AN IN-SITU COMBUSTION SIMULATOR FOR ENHANCED OIL RECOVERY

A Thesis Submitted to

THE UNIVERSITY OF SALFORD

for the degree of

DOCTOR OF PHILOSOPHY

by

JAAFAR SADIQ F. A. OKLANY

Under the Direction of

DR R HUGHES

and

DR D PRICE

Department of Chemical and Gas Engineering
University of Salford
Salford M4 5WT

March 1992

DECLARATION

Unless otherwise stated, the work in this thesis is that of the author, and has not previously been submitted, in part or in whole, at this or any other establishment.

ACKNOWLEDGEMENTS

I would like to express my appreciation and my thanks to all members of the Department of Chemical Engineering, staff and fellow students for a most enjoyable and rewarding course.

In particular, I would like to thank Dr. R Hughes (Chemical & Gas Engineering) and Dr. D Price (Chemistry Department) for their help and guidance as well as for their time and attention.

I am also very grateful to Dr. A Moqber (University of Bath).

I am indebted beyond hope of repayment to my family back home.

Finally, my deepest thanks to all who have assisted me in the preparation of this thesis.

TABLE OF CONTENTS

ABSTRACT

A multi-dimensional three phase flow simulator is written to simulate the process of in-situ combustion. It has six components (oxygen, inert gas, a light oil, a heavy oil, water, and coke). The inert gas consists of all the noncondensible gases other than oxygen. The vaporization and condensation of both water and oil is governed by vapour-liquid equilibria, using temperature and pressure dependent equilibrium coefficients. Splitting of the oil into fractions is done to include the distillation effect.

The model includes four chemical reactions. These represent in turn the formation of coke from the heavy oil, the oxidation of the coke, and the oxidation of the light and the heavy oil. Heat transport is assumed to take place by convection and conduction. The effects of gravity, capillary pressure between any two fluid phases , and heat losses to the surrounding rock via conduction are also included.

The model is validated using the ISCOM simulator. This has been carried out by comparing the results from the present simulator with these from similar calculations performed using the ISCOM simulator. ISCOM is a fully implicit multi-dimensional finite difference general thermal simulator developed by the computing modelling group (CMG) at Calgary. This comparison is needed in order to establish confidence in the predictive capability of the present model for a particular reservoir.

In addition, a comprehensive study of the effect of the main input parameters on the performance characteristics of the in-situ

combustion process is carried out. This study is made in order to indicate which data are "important" in the sense of having a significant influence on the performance characteristics of the process and to test the robustness of the program to changes in the input parameters.

Chapter One

INTRODUCTION

1.1 Conventional Oil Recovery Methods

The inescapable fact that hydrocarbon reserves in the world are decreasing, has led to considerable research and investment in exploration for new conventional hydrocarbon reserves, and development of unconventional hydrocarbon resources such as oil shale or tar sand. Developing a production technique that will recover close to one hundred percent of the oil in a petroleum reservoir has been the dream of petroleum engineers and production managers for years.

The three natural oil recovery mechanisms in oil reservoirs are solution gas drive, gas cap expansion drive and water drive. Water or gas injection drives are known as secondary recovery methods. Enhanced oil recovery (EOR) usually involves tertiary recovery--the third method for recovery of additional oil not economically recoverable by secondary methods. All processes are strongly affected by reservoir description. Recovery process effectiveness depends so heavily on reservoir fluid flow and phase behavior that a thorough knowledge of these factors is essential.

Ultimate oil recovery by conventional methods can vary from less than 10% of the oil originally in place with solution gas drive to well above 50% with water drive or water injection. Conventional recoveries might be only 5% or less for highly viscous oils and essentially zero in the case of extremely viscous bitumens. The average oil recovery from light and medium crude oil reservoirs produced by conventional methods worldwide, is around 35%.

There are three major reasons why over half of the oil in place in the average reservoir is unrecoverable by conventional means. First, only a portion of any reservoir can be contacted by the displacing fluid. Second, not all of the oil can be displaced from the rock that is contacted by the displacing fluid. And third, heavier, low-gravity oils are frequently too viscous to move to the production wells at rates sufficient to support an economic operation. The objective of an EOR process is to, at least partially, overcome one or more of these limitations.

1.2 Enhanced Oil Recovery (EOR) Methods

Enhanced oil recovery methods can be classified to non-thermal and thermal methods.

1.2.1 Non-Thermal Methods

The principal non-thermal methods are polymer flooding, surfactant flooding, and miscible flooding.

1.2.1.1 Polymer Flooding

In general, incremental recoveries for polymer flooding will be low, as this process cannot mobilize trapped residual oil. A major technical problem is that polymers are susceptible to thermal, chemical, or biological degradation.

Because the viscosity of a polymer solution is higher than that of water, polymer injection can result in reduced injectivity, increased project lives, and delayed production. In addition, the injected polymer may plug the sandface at the wellbore if the rock permeability is too low, particularly if an improper polymer is used or if it is not properly dissolved.

1.2.1.2 Surfactant Flooding

Surfactant flooding can recover trapped residual oil. In most applications, a surfactant solution is injected to reduce the interficial tension between the water and oil and mobilizes the trapped oil to form a flowing oil bank. Polymers are used in a surfactant flood to promote good reservoir sweep by increasing the viscosity of the water used to drive the surfactant bank through the reservoir. They are sometimes also used to increase the viscosity of the injected surfactant bank. Research has shown that high recovery levels of light oil can be obtained with well-designed surfactant floods in some reservoirs.

The high cost of injected chemicals is a major drawback of surfactant flooding and is frequently the most important factor in determining process economics. In addition, the high investment in chemical plant and facilities to manufacture and inject the surfactant and polymers causes heavy front-end loading of costs, which also detracts from favorable economics.

1.2.1.3 Miscible Flooding

Miscible flooding involves the injection of a fluid that is either miscible with the crude oil or generates a miscible solvent bank after contacting the oil. In most cases, the fluids normally injected (nitrogen, carbon dioxide, or light hydrocarbons) are gases at ambient conditions but have liquidlike densities at reservoir conditions.

Miscible flooding is less complex than surfactant flooding but considerably more complex than polymer flooding. Miscible floods are capable of completely displacing oil from zones that have been swept with solvent. A major hurdle for miscible flooding in many reservoirs is the need for a low-cost source of injectants. The major technical problem encountered in miscible flooding is poor volumetric sweep. The miscible gas is always less viscous and usually less dense than the reservoir crude oil. This causes fingering and gravity override of the injected solvent and can leave much of the oil uncontacted.

Most worldwide EOR activity to date has been concentrated in the United States. Of the 512 projects active in 1986, 201, or 39%, were thermal project. However, these thermal projects are responsible for 78% of the total domestic EOR production. So, the vast majority of commercial EOR production is obtained by thermal methods.

1.2.2 Thermal Methods

Thermal recovery methods include steam stimulation, hot water injection, steam flooding, and in-situ combustion.

1.2.2.1 Steam Stimulation

Steam stimulation, also known as cyclic steam injection, steam soak or "huff and puff" is the process in which steam is injected into a producing well for a specified period of time (normally 2-3 weeks). Following this, the well is 'shut-in' (i.e. the steam supply is shut off) for a few days (to allow sufficient heat dissipation) and then placed on production. Heat from the injected steam increases the reservoir temperature, resulting in a more pronounced increase in mobility of heavy oils and a corresponding improvement in production. Other positive benefits that may contribute to production stimulation include; thermal expansion of fluids, compression of solution gas, reduced residual oil saturation, and wellbore cleanup effects.

