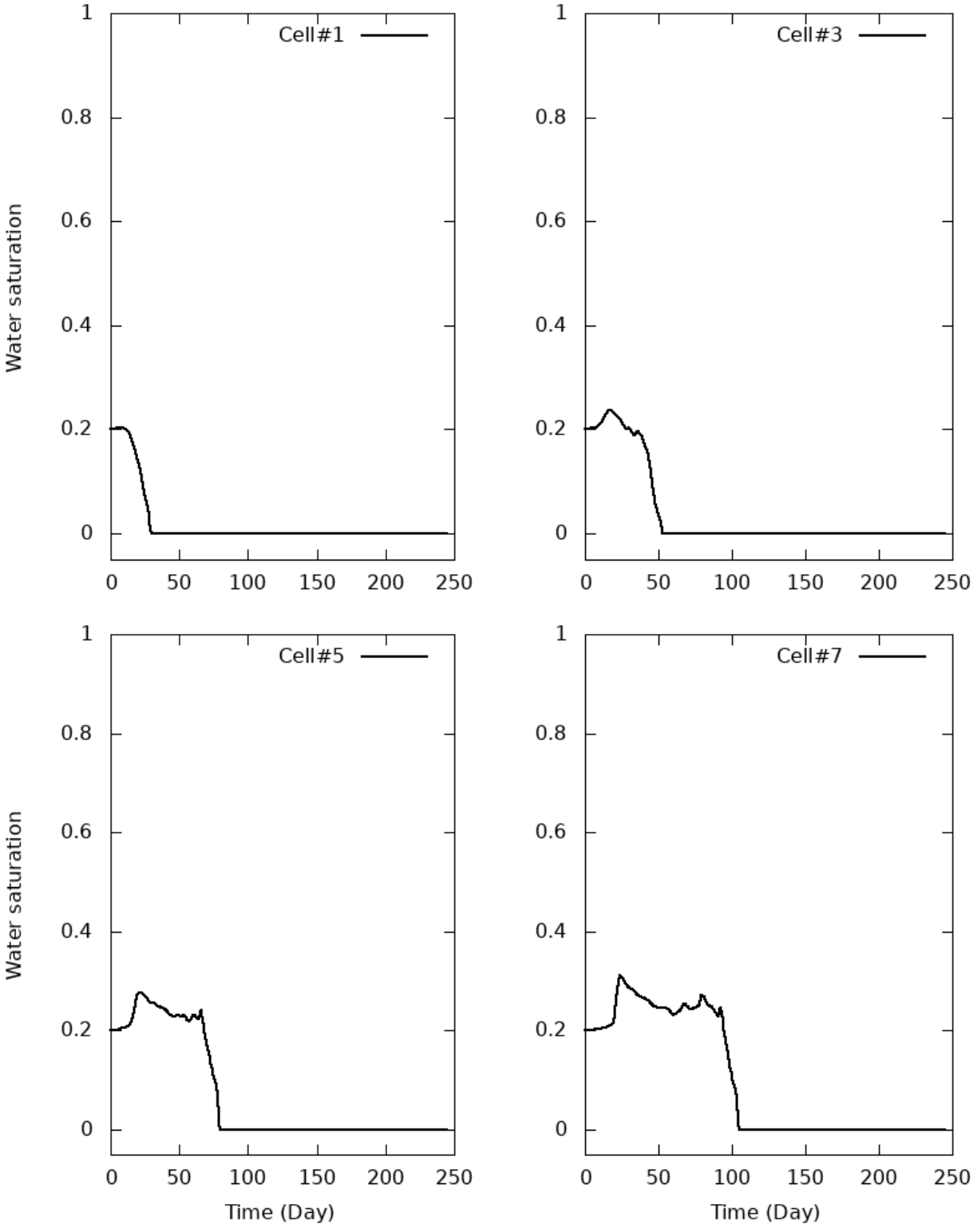

Figure 5.30: Water saturation profile for field-scale model.

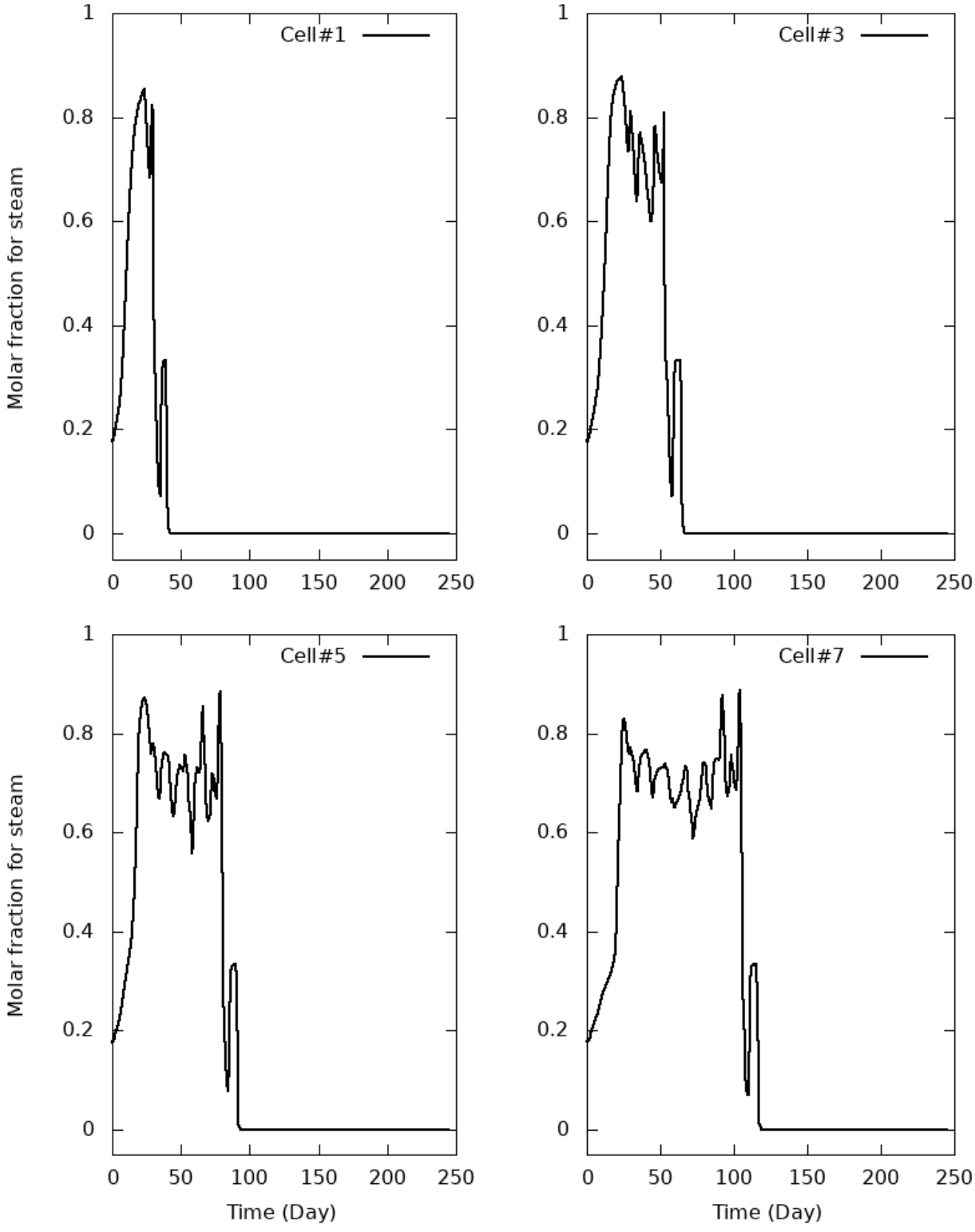

Figure 5.31: Gas molar fraction profile of steam for field-scale model.

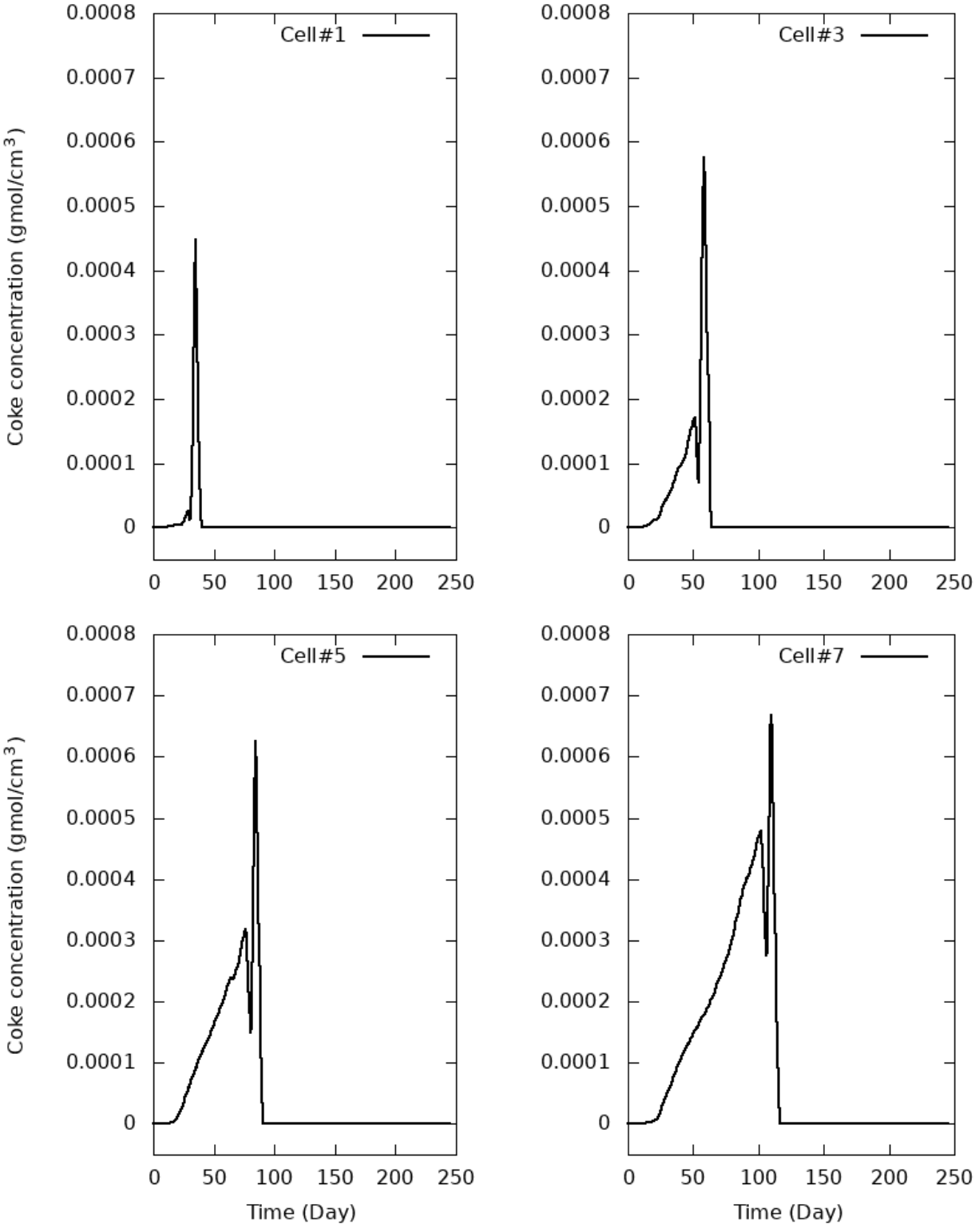

Figure 5.32: Coke concentration profile for field-scale model.

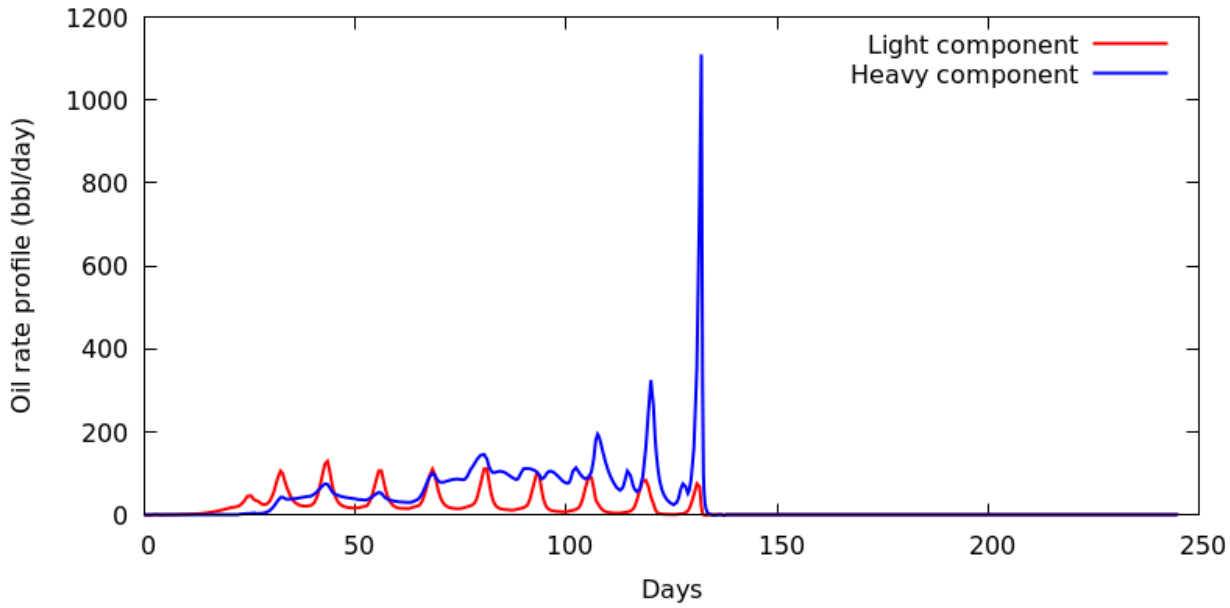

Figure 5.33: Oil rate profile for field-scale model.

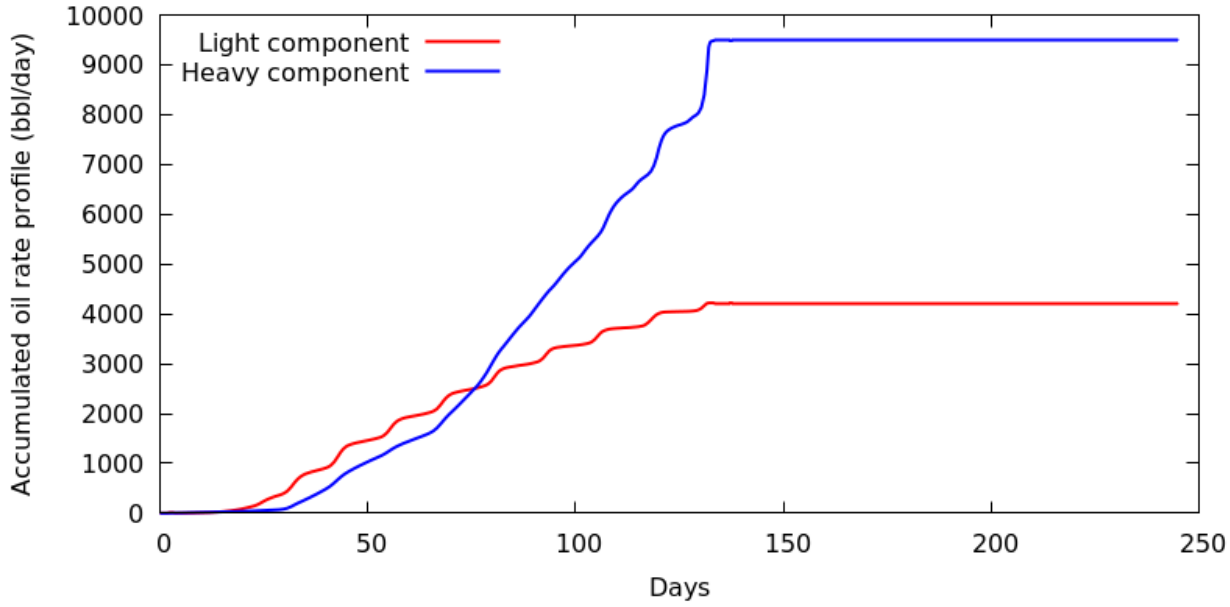

Figure 5.34: Accumulated oil rate profile for field-scale model.