1.2.2.2 Hot Water Injection

Hot water injection is perhaps the simplest of all thermal drive processes and is the closest to a conventional waterflood in its ease of operation. It is not widely used, however, because of the limitations of the mechanisms aiding oil displacement. As temperature increases, oil mobility improves relative to that of water because viscosity decreases more rapidly with increasing temperature for a heavy oil than for water. Also, there is some thermal expansion of oil.

A major disadvantage of hot water injection, compared to steam injection, is that maximum energy injection rates are usually lower because saturated steam has an energy content more than three times that of hot water below 423° F. Heat losses both from the injection wellbore and from the formation will reduce the volume of formation that can be heated effectively. This effect is most severe at low injection rates and in thin formations.

1.2.2.3 Steam Flooding

Steam flooding is a process similar to water flooding. A suitable well pattern is chosen and steam is injected into a number of wells while the oil is produced from adjacent wells. Ideally the steam forms a saturated zone around the injection well. The temperature of this zone is nearly equal to that of the injected steam. As the steam moves away from the well, its temperature drops as it continues to expand in response to pressure drop. At some distance from the well, the steam condenses and forms a hot water bank. In the steam zone, oil is displaced by steam distillation and gas (steam) drive. In the hot water zone, physical changes in the characteristics of the oil and reservoir rock take place and result in oil recovery. These changes are thermal expansion of the oil, and reduction of viscosity and residual saturation.

Steam flooding is currently the principal enhanced oil recovery method. The daily recovery by steam injection amounts to $405 * 10^3$ barrel per day (BPD) worldwide.

1.2.2.4 In-Situ Combustion

Since it is the major interest of this work, a more detailed introduction and description is given to the in-situ combustion method.

During in-situ combustion, heat is generated within the reservoir by igniting some of the original oil in place and then propagating the combustion front by continuous injection of some oxidant, typically air or oxygen enriched air. The heat generated decreases the viscosity and increases the mobility of the unburned oil in the immediate vicinity. If we consider the simple case of a reservoir consisting of one air injection well and one production well, and if the zone surrounding the injection well-bore is brought to a sufficiently high temperature, ignition takes place and the combustion front travels in the direction of the production well, i.e. in the direction of the fluid flow. This is known as forward combustion. On the other hand, if the zone surrounding the production well is heated, ignition takes place near this well and the combustion front travels towards the injection well, i.e. the opposite direction to the fluid flow. This is known as reverse combustion.

There are two different forms of forward combustion; dry, and wet. Dry forward combustion is the most commonly used form. In this process only air or oxygen enriched air is injected, whereas in wet combustion water is injected either alternately or simultaneously with air. This water is used to recover the heat generated by combustion in the burnout strata behind the combustion front and transfer this heat ahead of the combustion front.

1.2.2.4.1 Dry In-Situ Combustion

Figure 1.1 shows the mechanism of dry forward combustion under steady state conditions. Four zones exist in the reservoir between the injection well and the production well. These zones are described below [Latil (1980)]:

Zone 1: Combustion has already taken place and the formation in this zone is completely clean.

Zone 2: The combustion zone. Oxygen is consumed by a combination of reactions involving the hydrocarbons and the coke remaining on the rock surface. The maximum temperature is obtained in this zone.

Zone 3: The coke formation zone. The heavy oil fractions which have been neither displaced nor vaporized undergo pyrolysis.

Zone 4: No significant chemical changes happen in this zone. The zone is swept by the combustion gases and displaced fluids and the following phenomena take place.

(a) In the downstream region nearest the reaction zone, successive vaporization and condensation of the light oil fractions and the interstitial water take place, as does condensation of the water of combustion.

(b) In the region where the temperature is lower than that of water condensation, a zone with water saturation (saturation expresses how much of the pore space is occupied by the specific phase) higher than the initial water saturation exists (water bank) which displaces a zone with an oil saturation higher than the original (oil bank). Beyond the oil bank, the formation progressively approaches its original conditions.

1.2.2.4.2 Wet In-Situ Combustion

Figure 1.2 shows the process of wet combustion under steady state conditions. Five zones exist in the reservoir between the injection well and the production well. These are described below:

Zone 1: This zone has already been swept by the combustion front and contains little or no hydrocarbons. However, since the temperature is lower than the boiling point of water, the pores contain a liquid water saturation, the remainder of the space being occupied by the injected air.

Zone 2: Water is in the vapour phase in this zone, and the pores are saturated with a mixture of injected air and steam.

Zone 3: The combustion zone. Oxygen is consumed in the combustion of the hydrocarbons and of the deposited coke formed in the downstream part of the zone.

Zone 4: The vaporization-condensation zone. Progressive condensation of steam and combustion water takes place in this zone. In addition, some light and intermediate oil fractions are vaporized and carried downstream. If the temperature is high enough certain chemical reactions may occur in this zone.

Zone 5: Just downstream of the vaporization condensation zone is a zone of high back pressure, due to the formation of a water bank preceded by an oil bank. Further downstream the formation gradually approaches its initial conditions.

1.2.2.4.3 Reverse In-Situ Combustion

Reverse combustion might be likened to blowing through a cigarette, rather than inhaling, to propagate combustion. In reverse combustion, air flow is counter to the direction of the movement of the combustion front. This process was originally conceived from preheating cold, highly viscous oils, as in the Athabasca tar sands, where there would be no movement of the highly viscous oil bank ahead of the combustion front in a forward drive.

The process consists of injection of air until there is communication with the producing well, then using a downhole heater or other means to ignite the oil sand around the producing well. The combustion front burns back toward the injection well.

Four zones can be defined, starting from the injection well, as shown in figure 1.3:

Zone 1: The formation is at original conditions. However it is being swept by air, and if the formation temperature and oxidability of the oil are high, certain oxidation reactions may occur.

Zone 2: The temperature increases by conduction from the hot zone downstream. The start of oxidation also contributes to the temperature increase. The following phenomena occur: vaporization of the formation water, distillation of the light fractions of the oil and cracking of certain hydrocarbons in the presence of oxygen. The liquid and vapour fractions are displaced downstream, while certain components form the carbon deposit or "coke".

Zone 3: The combustion zone. The temperature reaches its maximum value. The oxidation and combustion reactions involving the most reactive hydrocarbon molecules consume all the oxygen not used by the reactions in the preceding zones.

Zone 4: The unburnt coke remains deposited on the matrix while the vapour and liquid phases flow downstream. If there were no heat losses, the downstream temperature would remain equal to that of the combustion front. In reality, the temperature decreases with distance from the combustion zone. Thus condensation of the distilled oil fractions occurs, and possibly of the steam also.

The reverse combustion process is impractical for a number of reasons:

1. Only the lighter components rich in hydrogen burn, leaving a tarry residuum on the rock matrix. This makes the ratio of the injected air to oil recovered extremely high.

2. Damage to the producing well completion can occur on ignition.

3. Spontaneous ignition near the injection well will occur eventually, and the process will revert to forward combustion.

Because of these defects, reverse combustion is of little practical interest and will not be considered further in this thesis.

The complex phenomena which occur during in-situ combustion are far from being fully clarified. Numerical simulation is a powerful tool which can help to achieve this objective. The simulation includes the development of a model which can predict the performance of a reservoir and can be used to make intelligent decisions regarding future reservoir operations.

Its the purpose of this work to develop a simulator which can help to understand in-situ combustion.