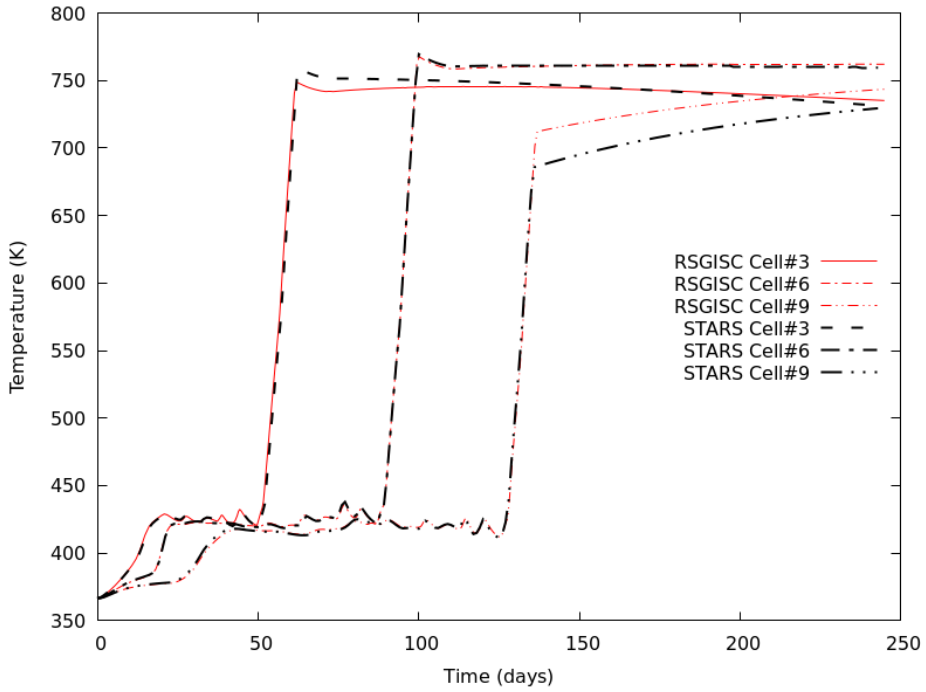

Figure 5.35: Temperature profile for field-scale model with results from CMG STARS.

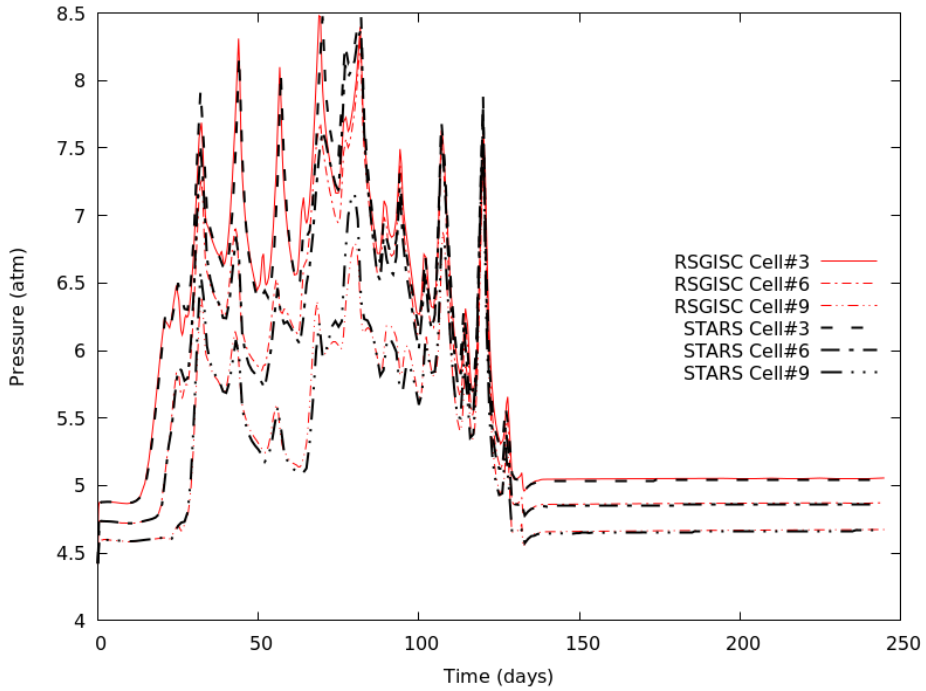

Figure 5.36: Pressure profile for field-scale model with results from CMG STARS.

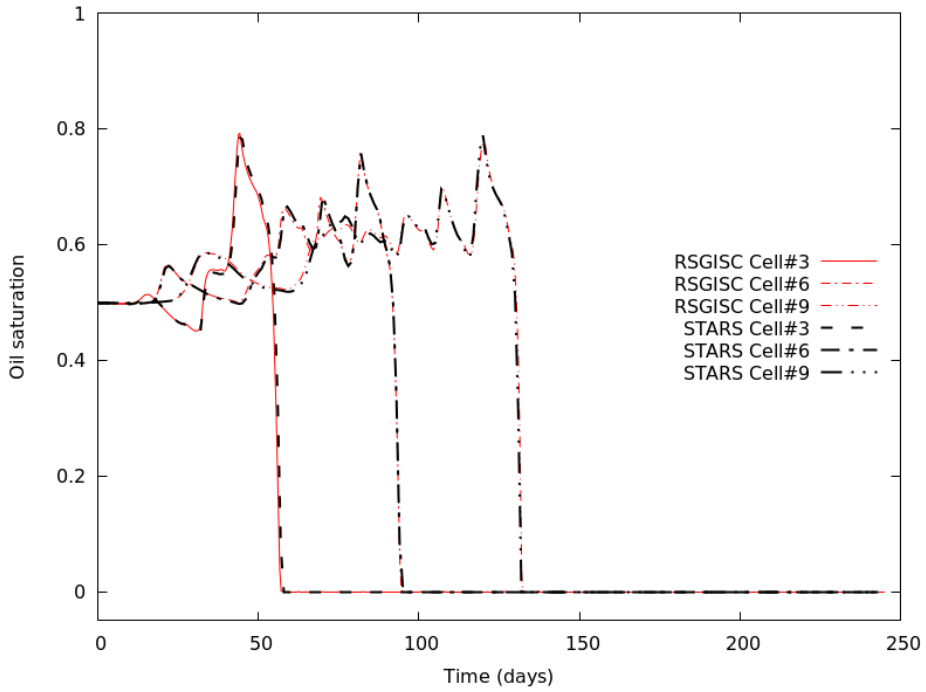

Figure 5.37: Oil saturation profile for field-scale model with results from CMG STARS.

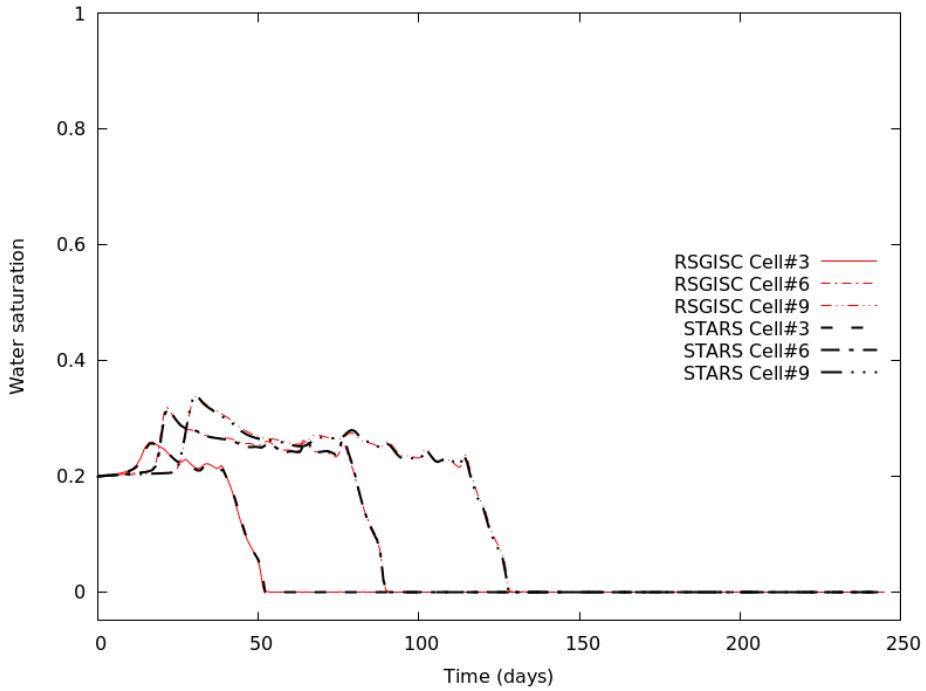

Figure 5.38: Water saturation profile for field-scale model with results from CMG STARS.

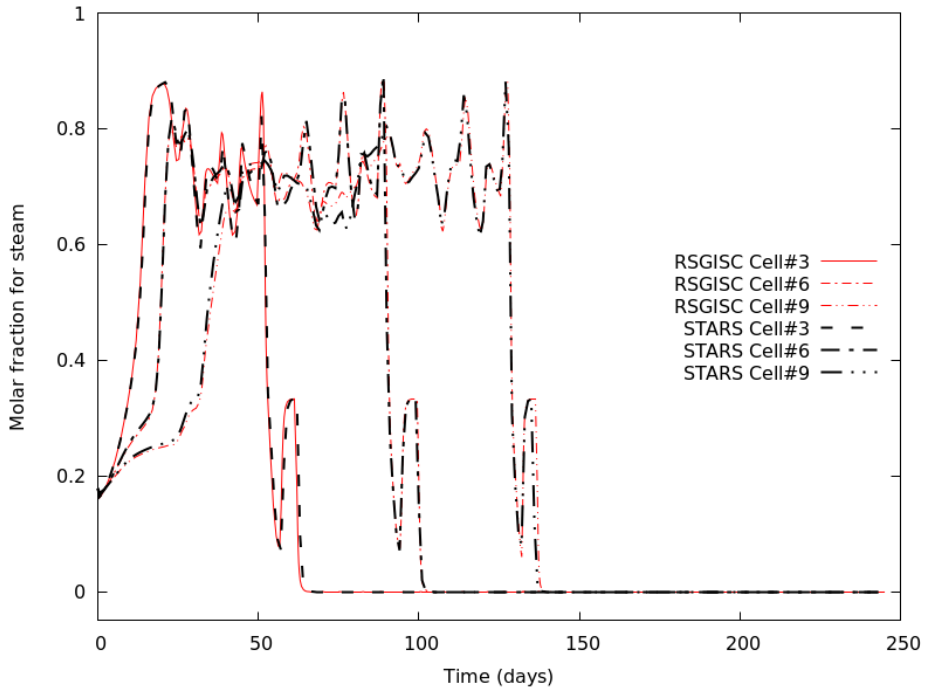

Figure 5.39: Gas molar fraction profile of steam for field-scale model with results from CMG STARS.

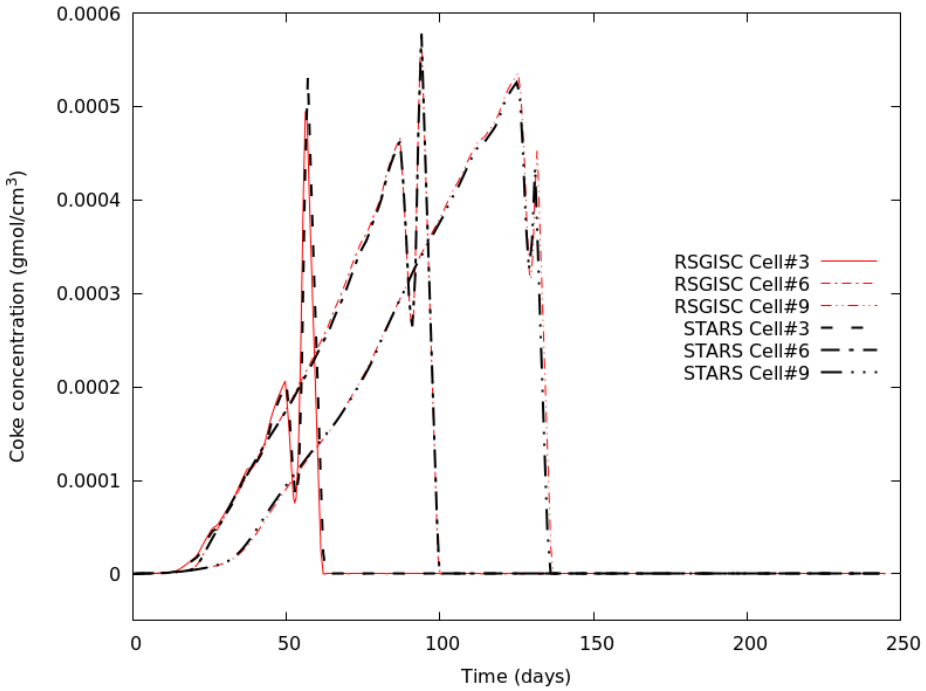

Figure 5.40: Coke concentration profile for field-scale model with results from CMG STARS.

5.2.3 Finer grid

In the previous subsection, a 1-D field scale model is presented. Here we consider extending this model into a 3-D model. We keep the division in the x-direction, and divide the edge in the y-direction into five parts and the edge in the y-direction into three parts. Therefore, totally 150 cells are included in the model. Compared to the 1-D model, a finer grid is applied. Two wells, the injector and the producer, are located at the centers of the two ends.

Results of the 3-D model are provided in Figures 5.41 to 5.48. In general, the temperature and oil saturation distribution at 60, 120, 180, and 240 days are given. In each figure, three layers in the reservoir are presented, which are the top layer on the left, the middle layer at the center and the bottom layer on the right.

Compared with the 1-D model, more detailed information is included in the 3-D model.

First of all, at the bottom layer, the oil phase concentrates, leading to a higher oil saturation, where all reactions mainly occur and displacement of the oil phase mainly happens. Thus in-situ combustion mainly happens in the lower part of the reservoir, which appears a high-temperature combustion front. For the heat conduction and convection, the temperature of the higher part in the reservoir is also lifted. But the temperature is not as high as in the lower part.

In the horizontal direction, propagation at different locations differ. Generally, the combustion front advances in a shape of a cone. At the very front of the injection well, the combustion front moves the fastest. In addition, the position in the very front of the injection well is much earlier to reach a breakthrough of gas.

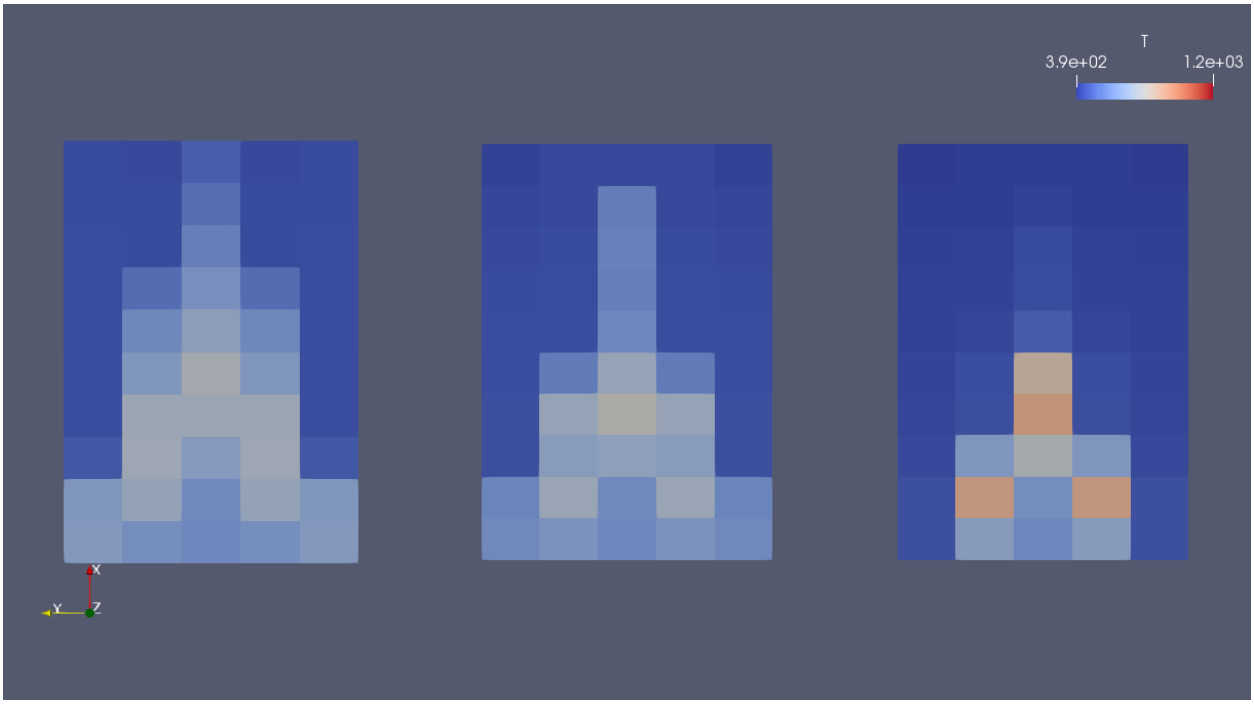

Figure 5.41: Temperature distribution on 3 layers after t = 60 days in 3-D model.

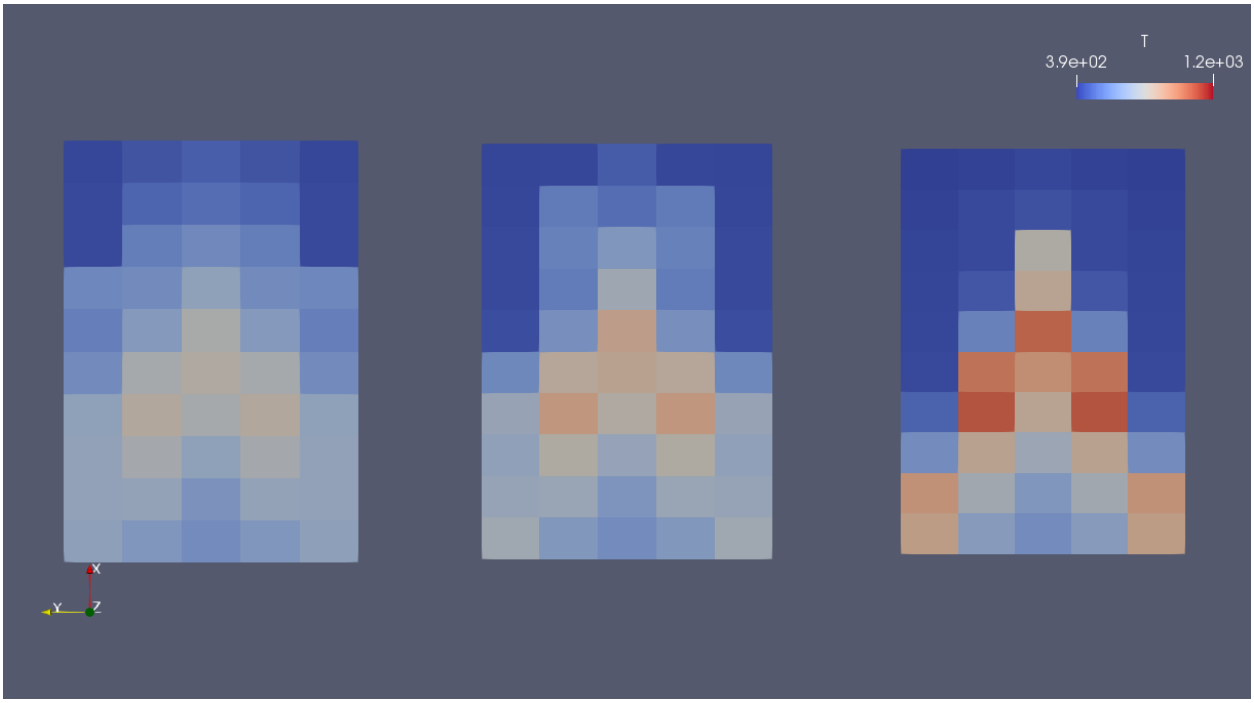

Figure 5.42: Temperature distribution on 3 layers after t = 120 days in 3-D model.

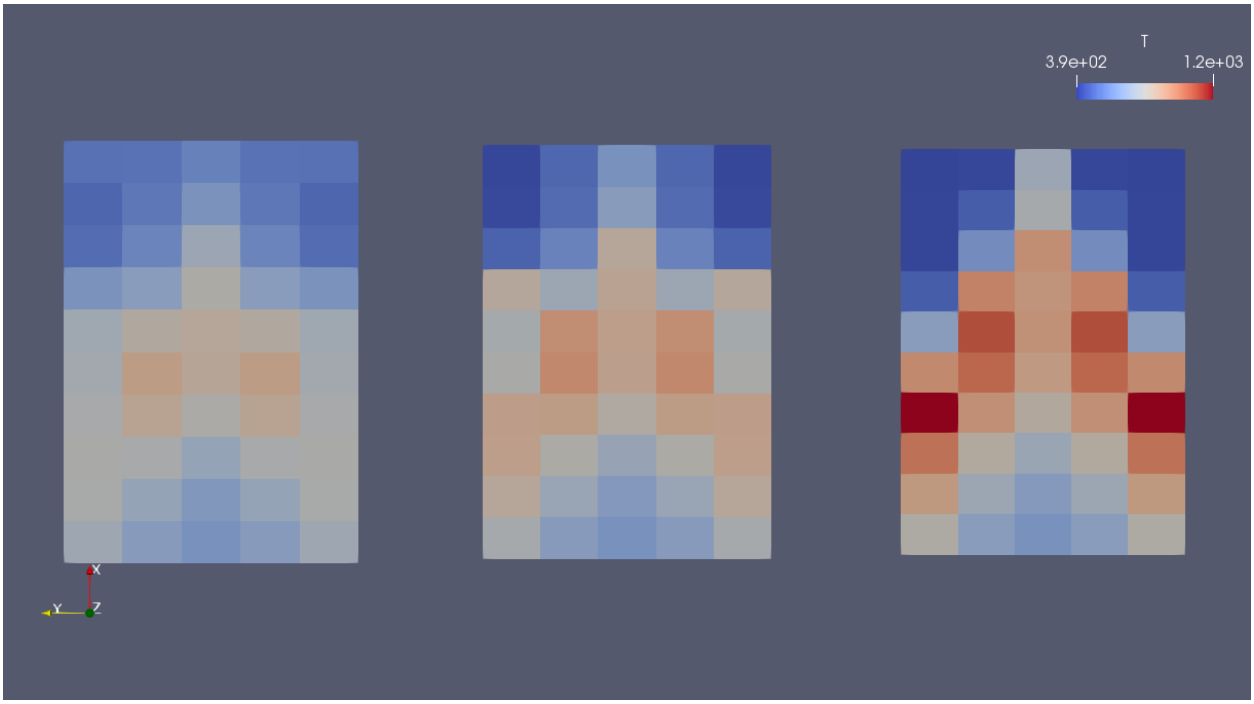

Figure 5.43: Temperature distribution on 3 layers after t = 180 days in 3-D model.

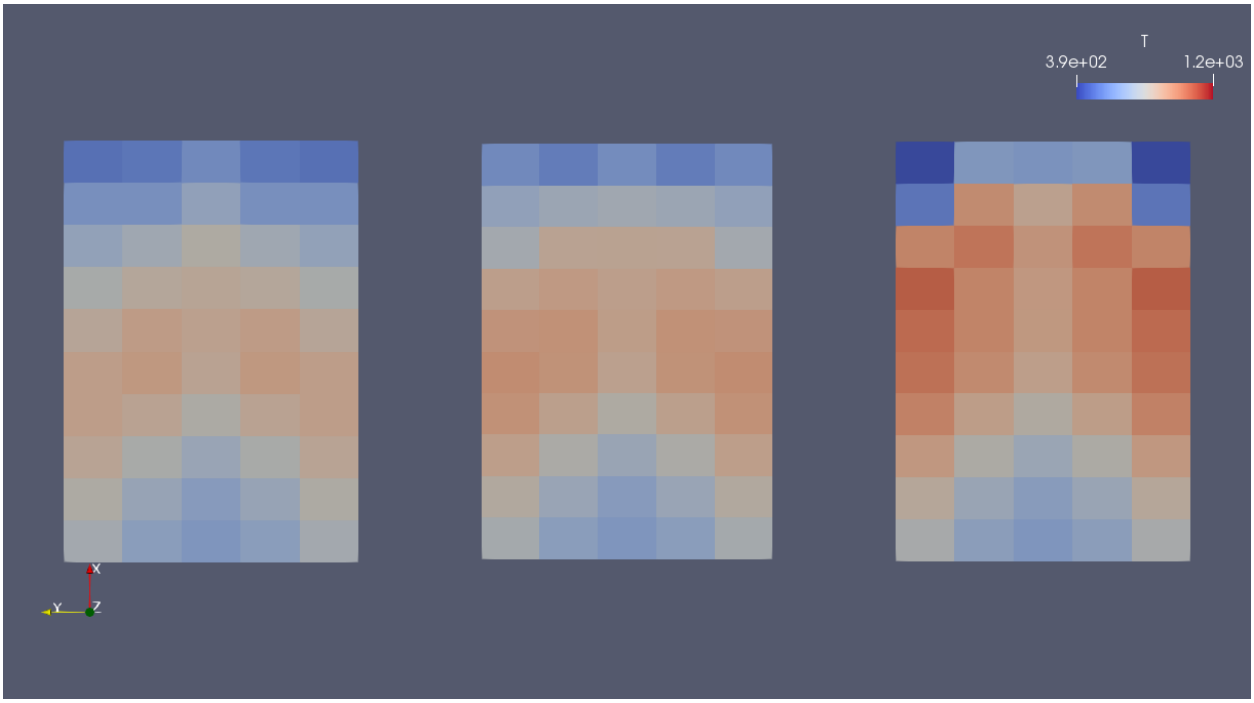

Figure 5.44: Temperature distribution on 3 layers after t = 240 days in 3-D model.

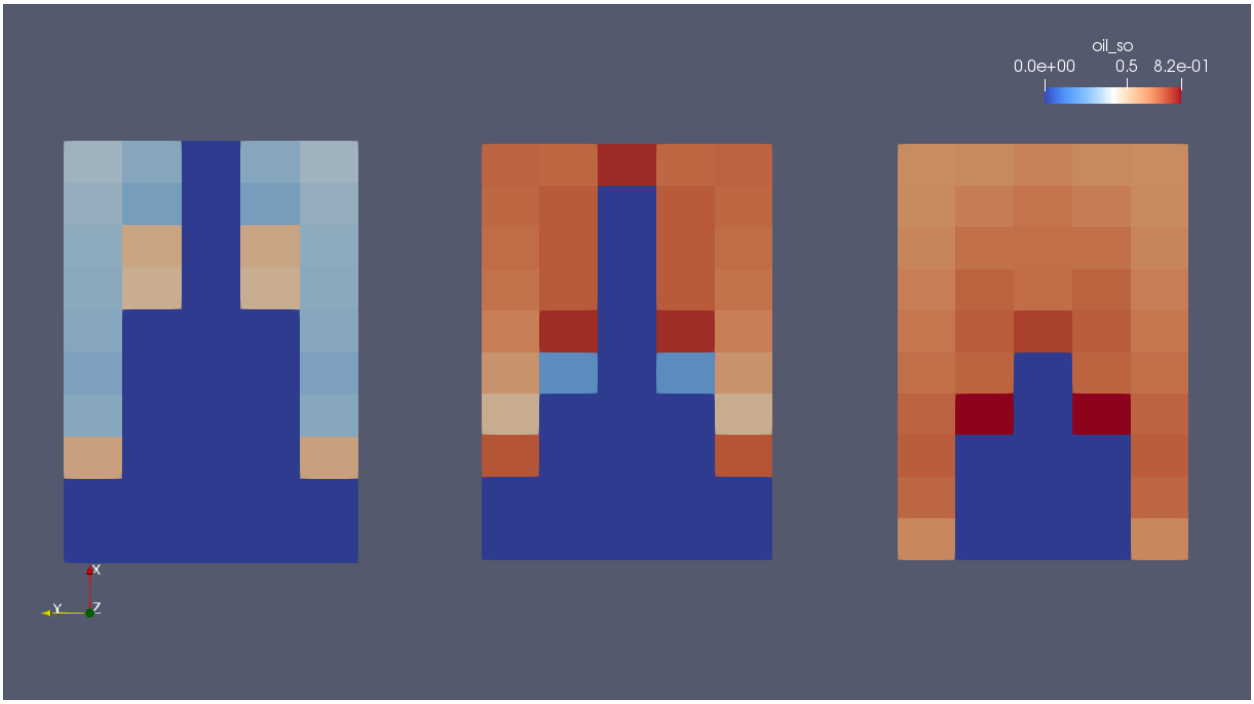

Figure 5.45: Oil saturation distribution on 3 layers after t = 60 days in 3-D model.

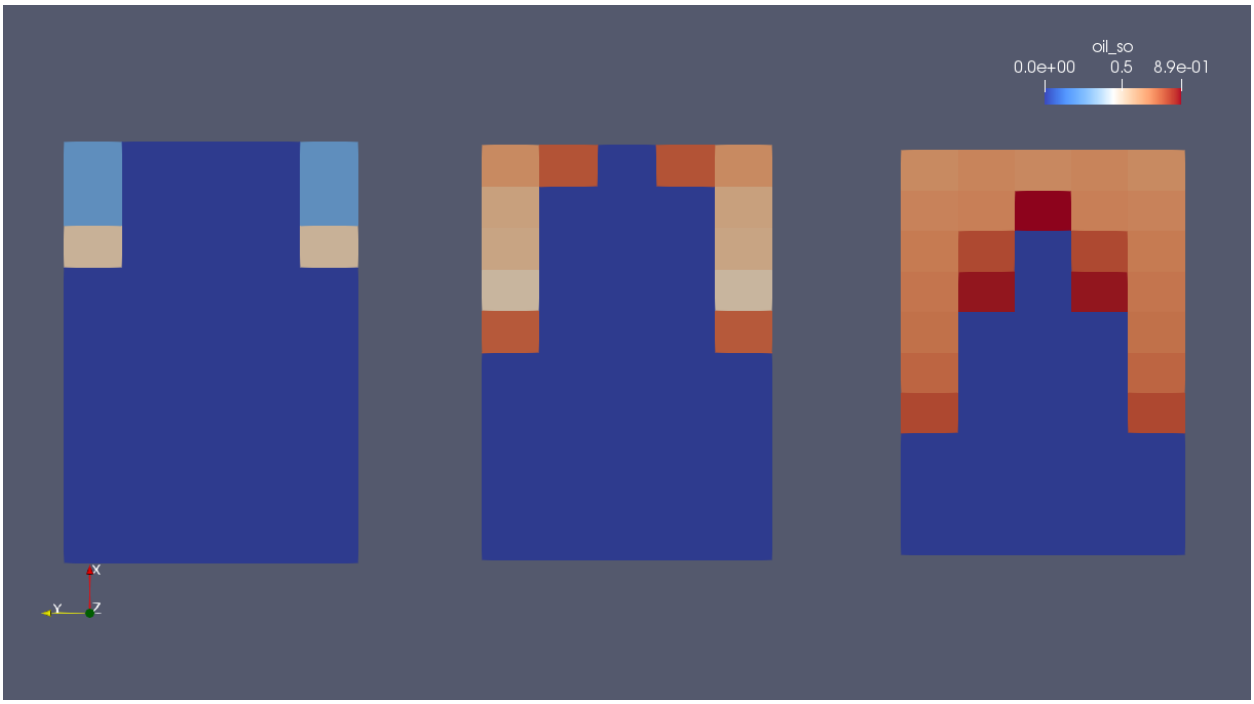

Figure 5.46: Oil saturation distribution on 3 layers after t = 120 days in 3-D model.