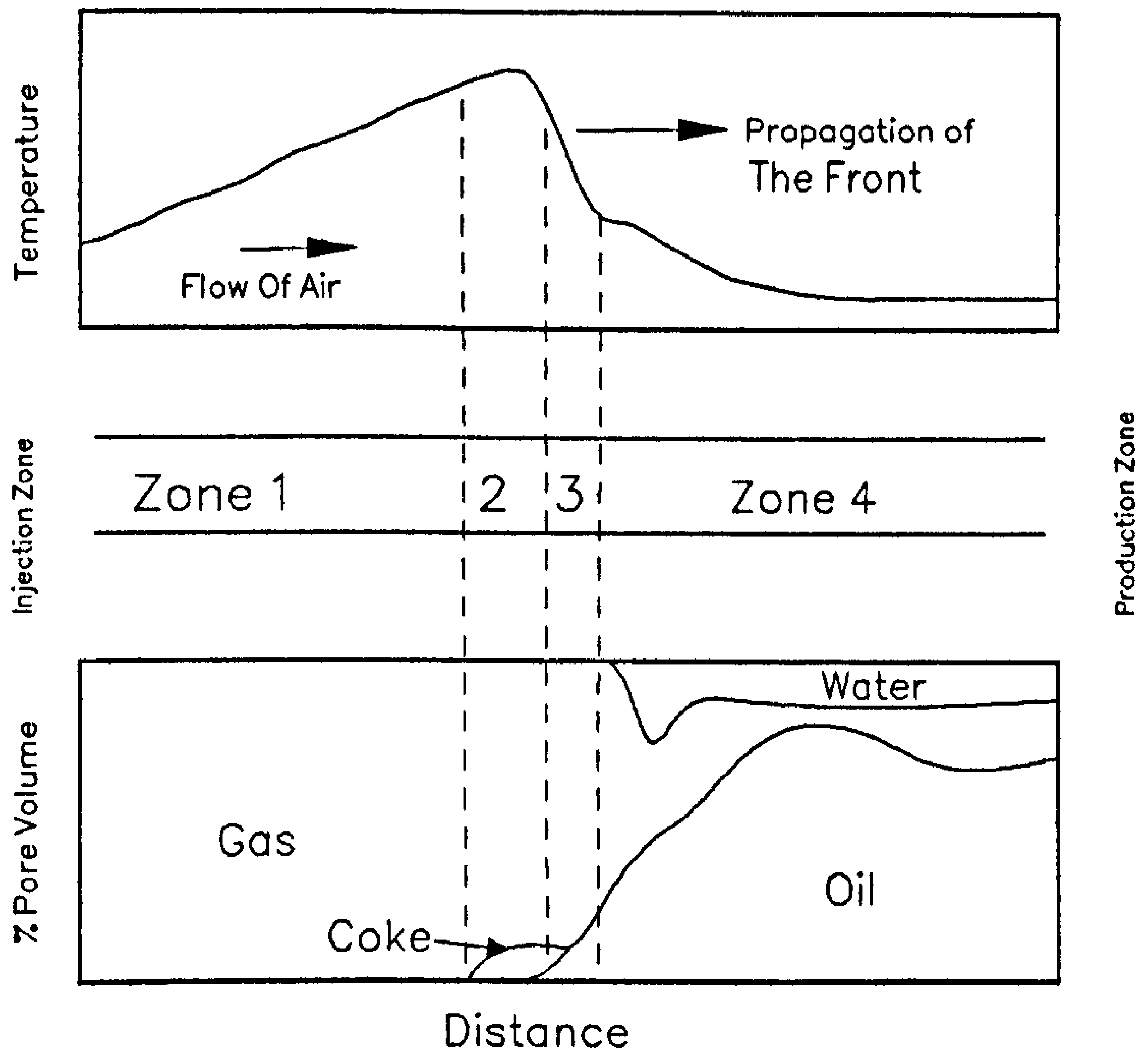


Figure 1.1 Forward Combustion

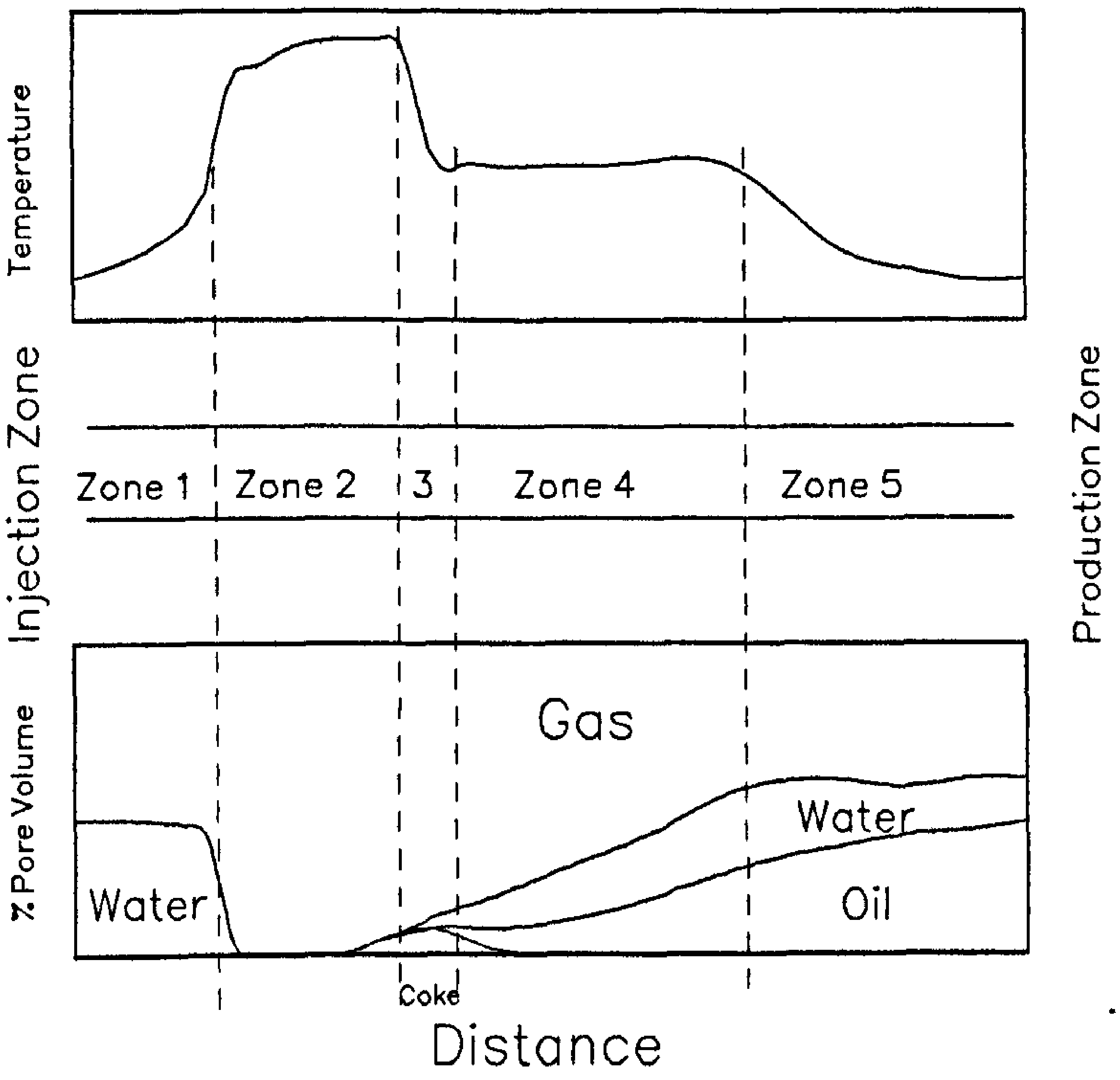


Figure 1.2 Wet Combustion

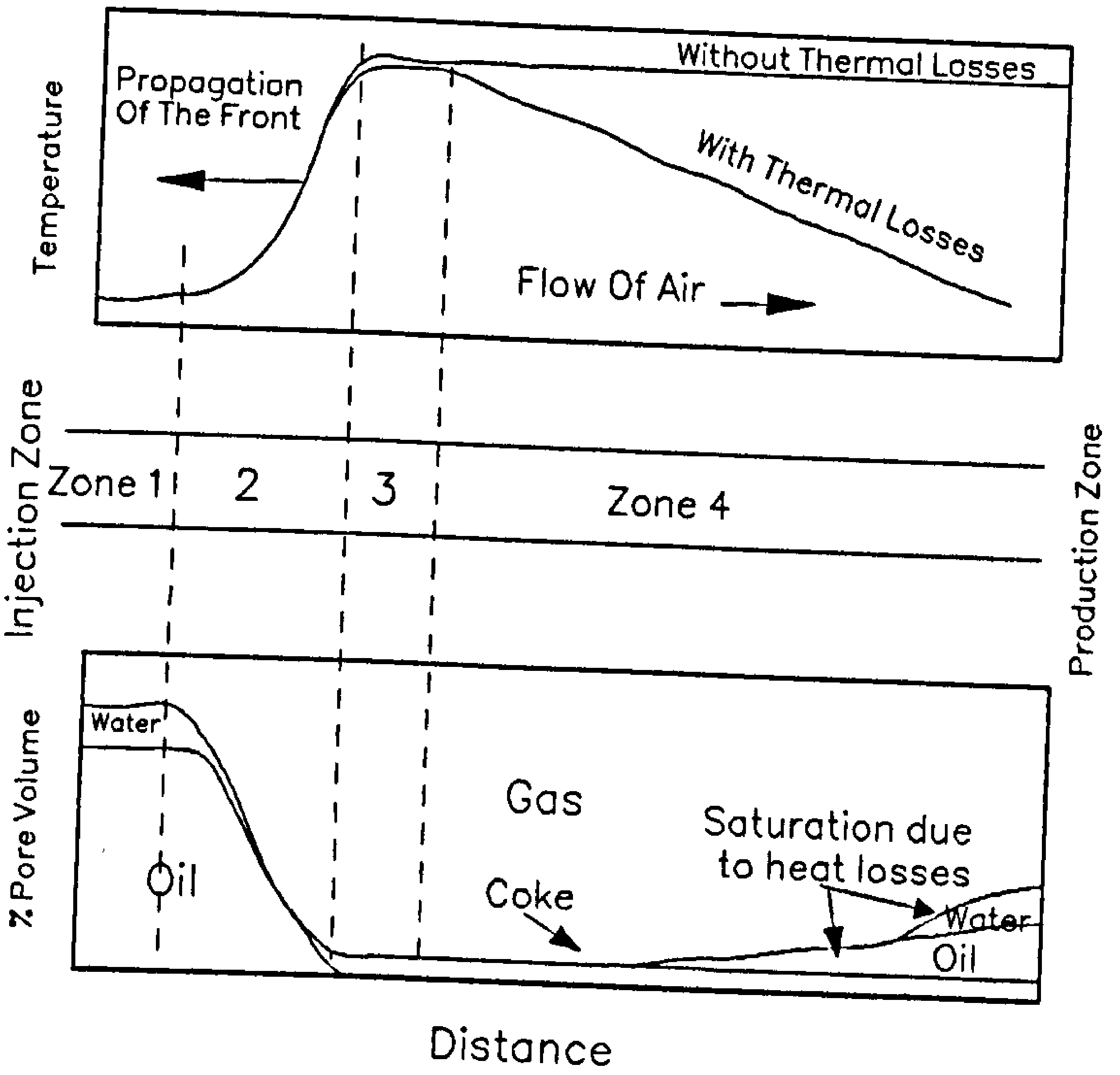


Figure 1 .3 Reverse Combustion

1.3 Historical Background On The Modeling Of In-Situ Combustion

The first attempts to model in-situ combustion were made by Ramey [1959] and Bailey and Larkin [1959], who assumed conduction to be the only mode of heat transfer in an infinite system. Later, Bailey and Larkin [1960] included convection within the formation, but neglected heat losses to the surrounding strata.

Chu [1963] developed a numerical radial flow model of combustion, taking into account conduction and convection within the reservoir, and also conductive heat losses to the adjacent formation. Thomas [1963] also presented an analytical treatment of conduction and convection combustion in an infinite system.

One of the first comprehensive models of the in-situ combustion process was that by Gottfried [1965]. It included three phase fluid flow, a heat generating reaction rate equation and vaporization and condensation of water. A one dimensional system was employed with heat losses from the reservoir by conduction approximated by a convective type heat transfer expression. Although all the major thermal and hydrodynamic characteristics of the in-situ combustion process, including the propagation of the combustion front, the formation of the steam plateau, and the formation of water and oil banks, were included in this model, it still had many limitations. For instance, the simulation required vast amounts of computer time as admitted by the author. In addition, gravity and capillary effects were not included and oil was considered to be the only fuel source with no allowance for coke formation and oxidation. Water and steam were not considered to be in equilibrium, and no oil vaporization was allowed. However, this model did represent a major advance in the ability to simulate the phenomena observed during recovery.

The above model was used by Couch and Rodriguez [1970] in a study on the effect of permeability and porosity on fuel content.

A two dimensional, single phase in-situ combustion model was developed by Smith and Farouq Ali [1971]. This model was based upon the concept of a moving front, exactly tracked, with a changing mobility ahead and behind the front and compressible fluid flow. The model assumes a single phase flow and a constant fuel consumption. Later, Eggenschwiler and Farouq-Ali [1977] presented an improved version of this model.

El-Khatib [1973] presented a multi-phase, multi-dimensional model. The author introduced a set of dimensionless variables and the physical significance of the resulting dimensionless groups was discussed.