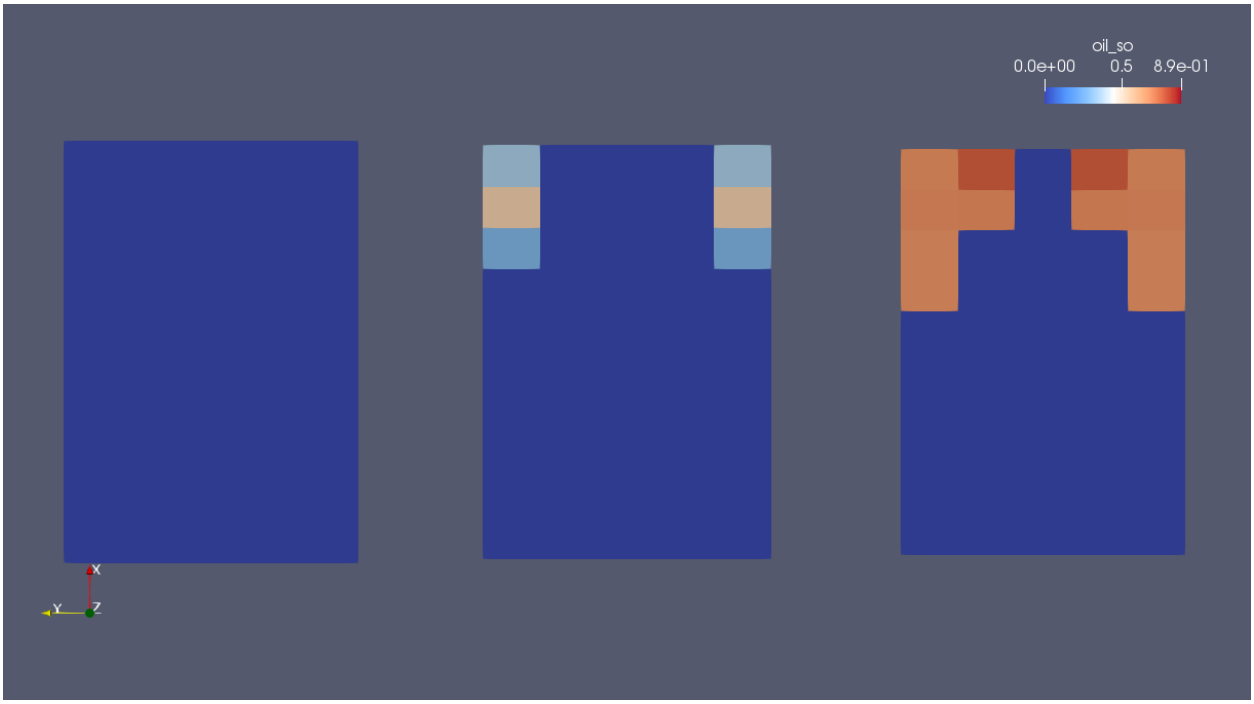

Figure 5.47: Oil saturation distribution on 3 layers after t = 180 days in 3-D model.

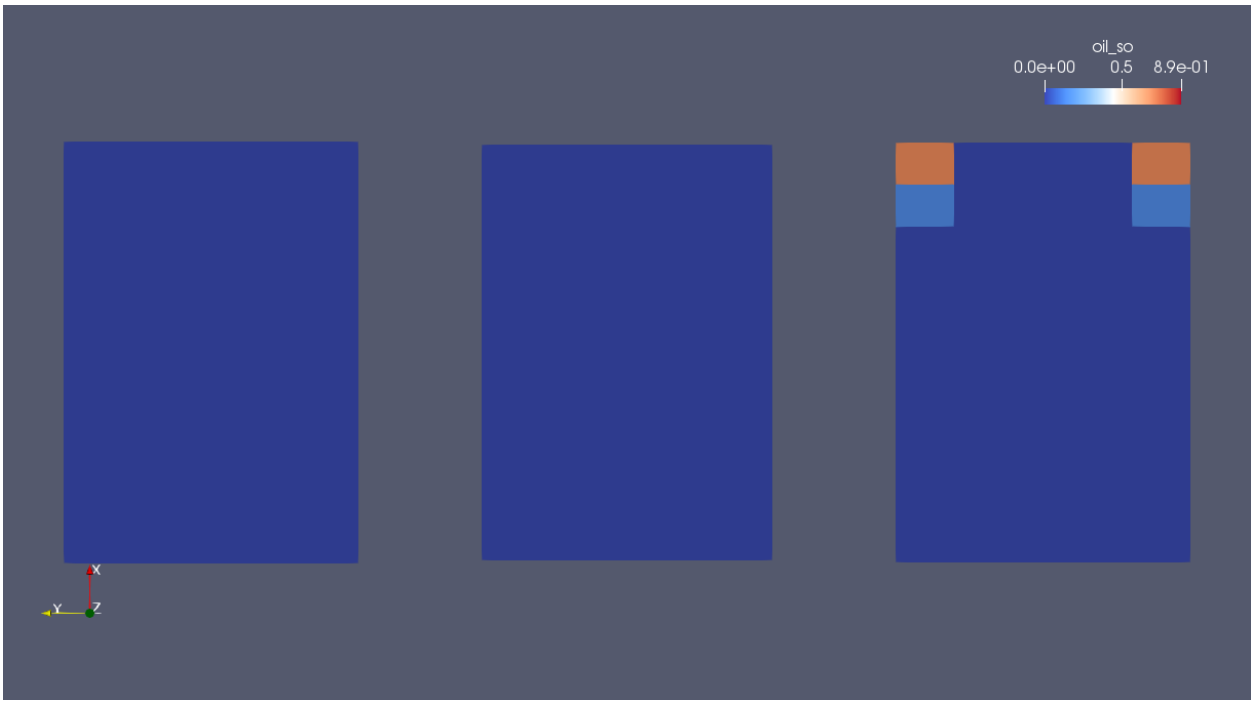

Figure 5.48: Oil saturation distribution on 3 layers after t = 240 days in 3-D model.

If we adopt an even finer grid to this model, the subtle physical behavior becomes even clearer. The total grid size is $20 \times 23 \times 7$, and two wells are still located at the center of each end. All conditions and other input remain the same. All the results are shown from Figures 5.49 to 5.56, in which temperature and oil saturation of three different layers are displayed.

In these figures, the characteristics mentioned above can be observed more clearly. The cone-shaped combustion front can be seen through both the temperature and oil saturation profiles.

Besides, it can be noticed that the peak temperature of the combustion zone reaches around 1000 K in the 3-D model. In the 1-D model, however, the peak temperature is only about 800 K. Therefore, with the finer grid, the temperature of the combustion front is much higher. With a coarse grid, the temperature of the combustion zone can even be lower than the ignition temperature, leading to a failure of extinguished combustion, which is not the case in reality. It is all because of numerical errors. Thus applying a fine grid can be essential in a typical simulator of in-situ combustion. Different from applying a dynamic grid and other approaches, parallel implementation is applied to fine-grid models. In the next section, we will discuss the performance of parallelization.

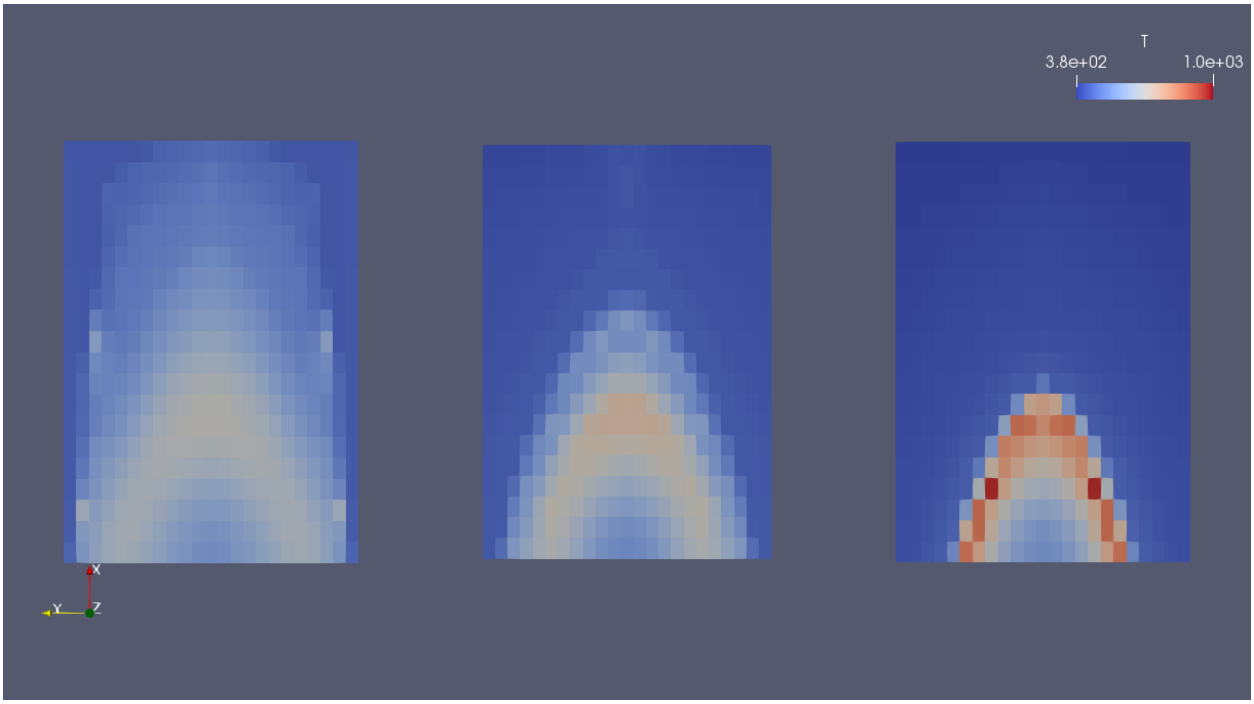

Figure 5.49: Temperature distribution on 3 layers after t = 60 days on finer grid.

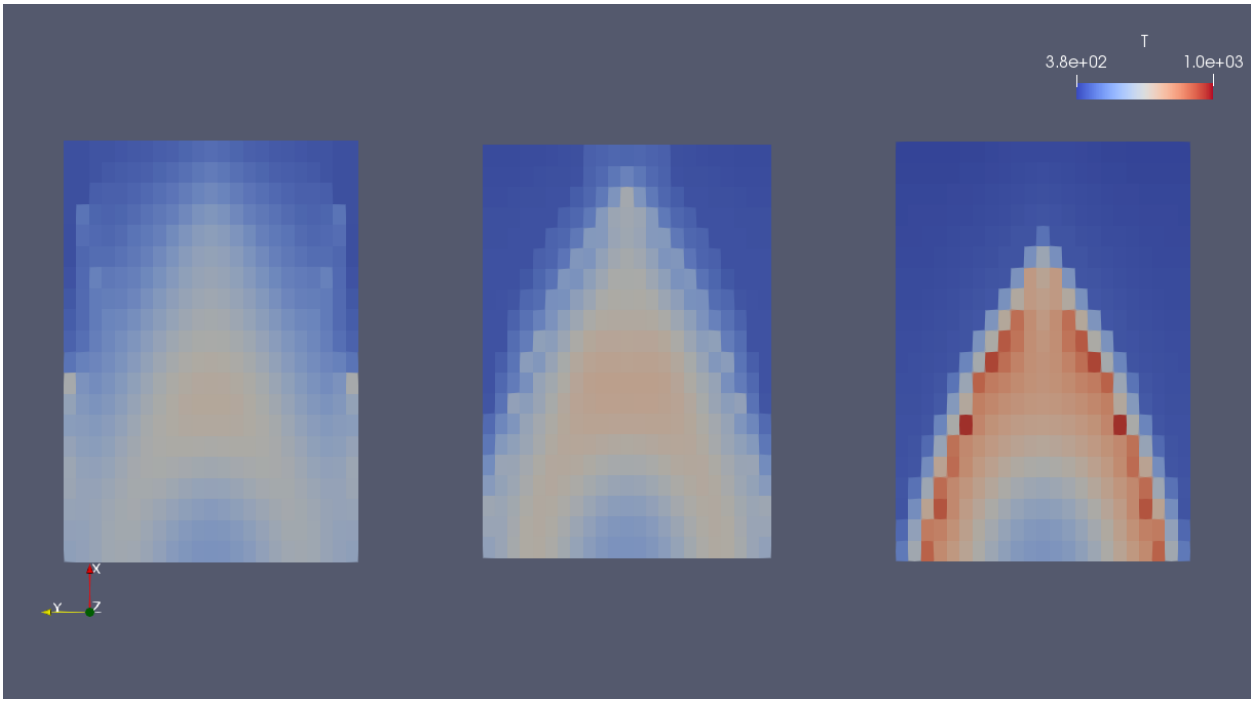

Figure 5.50: Temperature distribution on 3 layers after t = 120 days on finer grid.

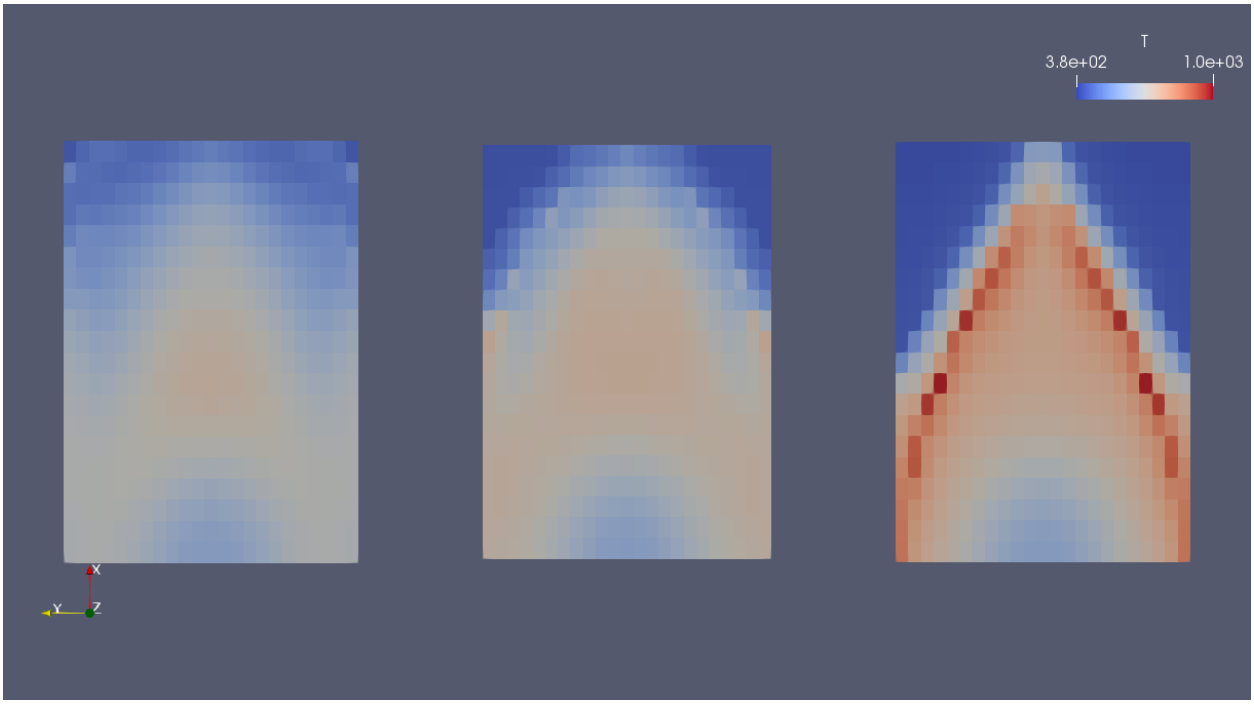

Figure 5.51: Temperature distribution on 3 layers after t = 180 days on finer grid.

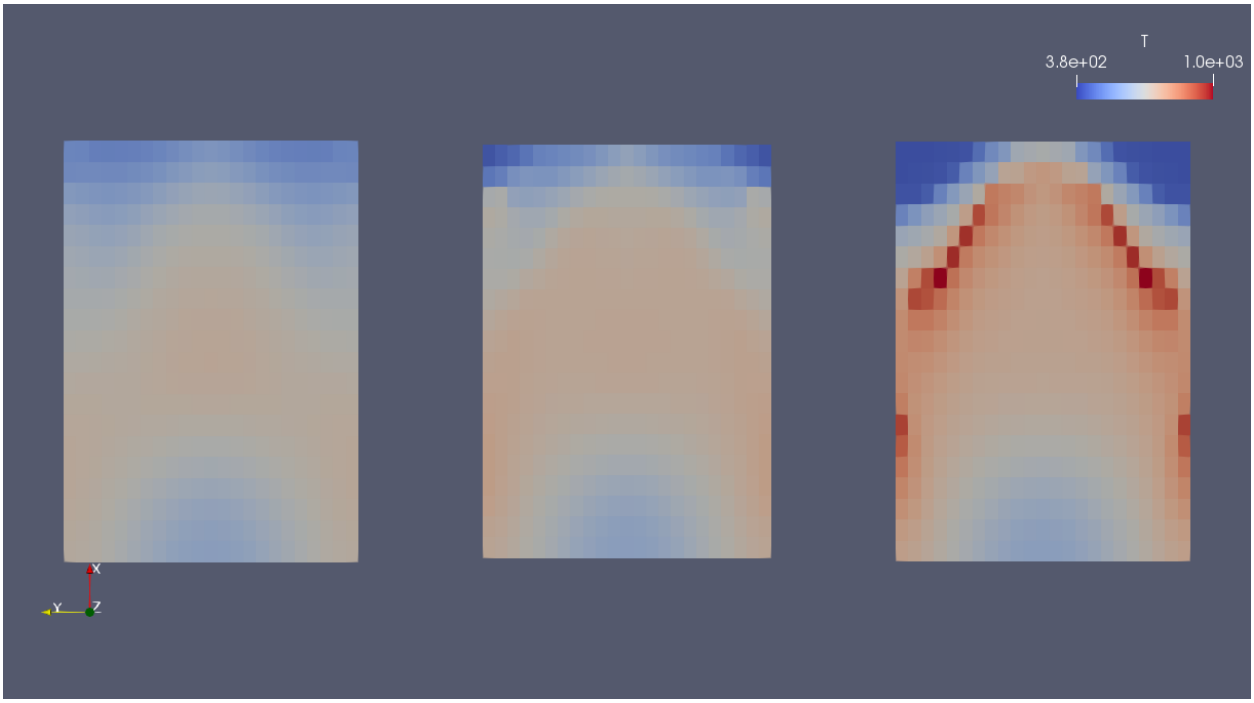

Figure 5.52: Temperature distribution on 3 layers after t = 240 days on finer grid.

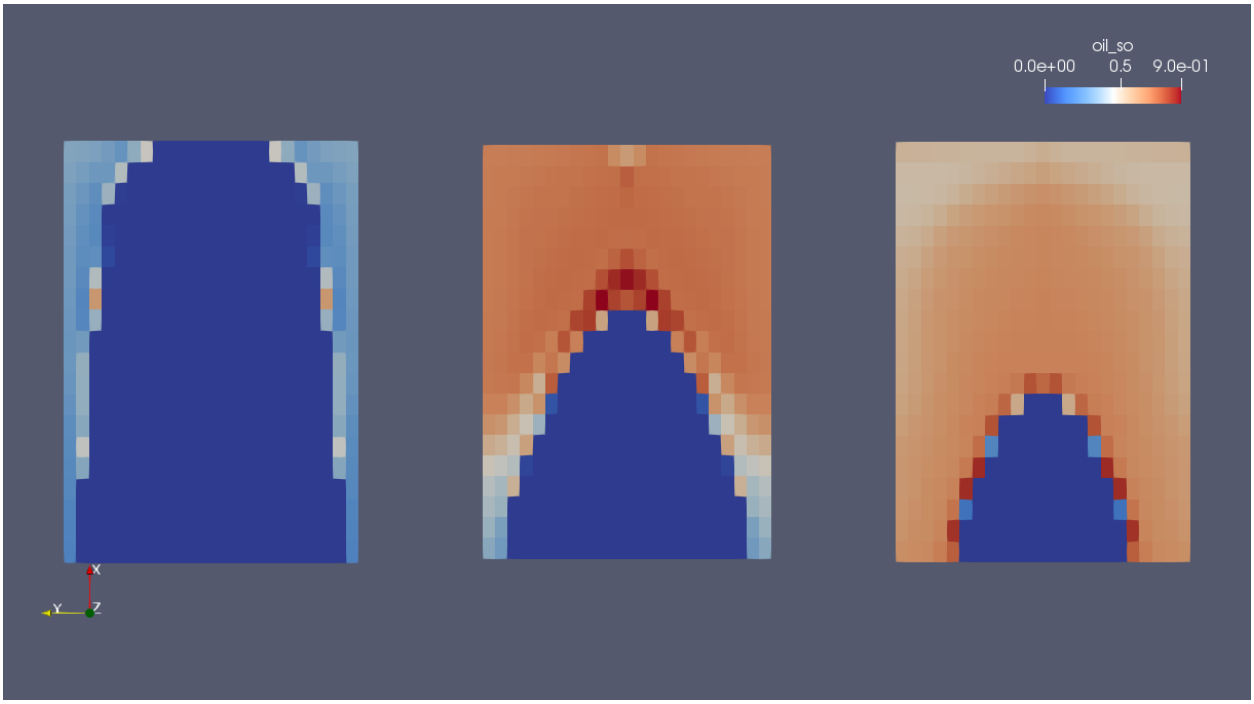

Figure 5.53: Oil saturation distribution on 3 layers after t = 60 days on finer grid.

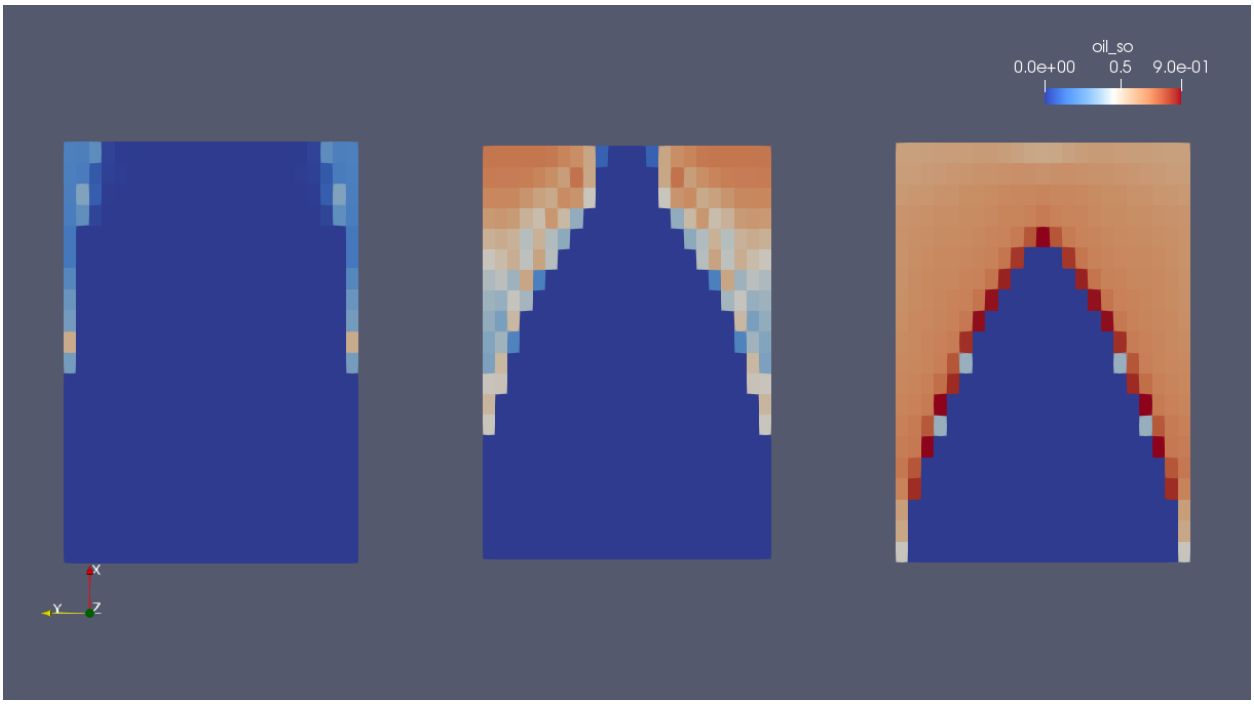

Figure 5.54: Oil saturation distribution on 3 layers after t = 120 days on finer grid.

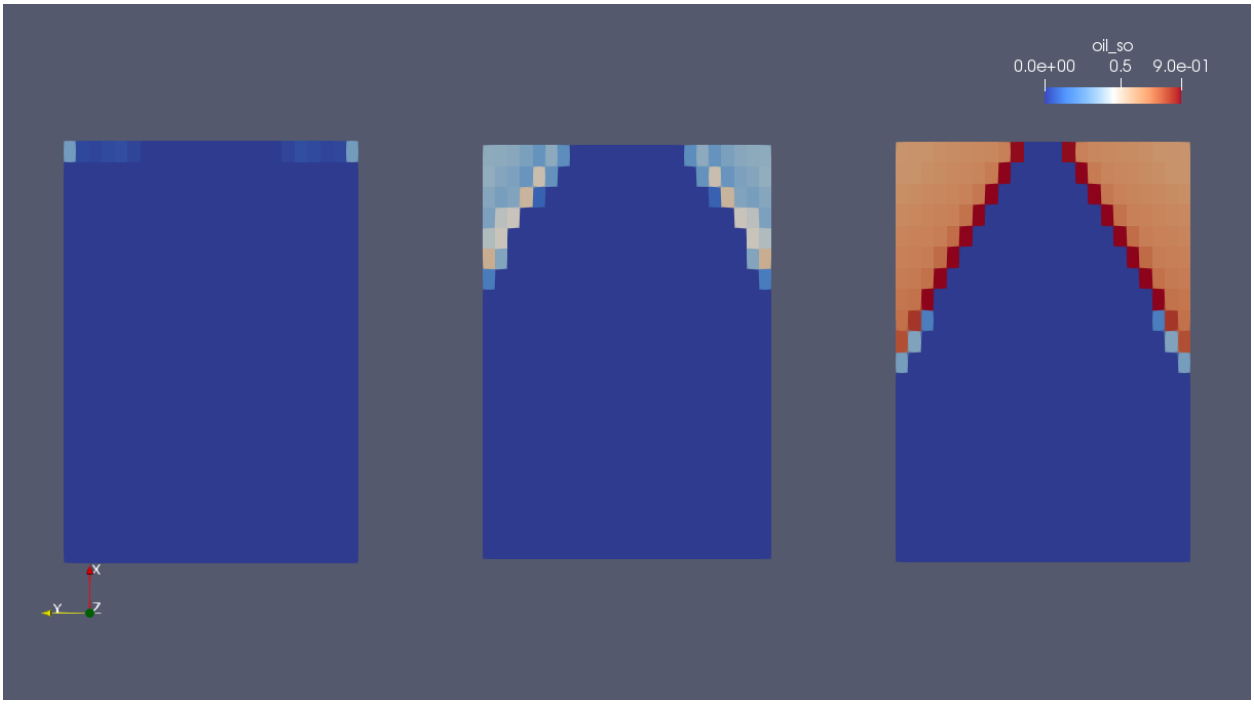

Figure 5.55: Oil saturation distribution on 3 layers after t = 180 days on finer grid.

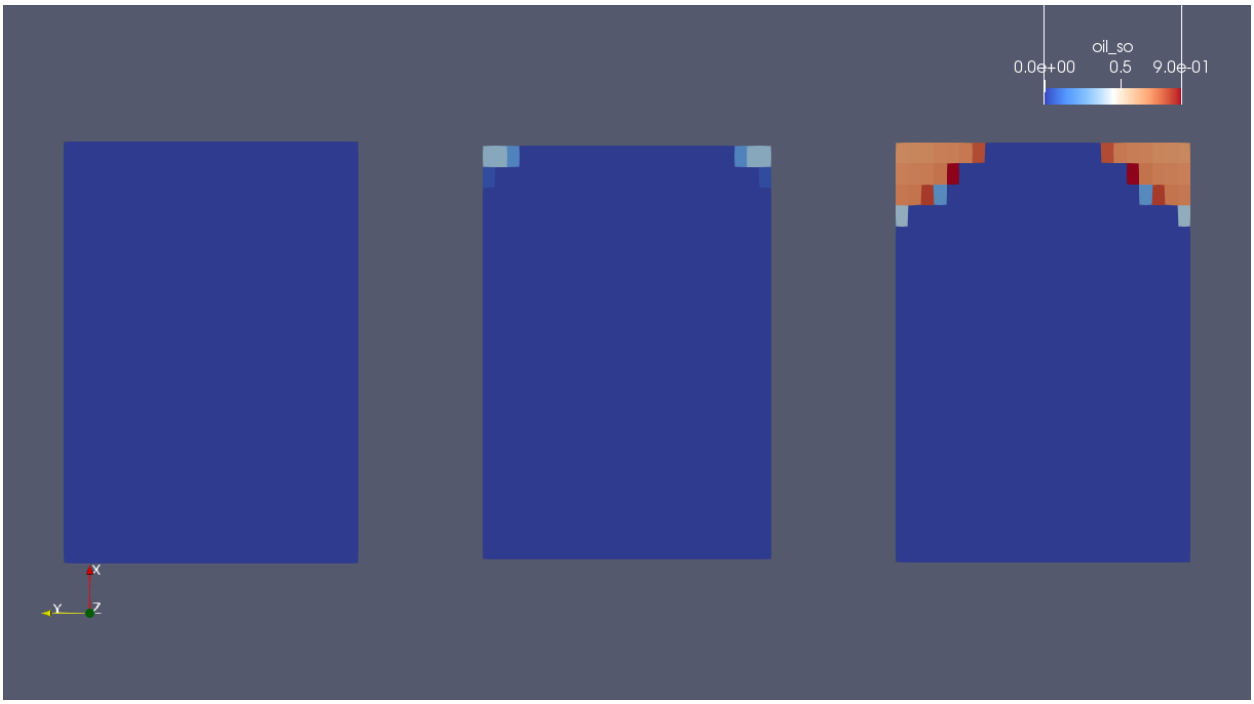

Figure 5.56: Oil saturation distribution on 3 layers after t = 240 days on finer grid.

5.3 Performance of parallel computation

In this section, we adopt finer grids for in-situ combustion, and parallelization is implemented to accelerate these models. For hardware, two clusters from Compute Canada are employed to our simulator, Niagara and Graham.

5.3.1 Example 1

In the previous chapter, we have discussed the factors in grid partitioning to influence the performance of the parallel implementation. To easily illustrate the result of grid partitioning, three models are employed in our simulator.

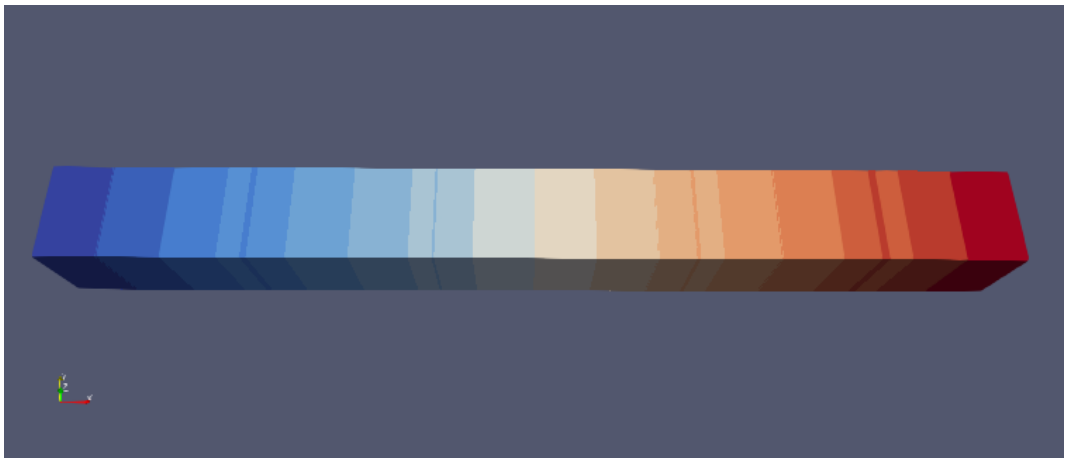

Figure 5.57: Schematic of grid partitioning of a 1-D model.