Adler [1975] presented a new mathematical model for in-situ combustion. The model included the following physico-chemical processes; combustion of the oil components and their vapours, evaporation of the liquids, solution of the combustion gas in the liquids and their effect on the viscosities and heat conduction. The author used Gottfried's model as the starting point for the physico-chemical aspects, but the mathematical methods used were different.

Farouq Ali [1977] developed a two-dimensional four-component (one oil component) combustion model. The effects of solubility of hydrocarbon gases in the liquids, water vaporization and condensation under equilibrium conditions, and solubility of oxygen, nitrogen, and carbon dioxide mixtures in the oil phase were accounted for. This model did not include the effects of coke formation and oxidation. The author used an implicit pressure explicit saturation (IMPES) solution method for the system of equations. This method is slow and a more implicit formulation was recommended by the author.

A multi-phase, multi-dimensional flow model of in-situ combustion has been produced by Crookston, Culham, and Chen [1979]. Heat transfer was assumed to take place by conduction and convection within the reservoir while heat losses to the surrounding rocks occurred by conduction. The effects of capillary pressure and gravity were included. Vaporization and condensation of both water and oil were estimated from vapour-liquid equilibria. The oil was divided into two pseudo components and four chemical reactions were used to represent respectively, the formation of coke from the heavy pseudo component of the oil, the oxidation of coke and oxidation of the heavy pseudo component and the light pseudo components. This was the first consistent model to have considered coke formation as well as oxidation. It is worth noting here that the model developed in the present work is based on the assumptions made in this model.

Acharya [1979] developed a model using the same equations as used by Gottfried [1965] but the model was extended to two dimensions to account for the effect of gravity.

Grabowski, Vinsome, Lin, Bahie, and Rubin [1979] have presented a general purpose thermal model. This is a three-dimensional, very general simulator, with a fully implicit formulation. It allowed a variable number of components to be present in four phases (water, oil, gas, and coke) with a variable number of chemical reactions. This model will be used to validate the present work by comparing the results of examples run on both simulators.

Later, Coats [1980] presented a comprehensive three-dimensional model which caters for any component and several reactions. Gravity and capillary forces were incorporated along with vaporization-condensation phenomena which allowed for any component to be distributed in any or all of the four phases. The formation permeability was allowed to vary with changing coke concentration. Coats obtained excellent agreement with previously published experimental data.