As shown in Figures 5.57, 5.58 and 5.59, three million-cell models are introduced. For the 1-D grid, grid dimenson is $1000000 \times 1 \times 1$. For the 2-D grid, the grid dimension is $1000 \times 1000 \times 1$. And for the 3-D grid, the grid dimension is $100 \times 100 \times 100$. For the three models, 16 processors are used to handle the whole model, and 16 sub-grids are generated in the process of grid partitioning. For grid partitioning, we know that to improve the load balancing effect and to minimize the volume of communications, there are two factors to

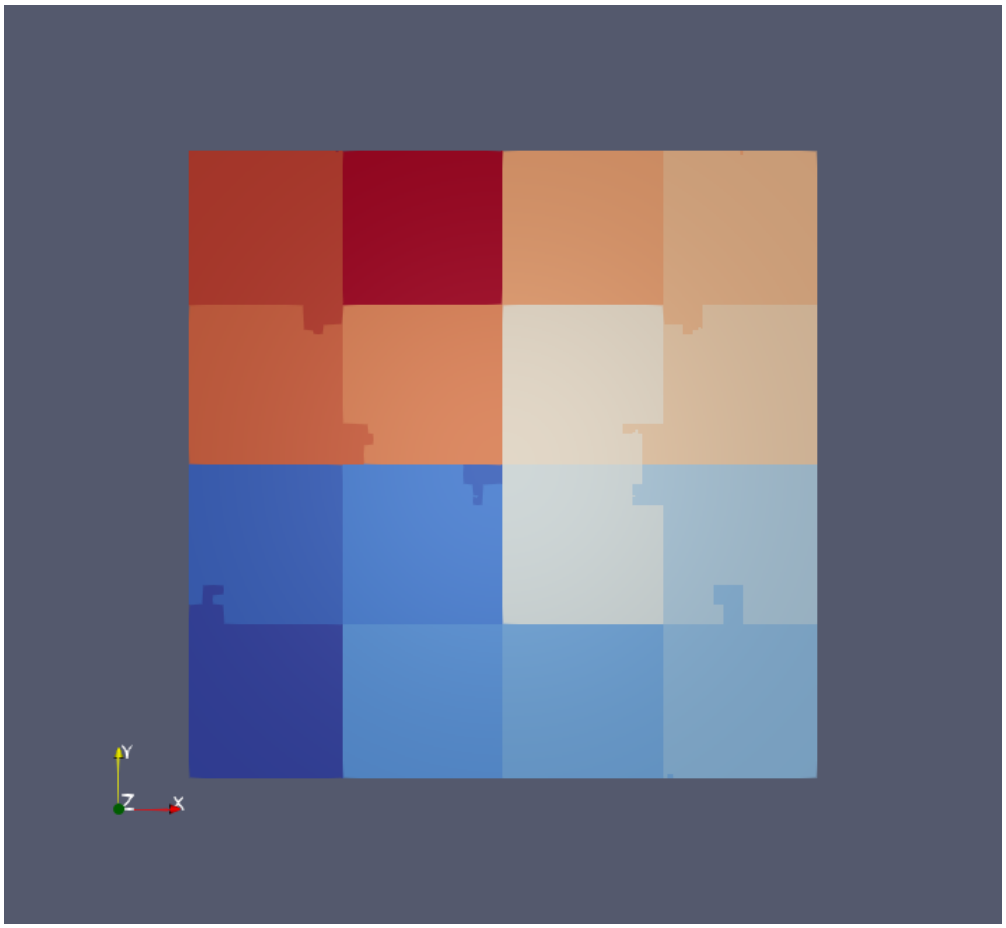

Figure 5.58: Schematic of grid partitioning of a 2-D model.

consider, which are the number of cells in each part, and the number of interfaces of the whole grid.

For the 1-D, 2-D and 3-D model, we can see that each sub-grid has nearly the same number of cells. Since each cell is equally constructed, each part of a processor has a similar area/volume.

For the number of interfaces, if a 1-D million case is employed, the number of interfaces between sub-grids are about $16 - 1 = 15$. But for the 2-D model, the number of of interfaces is about $1000 \cdot (3 + 3) = 6,000$. For 3-D model, it is $100 \cdot 100 \cdot (1 + 1 + 3) = 50,000$. We

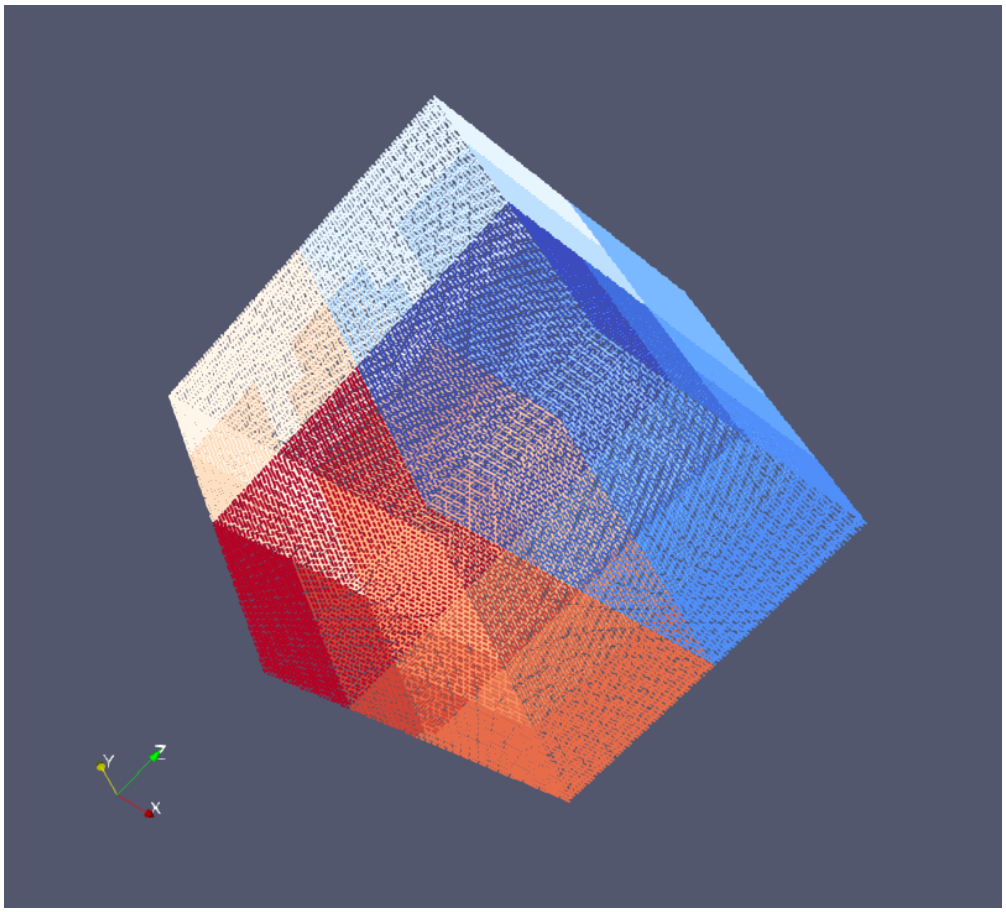

Figure 5.59: Schematic of grid partitioning of a 3-D model.

define the communication ratio

$$\text{comm\_ratio} = \frac{\text{number of interfaces}}{\text{number of cells}}.$$

Then the *comm_ratio* of 1-D, 2-D and 3-D models are 0.0015%, 0.6% and 5%, respectively. Therefore, the overhead resulting from data communication will also increase from a 1-D model to a 3-D model.

The following part will consider the performance of some examples, and the scalability is the main factor of concern. Also, some numerical treatments and grid characteristics are all taken into consideration.

#Processors	8×1	8×2	8×4	8×8	8×16
#Steps	77	77	77	77	77
#Newton	202	202	202	202	202
#Solver	871	896	910	923	963
#Avg. solver	4.31	4.44	4.50	4.57	4.77
Time (s)	2552.72	1334.31	704.32	366.33	208.91
Avg. time (s)	2.931	1.489	0.774	0.397	0.217
speedup (total)	1.000	1.913	3.624	6.968	12.219
speedup (solver)	1.000	1.968	3.787	7.384	13.510

Table 5.11: Numerical summaries of 1D model in Example 1.

The first case is employed as an in-situ combustion model. The grid is a sufficiently fine grid. The grid dimension of the model is $1,000,000 \times 1 \times 1$ cells. That's actually a 1-D model with one million cells. The chemical model is the basic chemical model from (Crookston, R. B., Culham, W. E., and Chen, W. H. (1979)). The BiCGSTAB solver is employed with the tolerance of 1e-5. In addition, the stopping criterion for Newton iteration is 1e-2. The preconditioner pc_index 3 is used.

The second and third models are applied with the different grids. For the second case, it is a 2-D model, and the grid dimension of the model is $1,000 \times 1,000 \times 1$ cells. For the third case, the 3-D model, the grid dimension is $100 \times 100 \times 100$.

All the models are run on 8 nodes. From 1 core per node to 16 cores per node are leveraged. The numerical summary is listed in Table 5.11, 5.12 and 5.13.

For the total running time, we can generate a graph for speedup in Figure 5.60. In this

#Processors	8×1	8×2	8×4	8×8	8×16
#Steps	70	70	70	70	70
#Newton	137	137	137	137	137
#Solver	1412	1455	1501	1554	1686
#Avg. solver	10.31	10.62	10.96	11.34	12.31
Time (s)	3703.12	1984.52	1050.18	578.53	322.79
Avg. time (s)	2.623	1.489	0.774	0.397	0.217
speedup (total)	1.000	1.866	3.526	6.401	11.472
speedup (solver)	1.000	1.923	3.748	7.045	13.699

Table 5.12: Numerical summaries of 2D model in Example 1.

#Processors	8×1	8×2	8×4	8×8	8×16
#Steps	156	155	151	152	151
#Newton	698	695	687	690	685
#Solver	5440	5500	5584	5750	5961
#Avg. solver	4.47	7.91	8.13	8.33	8.70
Time (s)	34353.9	18468.1	10551.8	6233.7	3682.7
Avg. time (s)	6.315	3.358	1.890	1.084	0.618
speedup (total)	1.000	1.860	3.256	5.511	9.328
speedup (solver)	1.000	1.881	3.342	5.825	10.222

Table 5.13: Numerical summaries of 3D model in Example 1.

case, only one processor is used on each node, the first 8×1 is the base case.

As shown in the figure, the speedup deviates from the ideal case, which means that the overhead wasted in the total solution process cannot be neglected. If we concentrate on the numerical summary table, we will find that the number of linear solver iterations increases a lot in all three models, which means that the stability of the solver is not strong in this case. When the number of processors or MPI tasks increase, the whole solution of the linear system is hard to achieve the convergence. Therefore, more iterations are required for the whole linear system.

This result indicates that not only will synchronization and communications affect the total overhead, but also the parallel algorithm itself will also require additional operations and additional time.

In the figure, we can conclude that for the same model, the 1-D model requires fewer communications than the 2-D model does, and the 2-D requires fewer communications than the 3-D model does. Therefore, the 1-D model has the best speed up in this case.

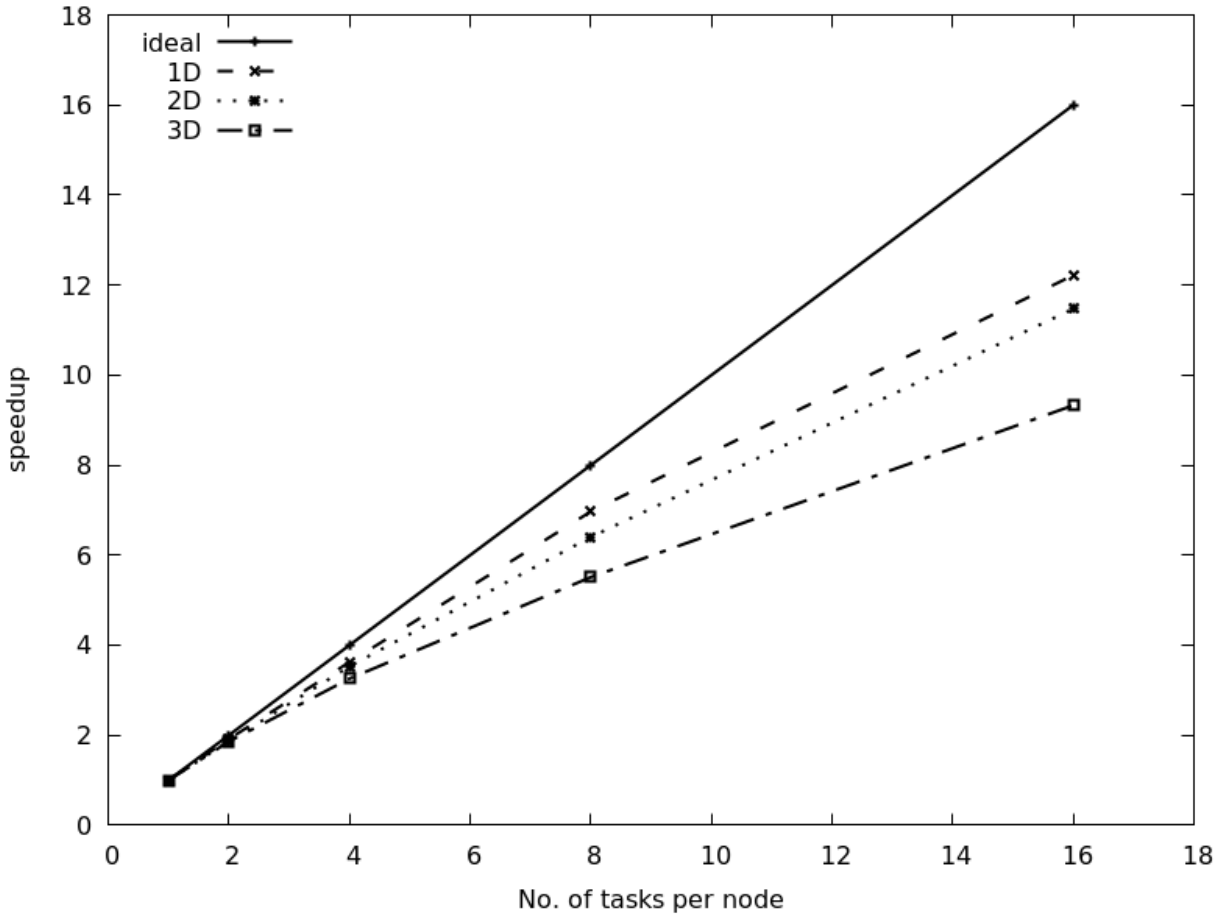

Figure 5.60: Comparison of speedup profiles of Example 1.

5.3.2 Example 2

In this example, we would like to investigate the influence of nodes number. Bandwidth of memory access played an important role in the speedup profile.

For the model, we directly apply the 3D model in last example, and run the simulation on 1 node, 2 nodes and 4 nodes separately.

The numerical summary is listed below:

From the numerical summaries, it can be noticed that number of solver increases rapidly

#Processors	1×1	1×2	1×4	1×8	1×16	1×32
#Steps	10	10	10	10	10	10
#Newton	22	22	22	22	22	22
#Solver	22	23	26	26	28	33
#Avg. solver	1.00	1.05	1.18	1.18	1.27	1.50
Time (s)	5026.9	2687.9	1364.3	733.7	416.6	243.9
Avg. time (s)	228.50	116.87	52.47	28.22	14.88	7.39
speedup (total)	1.000	1.870	3.684	6.851	12.066	20.614
speedup (solver)	1.000	1.955	4.354	8.097	15.357	30.921

Table 5.14: Numerical summaries of 1-node case in Example 2.

#Processors	2×1	2×2	2×4	2×8	2×16	2×32
#Steps	10	10	10	10	10	10
#Newton	22	22	22	22	22	22
#Solver	22	23	26	28	33	33
#Avg. solver	1.00	1.05	1.18	1.27	1.50	1.50
Time (s)	2642.0	1356.8	723.8	395.9	232.1	140.9
Avg. time (s)	120.09	58.99	27.84	14.14	7.03	4.27
speedup (total)	1.000	1.947	3.650	6.673	11.385	18.746
speedup (solver)	1.000	2.036	4.314	8.493	17.077	28.119

Table 5.15: Numerical summaries of 2-node case in Example 2.

#Processors	4×1	4×2	4×4	4×8	4×16	4×32
#Steps	10	10	10	10	10	10
#Newton	22	22	22	22	22	22
#Solver	23	26	28	33	35	36
#Avg. solver	1.05	1.18	1.27	1.50	1.59	1.64
Time (s)	1330.2	717.9	390.4	220.3	129.4	78.4
Avg. time (s)	57.84	27.61	13.94	6.68	3.70	2.18
speedup (total)	1.000	1.853	3.408	6.039	10.277	16.960
speedup (solver)	1.000	2.095	4.149	8.664	15.639	26.547

Table 5.16: Numerical summaries of 4-node case in Example 2.

with number of tasks per node increasing, which has a significant effect on the speedup. Thus, a speedup of solver is chosen for comparison.

In Figure 5.61, it indicates that number of nodes will affect the effects the result of speedup. Generally, when the number of nodes increases, more data communications are required, which will cost a large number of time. In the 1-nodes case, all data communications happen between cores on the same node. Message passing between cores on one node is faster than that on different nodes. And

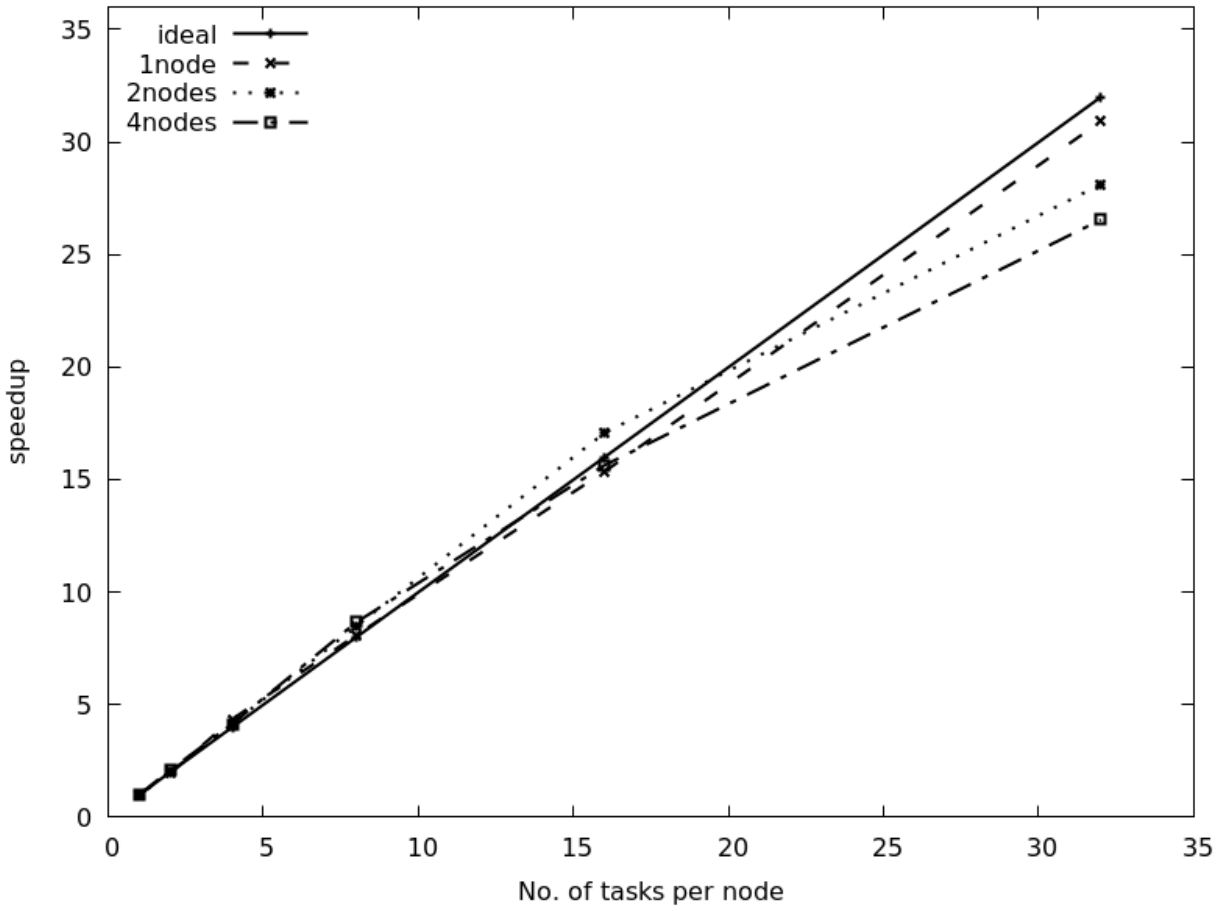

Figure 5.61: Comparison of speedup profiles of Example 2.

5.3.3 Example 3

In this section, we adopt the models in Example 1, and changes the tolerance of linear solver. Also, the time steps are adjusted to similar. Preconditioner is changed to CPR-PF from RAS. With this modification, the number of solvers will remain stable when number of tasks increases. Also, for efficiency of parallelization, 4 nodes are adopted in this example.

The numerical summary is listed below:

It indicates from Figure 5.62 that without the effects of unstable solver, the speedup is

#Processors	4×1	4×2	4×4	4×8	4×16
#Steps	41	41	41	41	41
#Newton	77	77	77	77	77
#Solver	77	77	77	77	77
#Avg. solver	1.00	1.00	1.00	1.00	1.00
Time (s)	2248.2	1029.2	518.0	274.7	168.8
Avg. time (s)	29.197	13.366	6.728	3.568	2.192
speedup (total)	1.000	2.184	4.340	8.183	13.319
speedup (solver)	1.000	2.184	4.340	8.183	13.319

Table 5.17: Numerical summaries of 1-D case in Example 3.

#Processors	4×1	4×2	4×4	4×8	4×16
#Steps	48	48	48	48	48
#Newton	78	78	78	78	78
#Solver	78	78	78	78	78
#Avg. solver	1.00	1.00	1.00	1.00	1.00
Time (s)	2900.4	1481.9	764.0	404.5	231.2
Avg. time (s)	37.184	18.999	9.794	5.185	2.965
speedup (total)	1.000	1.957	3.797	7.171	12.543
speedup (solver)	1.000	1.957	3.797	7.171	12.543

Table 5.18: Numerical summaries of 2-D case in Example 3.

#Processors	4×1	4×2	4×4	4×8	4×16
#Steps	60	60	60	60	60
#Newton	72	72	72	72	72
#Solver	72	72	72	75	83
#Avg. solver	1.00	1.00	1.00	1.04	1.15
Time (s)	4716.0	2568.1	1434.6	803.9	458.9
Avg. time (s)	65.499	35.668	19.926	10.718	5.529
speedup (total)	1.000	1.836	3.287	5.867	10.277
speedup (solver)	1.000	1.836	3.287	6.111	11.847

Table 5.19: Numerical summaries of 3-D case in Example 3.

better than the Example 1.

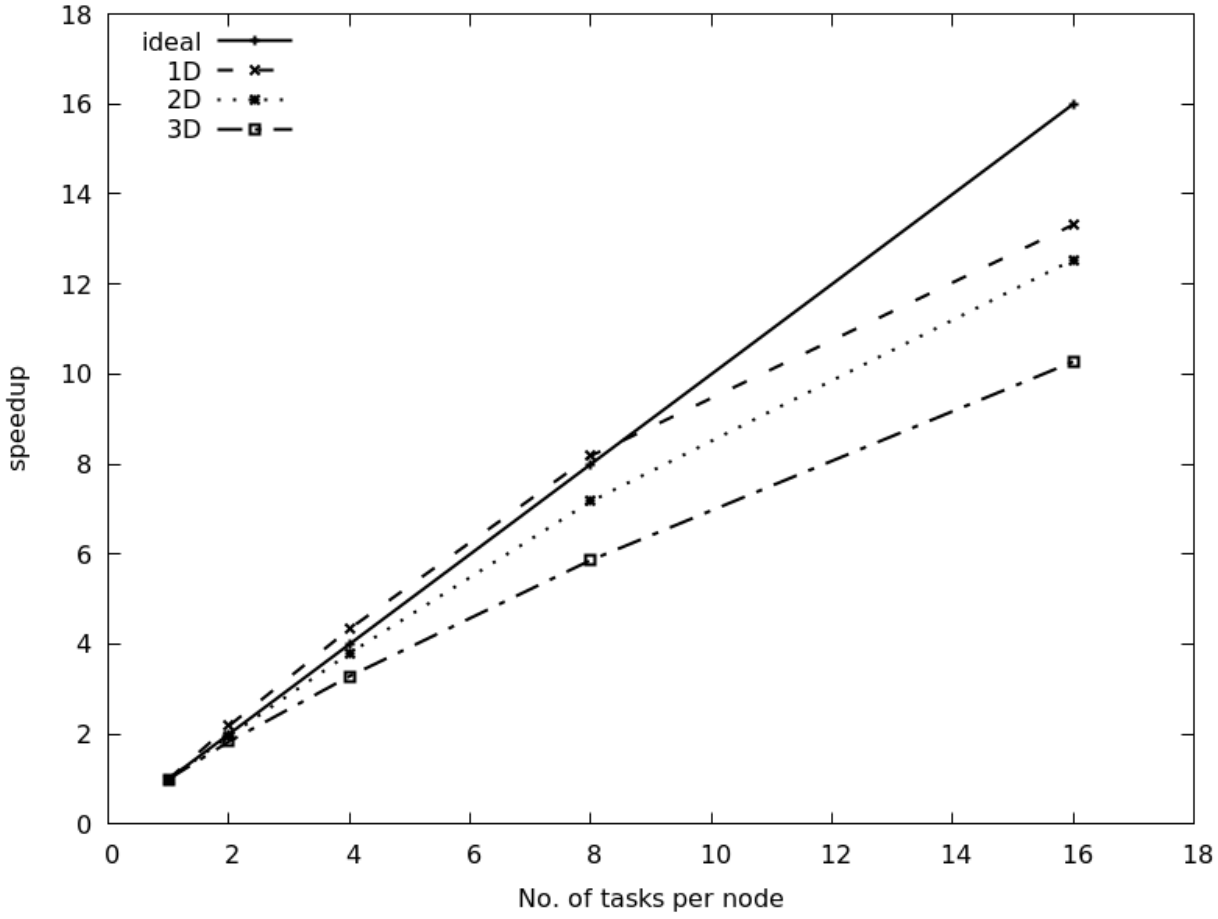

Figure 5.62: Comparison of speedup profiles of Example 3.

Chapter 6

Summary and Future Work

6.1 Summary

In this work, a new parallel In-Situ Combustion Simulator is developed. On the mathematical model, some modifications of methods are implemented. For numerical methods, a novel decoupling method is proposed. In parallelization, some advanced algorithms are first adopted to an ISC simulator.

For the mathematical base of the simulator, both mass conservation equations and energy conservation equations are included. With well equations and the oil phase and gas phase constraint equations, a PDE (partial differential equation) system is established. The multi-component multi-phase flow is described by conservation equations and Darcy's flow equations. Effects of temperature, pressure, and saturation on fluids, rock and rock-fluid properties are taken into account. With all these equations, the non-isothermal process can be well described. For chemical reactions, the Arrhenius models are considered to couple with the equations. With the mathematical model, the discretization and linearization can

be achieved with the finite difference method and Newton method. For a general component model, different from adopting the VS method, a modified PER method is adopted. The results indicate its simplicity and effectiveness. In the validation part, the total error of the simulator is not apparent, which illustrates that the error introduced by this method is tiny. There is no noticeable difference between this method and the VS method in the aspect of accuracy.

In the numerical method, a novel decoupling method, Gauss-Jordan transformation is introduced. Due to the existence of multiple components, the strong coupling phenomenon resulted in the complexity of solving an ISC problem. Without a proper decoupling method, the parallel iterative linear solver cannot reach convergence in expected steps. The new decoupling method not only solves the coupling phenomenon, but its efficiency is also better than some traditional decoupling methods. Furthermore, with a row exchange, a dynamic equation alignment is achieved, which is suitable for the process with rapid and frequent phase appearance and disappearance.

To deal with a complex ISC model and to accurately describe a combustion front, parallelization is implemented. Parallel algorithms are adopted in the simulator, including grid partitioning, parallel preconditioners, and parallel solvers. Among them, the grid partitioning method played a significant role. Since clusters have a distributed-memory architecture, the message passing among nodes is fulfilled by MPI (Message Passing Interface). The volume of data communications will affect the speedup of parallelization, which is an ordinary index of quality or scalability of parallelization. Besides, loading balance will affect the performance as well. Therefore, a dynamic load balancing method, HSFC, is the basis of the

parallel simulator. With this method, the effect of communications and load balancing on the quality of parallelization can be minimized with high efficiency.

With the simulator, results can be obtained from a lab-scale model to a field-scale model. Results from the multi-dimensional model can be validated with CMG STARS, and properties and profiles match the physical process as well. Both suggest positive validations of our simulator. Afterward, the performance of parallelization is evaluated, which indicates the scalability of the simulator.

6.2 Future Work

Although this is the last chapter of the thesis, it does not signify the coda of the whole project. Several aspects of work or improvements remain valuable to finish.

First of all, the improvement of computational efficiency needs to be discussed. For example, using ABF decoupling mainly prevented the consideration of an equation alignment approach especially for the diagonal block. However, for other blocks, a different alignment approach may affect the efficiency of solving linear systems. Also, different decoupling methods and preconditioners may have a different impact on the performance of the simulator, which needs to be discussed in future work. Moreover, this simulator includes the exact and inexact Newton methods, and numerical and analytic Jacobian matrices, their efficiency and behavior in parallelization remain to be investigated.

Second, although the simulator is a general in-situ combustion simulator, more chemical models such as a SARA model have not been tested. In the future work, more models will be tested with this ISC simulator. Also, since parallelization aims to deal with a thin

combustion front, its performance remains to be compared with other approaches such as dynamic gridding.

Bibliography

Abou-Kassem, J. H. and Aziz, K. (1985). "Handling of Phase Change in Thermal Simulators". In: *Journal of petroleum technology* 37(09), pp. 1661–1663.

Adler, G. (1975). "A Linear Model and a Related Very Stable Numerical Method for Thermal Secondary Oil Recovery". In: *Journal of Canadian Petroleum Technology* 14(03).

Akkutlu, I. Y. and Yortsos, Y. C. (2003). "The Dynamics Of In-Situ Combustion Fronts In Porous Media". In: *Combustion and Flame* 134(3), pp. 229–247.

Ali, S. M. (1977). "Multiphase, Multidimensional Simulation of In Situ Combustion". In: *SPE Annual Fall Technical Conference and Exhibition*. Society of Petroleum Engineers.

Anguille, L. et al. (1995). "Static And Dynamic Load-Balancing Strategies for Parallel Reservoir Simulation". In: *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Anis, M., Hwang, M. K., and Odeh, A. S. (1983). "A Sensitivity Study of the Effect of Parameters on Results From an In-Situ Combustion Simulator". In: *Society of Petroleum Engineers Journal* 23(2), pp. 259–264.

Bailey, H. R. and Larkin, B. K. (1959). "Heat Conduction In Underground Combustion". In: *Transactions of The American Institute of Mining and Metallurgical Engineers* 216, pp. 123–129.

Bailey, H. R. and Larkin, B. K. (1960). "Conduction-Convection in Underground Combustion". In: *Petroleum Transactions, AIME* 219, pp. 320–331.

Bank, R. E. et al. (1989). "The Alternate-Block-Factorization Procedure for Systems of Partial Differential Equations". In: *BIT Numerical Mathematics* 29(4), pp. 938–954.

Barney, B. (2007). *Introduction to Parallel Computing*. https://computing.llnl.gov/tutorials/parallel_comp/. Lawrence Livermore National Laboratory.

Belgrave, J. D. M. et al. (1993). "A Comprehensive Approach to In-Situ Combustion Modeling". In: *SPE Advanced Technology Series* 1(01), pp. 98–107.

Belsley, D. A., Kuh, E., and Welsch, R. E. (2005). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Vol. 571. John Wiley & Sons.

Boman, E. et al. (2007). *Zoltan 3.0: Parallel Partitioning, Load-Balancing, and Data Management Services; User's Guide*. Sandia National Laboratories, Albuquerque, NM.

Cai, X. C. and Sarkis, M. (1999). "A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems". In: *SIAM Journal on Scientific Computing* 21(2), pp. 792–797.

Cao, H. (2002). "Development of Techniques for General Purpose Simulators". PhD thesis. Stanford University.

Cao, H. et al. (2005). "Parallel Scalable Unstructured CPR-type Linear Solver for Reservoir Simulation". In: *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Chappelear, J. E. and Rogers, W. L. (1974). "Some Practical Considerations in the Consruction of a Semi-Implicit Simulator". In: *Society of Petroleum Engineers Journal* 14(03), pp. 216–220.

Chen, T. et al. (2009). *Fast Computational Methods for Reservoir Flow Models*. University of Minnesota. Institute for Mathematics and Its Applications.

Chen, X. et al. (2014). "Kinetic Modeling of the In-Situ Combustion Process for Athabasca Oil Sands". In: *SPE Heavy Oil Conference-Canada*. Society of Petroleum Engineers.

Chen, Z. (2007). *Reservoir Simulation: Mathematical Techniques in Oil Recovery*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM.

Chen, Z., Huan, G., and Ma, Y. (2006). *Computational Methods for Multiphase Flows in Porous Media*. Vol. 2. SIAM.

Christensen, J. R. et al. (2004). "Applications of Dynamic Gridding to Thermal Simulations". In: *SPE International Thermal Operations and Heavy Oil Symposium and Western Regional Meeting*. Society of Petroleum Engineers.