Numerical simulation results for in-situ combustion in one dimension were reported by Rubin and Vinsome [1980]. They treated coke concentration by following the approach reported by Crookston et al [1979].

A three dimensional, reactant controlled simulator was presented by Youngren [1980]. The author argued that there is insufficient quantitative data on combustion kinetics of crude oil in porous media to allow a rigorous treatment of reaction kinetics and that the assumption of a combustion rate limited by the oxygen flux is simple yet realistic. However, due to this assumption the simulator could not model partially quenched wet combustion or low temperature spontaneous ignition. Coke formation, oxidation, and thermal cracking were left out of the formulation. Consequently, fuel lay down could not be predicted. Relative permeability and capillary pressure were included as a function of saturation and temperature. The author obtained excellent agreement with experimental in-situ combustion data and was the first to report field-scale reservoir simulation results.

Hwang, Jines, and Odeh [1980] introduced a new three phase black oil simulator for in-situ combustion processes (as well as steam and hot water injection operations). The model treats the burning zone as a moving front. The front is viewed not only as a moving heat source, but also as a displacement pump enhancing the oil flow.

The model developed by Crookston et al [1979] was used by Lin, Chen, Lee, and Culham [1984] to investigate the mechanism of fuel deposition and to identify the key kinetic parameters affecting the performance of a dry, forward combustion process. They reported the necessity to further split the light oil components into two or more components due to the sensitivity of the simulation results to the phase equilibrium K-values.

Anis, Hwang, and Odeh [1983] studied the problems related to grid block size and dimensionality. They suggested that in order to overcome such problems, that use be made of pseudo-functions for relative permeability and kinetic parameters as used previously by Camey and Emanuel [1977] in the context of CO_2 flooding.

A fully implicit four-phase numerical reservoir model for simulating in-situ combustion (as well as hot water injection and steam injection) in multi-dimensional (cartesian, radial, or curvilinear) form was presented by Rubin and Buchannan [1985]. They also used a fully implicit well model. The variation in relative permeability and capillary pressure with temperature was allowed for in this model.

Vaughn [1986] published details of a one-dimensional, multi-phase model, essentially using the same equations as Crookston et al [1979] except that a Newton-Raphson scheme with a variable time step was used. The author admitted that this model was not written to compete with more general models due to the large number of assumptions made. Genrich and Pope [1988] presented a one-dimensional, multi-phase in-situ combustion model. This model was written to be used to compare the production response from process modes (dry, wet, and enriched) and to determine the optimum combination of process parameters. The model considers only four homogeneous zones (burned, combustion, steam, and cold). Movement and saturations of each zone are calculated on a time step basis from the solution of the fractional-flow equations by the method of characteristics.

A field-scale simulator was presented by Ito and Chow [1988]. They introduced a pseudo kinetic scheme in this model and in addition the effects of viscous fingering and channeling were accounted for.

Thies and Lemonnier [1988] developed a numerical simulator for modelling in-situ combustion using four components (water, oil, oxygen, and inert gas) and one reaction for fuel combustion. They used a new formulation using a heat release curve to represent the combustion front. This technique improved the numerical stability and gave accurate temperature distributions with large grid block sizes.

In order to tackle the grid size problem, the thin flame technique was applied to a conventional in-situ combustion simulator by Davies [1988]. This involved representing the flame as an infinitely thin interface between the injection fluids and the oil and exhaust gas. The thin flame moves through the region acting as a heat source and displacing pump. A steady state analysis was used to obtain the flame velocity and reaction rate which are used to advance the flame the required distance and supply the reaction rate for the burning grid block over the next time step.

Belegrave, Moore, Ursenbach, and Bennion [1990] described a unified-mechanistic reaction model for mathematical modeling of in-situ combustion of Athabasca bitumen. The formulation is comprehensive in that it allows bitumen to undergo density and viscosity increases, as well as reduced reactivity to oxidation, with increased extent of oxidation. Hydrocarbon bypassing due to quenching of the combustion front is also permitted with the proposed kinetic model.

1.4 The Present Work

All the general models mentioned above are available at high cost difficult to fund in academic institutions; therefore, it is the purpose of this work to develop a model which is not intended to compete with the general private models, but to provide a comprehensive, robust, and efficient model accessible to the academic community. This model should be capable of general use and be able to be modified by the user.

Chapter Two

DESCRIPTION OF THE MODEL

2.1 Introduction

The complex phenomena which occur in the in-situ combustion process are far from being fully clarified. Numerical simulation is a powerful tool which can help to identify some of the key features of this process. Such a simulation can provide a model to predict the performance of a reservoir and which can be used to make intelligent decisions regarding future reservoir operations.

The development of a robust mathematical model requires considerable knowledge of several different areas. Among these are, i) reservoir engineering principles, ii) the theory of partial differential equations, iii) numerical analysis techniques, iv) matrix algebra with matrix computations, and v) computer programming. A proper combination of all of these areas is necessary to produce a satisfactory mathematical model.

Any numerical model of in-situ combustion should account for the major phenomena that occur in the process including: multi-phase flow, mass transfer among the phases, reaction kinetics, and equilibrium between the liquid and vapour for the various components.

This chapter provides a full description of the model, the system of equations of which it consists , and the numerical method used to solve that system.

2.2 The Physical System

The simulator is a three phase flow simulator. It has six components (oxygen, inert gas, a light oil, a heavy oil, water, and coke). The inert gas consists of all the noncondensible gases other than the oxygen and will therefore include oxides of carbon. Water is assumed to be immiscible with oil. The vaporization and condensation of both water and oil is governed by vapour-liquid equilibria, using temperature and pressure dependent equilibrium coefficients. Splitting of the oil into fractions is done to include the distillation effect.

The model includes four chemical reactions. These represent in turn the oxidation of the light and the heavy oil, the formation of coke from the heavy oil, and the oxidation of the coke. They are designated as reaction A, B, C, and D respectively. The model can be modified by the user to include other different reactions.

Heat transport is assumed to take place by convection and conduction; the radiation effect is ignored because the temperatures at which the process takes place are not high enough to make any radiation effects significant. The effects of gravity, capillary pressure (capillary pressure may be defined as the pressure difference across a curved interface between two immiscible fluids) between any two fluid phases, and heat loss to the surrounding rock via conduction only are also included.

2.3 Mathematical Description

The model consists of the following equations;

a) The Conservation Equations

Seven conservation equations are included in the model. These equations are;

(i) Oxygen Mass Balance

$$- \nabla (\rho_g Y_1 V_g) + q_1 - (s_1 r_A + s_4 r_B + s_{10} r_D) = \frac{\partial}{\partial t} (\phi S_g \rho_g Y_1) \tag{2.1}$$

(ii) Inert Gas Mass Balance

$$- \nabla (\rho_g Y_2 V_g) + q_2 - (s_1 r_A + s_5 r_B + s_9 r_C + s_{11} r_D) =$$

$$\frac{\partial}{\partial t} (\phi S_g \rho_g Y_2) \tag{2.2}$$

(iii) Light Oil Mass Balance

$$- \nabla (\rho_g K_3^* V_g + \rho_o V_o) X_3 + q_3 - (r_A - s_7 r_C) =$$

$$\frac{\partial}{\partial t} [\phi(S_g \rho_g K_3^* + S_o \rho_o)X_3] \tag{2.3}$$

(iv) Heavy Oil Mass Balance

$$- \nabla (\rho_g K_4^* V_g + \rho_o V_o) X_4 + q_4 - (r_B + r_c) =$$

$$\frac{\partial}{\partial t} [\phi(S_g \rho_g K_4^* + S_o \rho_o) X_4] \tag{2.4}$$

(v) Water Mass Balance

$$- \nabla (\rho_g K_5^* V_g + \rho_o V_o) + q_5 - (s_3 r_A + s_6 r_B + s_{12} r_D) =$$

$$\frac{\partial}{\partial t} [\phi(S_g \rho_g K_5^* + S_w \rho_w)] \tag{2.5}$$

(vi) Coke Mass Balance

$$s_8 r_C - r_D = \frac{\partial C_c}{\partial t} \tag{2.6}$$

(vii) Energy Balance

$$\nabla (K_T \nabla T) - \nabla (\rho_g h_g V_g + \rho_o h_o V_o + \rho_w h_w V_w) +$$

$$(q_g h_g^* + q_o h_o^* + q_w h_w^*) + (H_A r_A + H_B r_B + H_C r_C + H_D r_D) =$$

$$\frac{\partial}{\partial t} [(1 - \phi)\rho_r U_r + \phi(S_g \rho_g U_g + S_o \rho_o U_o + S_w \rho_w U_w) + C_c U_c] \tag{2.7}$$

where C_c is the coke concentration (gmol/m^3)

K_3^* is the light oil pseudo vapour/liquid equilibrium constant which will be defined in later section

K_4^* is the heavy oil pseudo vapour/liquid equilibrium constant

K_5^*	is the water pseudo vapour/liquid equilibrium constant	
K_T	is the formation overall thermal conductivity (kJ/m.day.K)	
h_i	is the phase enthalpy (kJ/gmol)	$i = g,o,w$
h_j^*	is the phase injection or production enthalpy (kJ/gmol)	$j = g,o,w$
H_j	is the heat of reaction (kJ/gmol)	$j = A,B,C,D$
q_i	is the component injection or production rate (gmol/day.m^3)	$i = 1,2,\ldots,5$
q_j	is the phase injection or production rate (gmol/day.m^3)	$j = g,o,w$
r_j	is the reaction rate (gmol/m^3.day)	$j = A,B,C,D$
s_i	is the reaction stoichiometry	$i = 1,2,\ldots,12$
S_i	is the phase saturation	$i = g,o,w$
T	is the formation temperature (K)	
U_i	is the phase internal energy (kJ/gmol)	$i = g,o,w,c,r$
V_i	is the phase velocity (m/day)	$i = g,o,w$
X_i	is the mole fraction of component i in the oil phase	$i = 3,4$
Y_i	is the mole fraction of component i in the gas phase	$i = 1,2,\ldots,5$
ρ_i	is the phase density (gmol/m^3)	$i = g,o,w,r$
ϕ	is the porosity	

where the subscripts g, o, w, c, and r refer to gas, oil, water, coke, rock,

A, B, C, D refer to the four reactions,

and 1, 2, 3, 4, and 5 refer to oxygen, inert gas, light oil, heavy oil, and water respectively.

Since coke is assumed to be immobile, it can be seen that its mass balance equation contains no flow terms (i.e. no 'interblock' terms), so it can be eliminated from the seven primary conservation equations. Details of this elimination are shown in Appendix A.

Darcy's law can be used to calculate the phase velocities as follows;

Gas Phase Velocity

$$V_g = - \frac{K\,K_{rg}}{\mu_g} \frac{\partial P_g}{\partial x} \qquad (2.8)$$

$$V_g = - \frac{K\,K_{rg}}{\mu_g} \frac{\partial P_g}{\partial y} \qquad (2.9)$$

$$V_g = - \frac{K\,K_{rg}}{\mu_g} \left(\frac{\partial P_g}{\partial z} - \rho_g\, g\right) \qquad (2.10)$$

Oil Phase Velocity

$$V_o = - \frac{K\,K_{ro}}{\mu_o} \frac{\partial P_o}{\partial x} \qquad (2.11)$$

$$V_o = - \frac{K\,K_{ro}}{\mu_o} \frac{\partial P_o}{\partial y} \qquad (2.12)$$

$$V_o = - \frac{K\,K_{ro}}{\mu_o} \left(\frac{\partial P_o}{\partial z} - \rho_o\, g\right) \qquad (2.13)$$

Water Phase Velocity

$$V_w = - \frac{K\,K_{rw}}{\mu_w} \frac{\partial P_w}{\partial x} \qquad (2.14)$$

$$V_w = - \frac{K\ K_{rw}}{\mu_w} \frac{\partial P_w}{\partial y} \qquad (2.15)$$

$$V_w = - \frac{K\ K_{rw}}{\mu_w} \left(\frac{\partial P_w}{\partial z} - \rho_w\ g\right) \qquad (2.16)$$

where K is the absolute permeability (m^2)

K_{ri} is the phase relative permeability i=o,w,g

x, y, and z are the spatial variables (m)

μ_i is the phase viscosity (kPa.day) i=o,w,g

P_i is the phase pressure (kPa) i=o,w,g

b) The Reaction Rate Equations

The model includes the following four reactions:

i) Light Oil Oxidation

$$Oil_3 + s_1O_2 \xrightarrow{r_A} s_2CO_x + s_3H_2O$$

ii) Heavy Oil Oxidation

$$Oil_4 + s_4O_2 \xrightarrow{r_B} s_5CO_x + s_6H_2O$$

iii) Heavy Oil Cracking

$$Oil_4 \xrightarrow{r_C} s_7Oil_3 + s_8Coke + s_9Gas_2$$

iv) Coke Oxidation

$$\text{Coke} + s_{10}O_2 \xrightarrow{r_D} s_{11}CO_x + s_{12}H_2O$$

The "s" coefficients are determined by stoichiometry. The kinetic reaction rate expressions which are taken from Crookston et al [1979] and used for the above reactions are:

$$r_A = A_A\, e^{(-E_A/RT)}\, (Y_1)\, (P_o + P_{cg})(\phi\, S_o\, \rho_o\, X_3) \tag{2.17}$$

$$r_B = A_B\, e^{(-E_B/RT)}\, (Y_1)\, (P_o + P_{cg})(\phi\, S_o\, \rho_o\, X_4) \tag{2.18}$$

$$r_C = A_C\, e^{(-E_C/RT)}(\phi\, S_o\, \rho_o\, X_4)(1 - (\frac{C_c}{C_{Cmax}})^5) \tag{2.19}$$

$$r_D = A_D\, e^{(-E_D/RT)}\, (Y_1)\, (P_o + P_{cg})(C_c) \tag{2.20}$$

where A_j are the reaction Arrhenius constants j=A,B,C,D

E_j are the reaction activation energies j=A,B,C,D

and P_{cg} is the gas phase capillary pressure (kPa)

The term $(1 - (\frac{C_c}{C_{Cmax}})^5)$ in equation (2.19) is used in order to prevent the coke from exceeding a specified maximum.

c) The Algebraic Constraints

The set of unknown variables in the primary conservation equations can be reduced to six primary variables using the algebraic constraints, i.e. one saturation constraint, two mole fraction constraints, three phase equilibrium ratios, and two capillary pressure constraints.

Saturation constraint

$$S_g + S_w + S_o = 1.00 \tag{2.21}$$

Mole Fraction Constraints

$$Y_1 + Y_2 + Y_3 + Y_4 + Y_5 = 1.00 \tag{2.22}$$

$$X_3 + X_4 = 1.00 \tag{2.23}$$

Phase Equilibrium Ratios

$$Y_3 = K_3^* X_3 \tag{2.24}$$

$$Y_4 = K_4^* X_4 \tag{2.25}$$

$$Y_5 = K_5^* \tag{2.26}$$

Capillary pressure constraints

$$P_{cw} = P_o - P_w \tag{2.27}$$

$$P_{cg} = P_g - P_o \tag{2.28}$$

where P_{cw} is the water phase capillary pressure (kPa)

P_{cg} is the gas phase capillary pressure (kPa)

The Equilibrium Constants are expressed as a ratio of component vapour pressure to the system gas pressure. The vapour pressure dependence on temperature is accounted for by the Antoine equation (Smith and Van Ness [1959]).

$$K_3 = \frac{1}{P_g} e^{\left(\frac{a_3 - b_3}{T + c_3}\right)} \tag{2.29}$$

$$K_4 = \frac{1}{P_g} e^{\left(\frac{a_4 - b_4}{T + c_4}\right)} \tag{2.30}$$

$$K_5 = \frac{1}{P_g} \left[\frac{T}{a_5}\right]^{b_5} \tag{2.31}$$

where a_i, b_i, and c_i are constants.