Chu, C. (1963). "Two-Dimensional Analysis of a Radial Heat Wave". In: *Journal of Petroleum Technology* 15(10), pp. 1137–1144.

Clapeyron, É. (1834). "Mémoir sur la Puissance Motrice de la Chaleur". In: *Journal de l'École Royale Polytechnique*, pp. 153–190.

CMG (2015). *STARS User's Guide*. Computer Modelling Group Ltd.

CMG STARS (2018). *Thermal and Advanced Processes Simulator*. https://www.cmgl.ca/stars. Computer Modelling Group Ltd.

Coats, K. H. (1976). "Simulation of Steamflooding with Distillation and Solution Gas". In: *Society of Petroleum Engineers Journal* 16(05), pp. 235–247.

Coats, K. H. (1980). "In-Situ Combustion Model". In: *Society of Petroleum Engineers Journal* 20(06), pp. 533–554.

Coats, K. H. (1983). "Some Observations on Field-scale Simulation of the In-situ Combustion Process". In: *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Conti, J. et al. (2016). *International Energy Outlook 2016 With Projections to 2040*. Tech. rep. USDOE Energy Information Administration (EIA), Washington, DC (United States). Office of Energy Analysis.

Corey, A. T. et al. (1956). "Three-phase Relative Permeability". In: *Journal of Petroleum Technology* 8(11), pp. 63–65.

Crookston, R. B., Culham, W. E., and Chen, W. H. (1979). "A numerical Simulation Model for Thermal Recovery Processes". In: *Society of Petroleum Engineers Journal* 19(01), pp. 37–58.

Delshad, M. and Pope, G. A. (1989). "Comparison of the three-phase oil relative permeability models". In: *Transport in Porous Media* 4(1), pp. 59–83.

Dembo, R. S., Eisenstat, S. C., and Steihaug, T. (1982). "Inexact Newton Methods". In: *SIAM Journal on Numerical analysis* 19(2), pp. 400–408.

Elman, H. C., Silvester, D. J., and Wathen, A. J. (2014). *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation.

Ertekin, T., Abou-Kassem, J. H., and King, G. R. (2001). *Basic Applied Reservoir Simulations*. Society of Petroleum Engineers.

Falgout, R. D. and Yang, U. M. (2002). "Hypre: A Library of High Performance Preconditioners". In: *International Conference on Computational Science*. Springer, pp. 632–641.

Fassihi, M. R., Yannimaras, D. V., and Kumar, V. K. (1997). "Estimation of recovery factor in light-oil air-injection projects". In: *SPE Reservoir Engineering* 12(03), pp. 173–178.

Gottfried, B. S. (1965). "A Mathematical Model of Thermal Oil Recovery in Linear Systems". In: *Society of Petroleum Engineers Journal* 5(03), pp. 196–210.

Grabowski, J. W. et al. (1979). "A Fully Implicit General Purpose Finite-Difference Thermal Model for In-Situ Combustion and Steam". In: *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Greenbaum, A. (1997). *Iterative Methods for Solving Linear Systems*. Vol. 17. SIAM.

Gries, S. (2015). "System-AMG Approaches for Industrial Fully and Adaptive Implicit Oil Reservoir Simulations". PhD thesis. Universität zu Köln.

Gutierrez, D., Moore, R. G., et al. (2012). "The ABCs of In-Situ Combustion Simulations: from Laboratory Experiments to Field Scale". In: *Journal of Canadian Petroleum Technology* 51(04), pp. 256–267.

Gutierrez, D., Skoreyko, F., et al. (2009). "The Challenge of Predicting Field Performance of Air Injection Projects Based on Laboratory and Numerical Modelling". In: *Journal of Canadian Petroleum Technology* 48(04), pp. 23–33.

Hackbusch, W. (1994). *Iterative Solution of Large Sparse Systems of Equations*. Vol. 95. Springer.

Hager, G. and Wellein, G. (2010). *Introduction to High Performance Computing for Scientists and Engineers*. CRC Press.

Hasiba, H. H. and Wilson, L. A. (1974). "The Potential Contribution of Enhanced Oil Recovery Technology to Domestic Crude Oil Reserves". In: *Petroleum Chemistry Division's Symposium on the Role of Technology in the Energy Crisis*. The American Chemical Society, pp. 487–494.

Huang, C. K. (2009). "Development of a General Thermal Oil Reservoir Simulator under a Modularized Framework". PhD thesis. The University of Utah.

Hwang, M. K., Jines, W. R., and Odeh, A. S. (1982). "An In-Situ Combustion Process Simulator With a Moving-Front Representation". In: *Society of Petroleum Engineers Journal* 22(02), pp. 271–279.

Ito, Y. and Chow, A. K. Y. (1988). "A Field Scale In-Situ Combustion Simulator With Channeling Considerations". In: *SPE reservoir engineering* 3(02), pp. 419–430.

Jensen, O. K. (1980). "An Automatic Timestep Selection Scheme for Reservoir Simulation". In: *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Jiang, Y. (2008). "Techniques for Modeling Complex Reservoirs and Advanced Wells". PhD thesis. Stanford University.

Kaarstad, T. et al. (1995). "A Massively Parallel Reservoir Simulator". In: *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Karypis, G. and Kumar, V. (1998). *METIS - A Software Package for Partitioning Unstructured graphs, Partitioning meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*. University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center.

Karypis, G., Schloegel, K., and Kumar, V. (1997). "Parmetis: Parallel Graph Partitioning and Sparse Matrix Ordering Library". In: *Version 1.0, Dept. of Computer Science, University of Minnesota*, p. 22.

Kelly, E. N. et al. (2010). "Oil Sands Development Contributes Elements Toxic at Low Concentrations to the Athabasca River and Its Tributaries". In: *Proceedings of the National Academy of Sciences* 107(37), pp. 16178–16183.

Killough, J. E. and Bhogeswara, R. (1991). "Simulation of Compositional Reservoir Phenomena on a Distributed-Memory Parallel Computer". In: *Journal of Petroleum Technology* 43(11), pp. 1368–1374.

Killough, J. E., Camilleri, D., et al. (1997). "A Parallel Reservoir Simulator Based on Local Grid Refinement". In: *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Klie, H., Rame, M., and Wheeler, M. (1996). *Two-Stage Preconditions for Inexact Newton Methods in Multi-phase Reservoir Simulation*. Tech. rep. Center for Research on Parallel Computation, Rice University.

Kristensen, M. R. (2008). "Development of Models and Algorithms for the Study of Reactive Porous Media Processes". PhD thesis.

Lacroix, S., Vassilevski, Y. V., and Wheeler, M. F. (2000). *Iterative Solvers of the Implicit Parallel Accurate Reservoir Simulator (IPARS), I: Single Processor Case*. Tech. rep. TICAM, The University of Texas at Austin.

Liesen, J. and Strakos, Z. (2013). *Krylov Subspace Methods: Principles and Analysis*. Oxford University Press.

Lin, C. Y. et al. (1984). "Numerical Simulation of Combustion Tube Experiments and the Associated Kinetics of In-Situ Combustion Processes". In: *Society of Petroleum Engineers Journal* 24(06), pp. 657–666.

Liu, H. (2008). "Dynamic Load Balancing on Adaptive Unstructured Meshes". In: *10th IEEE International Conference on High Performance Computing and Communications*. IEEE.

Liu, H., Wang, K., and Chen, Z. (2016). "A Family of Constrained Pressure Residual Preconditioners for Parallel Reservoir Simulations". In: *Numerical linear algebra with applications* 23(1), pp. 120–146.

Liu, H., Wang, K., Chen, Z., Jordan, K. E., et al. (2015). "A Parallel Framework for Reservoir Simulators on Distributed-Memory Supercomputers". In: *SPE/IATMI Asia Pacific Oil & Gas Conference and Exhibition*. Society of Petroleum Engineers.

Liu, H., Wang, K., Chen, Z., Luo, J., et al. (2015). "Development of Parallel Reservoir Simulators on Distributed-memory Supercomputers". In: *SPE Reservoir Characterisation and Simulation Conference and Exhibition*.

Liu, H., Wang, K., Chen, Z., Yang, B., et al. (2016). "Large-Scale Reservoir Simulations on Distributed-Memory Parallel Computers". In: *Proceedings of the 24th High Performance Computing Symposium*. Society for Computer Simulation International, p. 8.

Liu, H., Wang, K., Yang, B., et al. (2017). "Dynamic Load Balancing Using Hilbert Space-Filling Curves for Parallel Reservoir Simulations". In: *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers.

Liu, H., Zhang, P., et al. (2016). "Performance and Scalability Analysis for Parallel Reservoir Simulations on Three Supercomputer Architectures". In: *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale*. ACM, p. 9.

Mahinpey, N., Ambalae, A., and Asghari, K. (2007). "In-Situ Combustion in Enhanced Oil Recovery (EOR): A Review". In: *Chemical Engineering Communications* 194(8), pp. 995–1021.

Mehra, R., Hadjitofi, M., and Donnelly, J. K. (1982). "An Automatic Time-Step Selector For Reservoir Models". In: *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Meurant, G. (1999). *Computer Solution of Large Linear Systems*. Vol. 28. Elsevier.

Montes, A. R. et al. (2010). "Is High-pressure Air Injection (HPAI) Simply a Flue-Gas Flood?" In: *Journal of Canadian Petroleum Technology* 49(02), pp. 56–63.

Moore, R. G. et al. (1995). "In-Situ Combustion in Canadian Heavy Oil Reservoirs". In: *Fuel* 74(8), pp. 1169–1175.

Naar, J. and Wygal, R. J. (1961). "Three-Phase Imbibition Relative Permeability". In: *Society of Petroleum Engineers Journal* 1(04), pp. 254–258.

Nissen, A. et al. (2015). "Upscaling Kinetics for Field-Scale In-situ Combustion Simulation". In: *SPE Reservoir Evaluation & Engineering* 18(02), pp. 158–170.

Odeh, A. S. (1969). "Reservoir Simulation... What is It". In: *Journal of Petroleum technology* 21(11), pp. 1383–1388.

Oklany, J. S. F. (1992). "An In-situ Combustion Simulator for Enhanced Oil Recovery". PhD thesis. University of Salford.

Olshanskii, M. A. and Tyrtshnikov, E. E. (2014). *Iterative Methods for Linear Systems: Theory and Applications*. Vol. 138. SIAM.

Parashar, M. et al. (1997). "A New Generation EOS Compositional Reservoir Simulator: Part II-Framework and Multiprocessing". In: *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Peaceman, D. W. (1978). "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation". In: *Society of Petroleum Engineers Journal* 18(03), pp. 183–194.

Peng, D. Y. and Robinson, D. B. (1976). "A New Two-Constant Equation of State". In: *Industrial & Engineering Chemistry Fundamentals* 15(1), pp. 59–64.

Ramey Jr, H. J. (1959). "Transient Heat Conduction During Radial Movement of a Cylindrical Heat Source-Applications to the Thermal Recovery Process". In: *Petroleum Transactions, AIME* 216, pp. 115–122.

Redlich, O. and Kwong, J. N. (1949). "On the Thermodynamics of Solutions. V. An Equation of State. Fugacities of Gaseous Solutions." In: *Chemical reviews* 44(1), pp. 233–244.

Rubin, B. and Vinsome, P. K. W. (1980). "The Simulation of the In-Situ Combustion Process in One Dimension Using a Highly Implicit Finite-Difference Scheme". In: *Journal of Canadian Petroleum Technology* 19(04).

Rubin, B. and Vinsome, P. K. W. (1985). "A General Purpose Thermal Model". In: *Society of Petroleum Engineers Journal* 25(02), pp. 202–214.

Rutledge, J. M. et al. (1992). "The Use of a Massively Parallel SIMD Computer for Reservoir Simulation". In: *SPE Computer Applications* 4(04), pp. 16–21.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Vol. 82. SIAM.

Saad, Y. and Schultz, M. H. (1986). "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems". In: *SIAM Journal on scientific and statistical computing* 7(3), pp. 856–869.

Sammon, P. H. (2003). "Dynamic Grid Refinement and Amalgamation For Compositional Simulation". In: *SPE reservoir simulation symposium*. Society of Petroleum Engineers.

Sammon, P. H. and Rubin, B. (1986). "Practical control of timestep selection in thermal simulation". In: *SPE Reservoir Engineering* 1(02), pp. 163–170.

Sarathi, P. (1999). *In-Situ Combustion Handbook–Principles and Practices*. Tech. rep. National Petroleum Technology Office (US).

Sequera, B. et al. (2010). "Numerical Simulation of In-Situ Combustion Experiments Operated Under Low Temperature Conditions". In: *Journal of Canadian Petroleum Technology* 49(01), pp. 55–64.

Sleijpen, G. L. and Fokkema, D. R. (1993). "BiCGstab (l) for Linear Equations Involving Unsymmetric Matrices with Complex Spectrum". In: *Electronic Transactions on Numerical Analysis* 1(11), p. 2000.

Smith, J. T. and Ali, S. M. (1971). "Simulation of In-Situ Combustion in a Two-Dimensional System". In: *Fall Meeting of the Society of Petroleum Engineers of AIME*. Society of Petroleum Engineers.

Soave, G. (1972). "Equilibrium Constants from a Modified Redlich-Kwong Equation of State". In: *Chemical Engineering Science* 27(6), pp. 1197–1203.

Stone, H. L. (1970). "Probability Model for Estimating Three-Phase Relative Permeability". In: *Journal of Petroleum Technology* 22(02), pp. 214–218.

Stone, H. L. (1973). "Estimation of Three-Phase Relative Permeability and Residual Oil Data". In: *Journal of Canadian Petroleum Technology* 12(4).

Thomas, G. W. (1963). "A Study of Forward Combustion in a Radial System Bounded by Permeable Media". In: *Journal of Petroleum Technology* 15(10), pp. 1145–1149.

Todd, M. R., O'dell, P. M., and Hirasaki, G. J. (1972). "Methods for Increased Accuracy in Numerical Reservoir Simulators". In: *Society of Petroleum Engineers Journal* 12(06), pp. 515–530.

Toselli, A. and Widlund, O. (2006). *Domain Decomposition Methods-Algorithms and Theory*. Vol. 34. Springer Science & Business Media.

Trefethen, L. N. and Bau III, D. (1997). *Numerical Linear Algebra*. Vol. 50. SIAM.

Ursenbach, M. G., Moore, R. G., and Mehta, S. A. (2010). "Air Injection in Heavy Oil Reservoirs-A Process Whose Time Has Come (Again)". In: *Journal of Canadian Petroleum Technology* 49(01), pp. 48–54.

Van der Steen, A. J. and Dongarra, J. J. (2012). "High-Performance Computing Systems: Status and Outlook." In: *Acta Numerica* 21, pp. 379–474.

Van der Vorst, H. A. (1992). "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems". In: *SIAM Journal on scientific and Statistical Computing* 13(2), pp. 631–644.

Van der Vorst, H. A. (2003). *Iterative Krylov Methods for Large Linear Systems*. Vol. 13. Cambridge University Press.

Van der Waals, J. D. (1910). "The Equation of State for Gases and Liquids". In: *Nobel lectures in Physics* 1, pp. 254–265.

Vaughn, P. and Wyoming, L. (1986). *A Numerical Model for Thermal Recovery Processes in Tar Sand: Description and Application*. Morgantown Energy Technology Center, US Department of Energy.

Wallis, J. R., Kendall, R. P., and Little, T. E. (1985). "Constrained Residual Acceleration of Conjugate Residual Methods". In: *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Wang, K., Liu, H., and Chen, Z. (2015). "A Scalable Parallel Black Oil Simulator on Distributed Memory Parallel Computers". In: *Journal of Computational Physics* 301, pp. 19–34.

Youngren, G. K. (1980). "Development and Application of an In-situ Combustion Reservoir Simulator". In: *Society of Petroleum Engineers Journal* 20(01), pp. 39–51.

Zhong, H. et al. (2016). "A Parallel Thermal Reservoir Simulator on Distributed-Memory Supercomputers". In: *SPE Asia Pacific Oil & Gas Conference and Exhibition*. Society of Petroleum Engineers.

Zhu, Z. (2011a). "Efficient Simulation of Thermal Enhanced Oil Recovery Processes". PhD thesis. Stanford University.

Zhu, Z. (2011b). "Upscaling for Field-Scale In-Situ Combustion Simulation". In: *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.