In order to avoid the reformation of the partial differential equations for the special case of phase disappearance and formation, Crookston's concept of pseudo K factors is used for the water and oil components.

Therefore,

$$K_3^* = K_3 \left[\frac{S_o}{S_o + \varepsilon}\right] \tag{2.29a}$$

$$K_4^* = K_4 \left[\frac{S_o}{S_o + \varepsilon}\right] \tag{2.30a}$$

$$K_5^* = K_5 \left[\frac{S_w}{S_w + \varepsilon}\right] \tag{2.31a}$$

Where ε is a small number of order 10^{-4}. The bracketed terms essentially prevent the liquid phases from disappearing completely.

d) Production/Injection Term

The production rate for each phase is calculated as follows:

$$q_g = \frac{\alpha\ K\ h}{V} \ \frac{\rho_g\ K_{rg}}{\mu_g} \ (P_g - P_{well}) \tag{2.32}$$

$$q_o = \frac{\alpha\ K\ h}{V} \ \frac{\rho_o\ K_{ro}}{\mu_o} \ (P_o - P_{well}) \tag{2.33}$$

$$q_w = \frac{\alpha\ K\ h}{V} \ \frac{\rho_w\ K_{rw}}{\mu_w} \ (P_w - P_{well}) \tag{2.34}$$

where h is the grid block height (m)

P_{well} is the well back pressure (kPa)

V is the grid block volume (m^3)

α is the productivity index

The production rates of a component are calculated from these phase production rates by multiplying by the mole fraction of that component in each phase.

The injection rate for each component is specified as well as other information such as temperature and phase in order to calculate the enthalpy.

e) Heat loss

The rate of the heat loss to the surrounding rocks is calculated using a semi-analytical method proposed by [Vinsome and Westerveld (1980)]. The method uses a fitting function

$$\varphi(t,z) = (T + c_1 z + c_2 z^2)\, e^{-z/d} \tag{2.35}$$

and the equation for heat flow into the cap rock

$$\frac{\partial \varphi}{\partial t} = k \frac{\partial^2 \varphi}{\partial z^2} \tag{2.36}$$

where, φ is the cap rock temperature, z is the distance into the cap rock, d is a diffusion length $\frac{\sqrt{kt}}{2}$, T is the cell temperature, and c_1 and c_2 are parameters determined by the analysis.

This method has the advantage of being much simpler and more economical than a finite difference calculation of the heat loss.

f) The Physical Properties

The following equations are used to calculate the physical properties;

1- Phase Densities

i) Gas Density

The gas phase density is calculated as;

$$\rho_g = \frac{P_g}{ZRT} \tag{2.37}$$

where Z is the average compressibility factor for the gas mixture.

ii) Water Density

Water density is calculated from the following equation:

$$\rho_w = \rho_{wR} \, [1 + \alpha_w (P_w - P_R) - \beta_w (T - T_R)] \tag{2.38}$$

where ρ_{wR} is the water density at the reference pressure and temperature.

α_w is the water compressibility factor $(\mathrm{kPa})^{-1}$

β_w is the water thermal expansion factor $(\mathrm{K})^{-1}$

iii) Oil Density

Oil density is calculated from the following equation:

$$\rho_o = \rho_{oR} \, [1 + \alpha_o (P_o - P_R) - \beta_o (T - T_R)] \tag{2.39}$$

where ρ_{oR} is the oil density at the reference pressure and temperature and is calculated using Amagat's law of partial volumes;

$$\rho_{oR} = \left(\frac{X_3}{\rho^3_{oR}} + \frac{X_4}{\rho^4_{oR}} \right)^{-1} \tag{2.40}$$

and α_o is the oil compressibility factor $(\mathrm{kPa})^{-1}$

β_o is the oil thermal expansion factor $(\mathrm{K})^{-1}$

ρ^3_{oR} and ρ^4_{oR} are the light and heavy oil density at reference pressure and temperature.

2- Viscosity

i) Gas Viscosity

The gas phase viscosity is related to temperature and composition according to;

$$\mu_g = \sum_{i=1}^{5} Y_i \, \mu_{1i} \, (T)^{\mu_{2i}} \qquad (2.41)$$

where μ_{1i} and μ_{2i} are constants $\qquad i=1,2,\ldots\ldots,5$

ii) Water Viscosity

The water phase viscosity is related to absolute temperature by;

$$\mu_w = 10^{-9} /(12.1 + 2.88(T - 273.15) + 7.78\text{x}10^{-4}(T - 273.15)^2) \qquad (2.42)$$

iii) Oil Viscosity

The oil phase viscosity is calculated using the following dependence on composition;

$$\mu_o = (\mu_{o3})^{X_3} \, (\mu_{o4})^{X_4} \qquad (2.43)$$

where μ_{o3} and μ_{o4} are the light and heavy oil viscosities which are calculated using Andrade's equation for temperature dependence (Watson [1943]);

$$\mu_{o3} = AV_3\ e^{(BV_3/T)} \tag{2.44}$$

$$\mu_{o4} = AV_4\ e^{(BV_4/T)} \tag{2.45}$$

where AV_3, AV_4, BV_3, and BV_4 are constants.

3- Relative Permeability

The relative permeability of each phase is calculated from the following empirical equations;

$$K_{rw} = K_{rwro}[(S_w - S_{wc})/(1 - S_{wc} - S_{orw})]^{Z_w} \tag{2.46}$$

$$K_{rg} = K_{rgro}[(S_g - S_{gc})/(1 - S_{wc} - S_{org} - S_{gc})]^{Z_g} \tag{2.47}$$

$$K_{rog} = K_{rocw}[1 - (S_g - S_{gc})/(1 - S_{wc} - S_{org} - S_{gc})]^{Z_{og}} \tag{2.48}$$

$$K_{row} = K_{rocw}[1 - (S_w - S_{wc})/(1 - S_{orw} - S_{wc})]^{Z_{ow}} \tag{2.49}$$

$$K_{ro} = K_{rocw}[(K_{row}/K_{rocw} + K_{rw})(K_{rog}/K_{rocw} + K_{rg}) - K_{rg} - K_{rw}] \tag{2.50}$$

where K_{rg} is the gas relative permeability

K_{rw} is the water relative permeability

K_{ro} is the oil relative permeability

K_{rog} is oil-gas two phase relative permeability to oil

K_{row} is oil-water two phase relative permeability to oil

K_{rgro} is oil-gas two phase relative permeability to water at residual oil saturation

K_{rwro} is oil-water two phase relative permeability to gas at residual oil saturation

K_{rocw} is oil-water two-phase relative permeability to oil at connate water saturation

S_{wc} is connate water saturation

S_{gc} is critical gas saturation

S_{org} is residual oil saturation for oil-gas system

S_{orw} is residual oil saturation for oil-water system

Z_i are fitting constants

4- Enthalpy and Internal Energy

a-Gas Enthalpy

The gas phase enthalpy is calculated as;

$$h_g = \int_{298}^{T} Cp_g \, dT \tag{2.51}$$

$$Cp_g = \bar{a} + \bar{b}T + \bar{c}T^2 \tag{2.52}$$

$$\bar{a} = \sum_{i=1}^{5} Y_i \, a_i \tag{2.53}$$

$$\bar{b} = \sum_{i=1}^{5} Y_i \, b_i \tag{2.54}$$

$$\bar{c} = \sum_{i=1}^{5} Y_i \, c_i \tag{2.55}$$

Where a_i, b_i, and c_i are the coefficients of the quadratic function of temperature for the heat capacity of each component.

b- Water Enthalpy

The water phase enthalpy is calculated as;

$$h_w = h_{gw} - h_{vw} \tag{2.56}$$

where h_{gw} is the water enthalpy at gas phase at the specific temperature and is calculated as;

$$h_{gw} = \int_{298}^{T} Cp_w \, dT \tag{2.57}$$

and h_{vw} is the water heat of vaporization at the specific temperature which is calculated as;

$$h_{vw} = 4.814(T_{crw} - T)^{0.38} \qquad \text{for} \qquad T \le T_{crw} \tag{2.58}$$

$$h_{vw} = 0.0 \qquad \text{for} \qquad T > T_{crw}$$

where T_{crw} is the water critical temperature.

c- Oil Enthalpy

The oil phase enthalpy is calculated as;

$$h_o = h_{go} - h_{vo} \qquad (2.60)$$

where h_{go} is the oil enthalpy ~~at~~ of the gas phase at the specific temperature and is estimated by;

$$h_{go} = \int_{298}^{T} Cp_{go}\, dT \qquad (2.61)$$

and h_{vo} is the oil heat of vaporization at the specific temperature and is calculated from:

$$h_{vo} = h^{\circ}_{vo} \, (T_{cro} - T)^{0.38} \qquad (2.62)$$

where h^{o}_{vo} is a constant

T_{cro} is the oil critical temperature.

d- Gas Internal Energy

By definition:

$$U = h - P/\rho \qquad (2.63)$$

so, the gas phase internal energy is;

$$U_g = h_g - P_g/\rho_g \qquad (2.64)$$

e- Internal Energy of other phases

The internal energy of water, oil, coke, and rock are assumed to be equal to their enthalpies.

f- Rock Enthalpy

The rock enthalpy is calculated from;

$$h_r = Cp_r\ (T - 298) \tag{2.65}$$

where Cp_r is constant and may be obtained from data on different rocks.

g- Coke Enthalpy

Coke enthalpy is determined form;

$$h_c = \int_{298}^{T} Cp_c\ dT \tag{2.66}$$

$$Cp_c = a + bT + cT^2 \tag{2.67}$$

5- Capillary Pressures

The capillary pressures are estimated from the following equations as;

$$P_{cw} = AC_{w1} + AC_{w2}\ S_w \tag{2.68}$$

$$P_{cg} = AC_{g1} + AC_{g2}\ S_g \tag{2.69}$$

where AC_{w1}, AC_{w2}, AC_{g1}, and AC_{g2} are constants

2.4 Numerical Solution

The reservoir is divided into a number of grid blocks and the finite difference representation of the six partial differential equations i.e. the conservation equations except the coke equation, are evaluated in each grid block. These equations are discretized using upstream weighting for the spatial variable (x) and a two-point backward difference in time. The discretized equations are shown in Appendix B. Upstream weighting is used in order to ensure convergence to a physically correct solution and to reduce the numerical instability near the sharp combustion fronts [Rubin and Vinsome (1980)].

The discretization leads to six nonlinear algebraic equations for each grid block. A similar alignment of these equations to that of the ISCOM simulator [Grabowski et al (1979)] is used. This alignment is shown in Table 2.1. The set of non-linear algebraic equations is solved using Newton's method. If the equations are written as:

$$F(\vec{X}) = 0 \tag{2.70}$$

then Newton's method can be written as

$$\Phi(\vec{X}_k)(\vec{X}_{k+1} - \vec{X}_k) = - F(\vec{X}_k) \tag{2.71}$$

Where $\vec{X}_k$ is the primary variables vector.

$F(\vec{X}_k)$ = the set of discritized conservation equations.

$\Phi(\vec{X}_k)$ = the Jacobian matrix with the entries i,j being $\left(\frac{\partial F_i(\vec{X}_k)}{\partial X_j}\right)$

Table 2.1 Alignment of Equations With Unknows

Variable	Equation
Pressure	Sum of Noncondensible Gas Equations
Water Saturation	Water Conservation Equation
Oil Saturation	Sum of Oil Component Conservation Equations
Temperature	Energy Balance Equation
Oxygen Mole Fraction	Oxygen Conservation Equation
Heavy Oil Mole Fraction	Heavy Oil Conservation Equation

Therefore for each grid block the Jacobian will have the form:

$$\begin{pmatrix} \frac{\partial F_P}{\partial P} & \frac{\partial F_P}{\partial w} & \frac{\partial F_P}{\partial o} & \frac{\partial F_P}{\partial T} & \frac{\partial F_P}{\partial Y} & \frac{\partial F_P}{\partial X} \\ \frac{\partial F_w}{\partial P} & \frac{\partial F_w}{\partial w} & \frac{\partial F_w}{\partial o} & \frac{\partial F_w}{\partial T} & \frac{\partial F_w}{\partial Y} & \frac{\partial F_w}{\partial X} \\ \frac{\partial F_o}{\partial P} & \frac{\partial F_o}{\partial w} & \frac{\partial F_o}{\partial o} & \frac{\partial F_o}{\partial T} & \frac{\partial F_o}{\partial Y} & \frac{\partial F_o}{\partial X} \\ \frac{\partial F_T}{\partial P} & \frac{\partial F_T}{\partial w} & \frac{\partial F_T}{\partial o} & \frac{\partial F_T}{\partial T} & \frac{\partial F_T}{\partial Y} & \frac{\partial F_T}{\partial X} \\ \frac{\partial F_Y}{\partial P} & \frac{\partial F_Y}{\partial w} & \frac{\partial F_Y}{\partial o} & \frac{\partial F_Y}{\partial T} & \frac{\partial F_Y}{\partial Y} & \frac{\partial F_Y}{\partial X} \\ \frac{\partial F_X}{\partial P} & \frac{\partial F_X}{\partial w} & \frac{\partial F_X}{\partial o} & \frac{\partial F_X}{\partial T} & \frac{\partial F_X}{\partial Y} & \frac{\partial F_X}{\partial X} \end{pmatrix} \tag{2.72}$$

In reservoir simulation the discretized equation depends also on the primary variables surrounding the particular grid block under consideration, so the full matrix equation will take the following form:

$$\begin{pmatrix} X & X & & & \\ X & X & X & & \\ & & \ddots & & \\ & & & & X \\ & & & X & X \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \vdots \\ \delta_N \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ \vdots \\ F_N \end{pmatrix} \tag{2.73}$$

where X is a submatrix which has the form of (2.72) and each δ_i and F_i have the form:

$$\delta_l = \begin{pmatrix} P_{k+1} - P_k \\ S_{w_{k+1}} - S_{w_k} \\ S_{o_{k+1}} - S_{o_k} \\ T_{k+1} - T_k \\ Y_{1_{k+1}} - Y_{1_k} \\ X_{4_{k+1}} - X_{4_k} \end{pmatrix} \qquad F_l = \begin{pmatrix} -F_P \\ -F_w \\ -F_o \\ -F_T \\ -F_Y \\ -F_X \end{pmatrix} \tag{2.74}$$

The derivatives required to evaluate the Jacobian matrix are not evaluated algebraically because this is a very complicated and time consuming process. Instead secant approximations to the Jacobian are used.

Simply if $F(P_g,S_o,S_w,T,Y_1,X_4)$ is the mass balance of a component or the energy balance equation the $\frac{\partial F}{\partial P_g}$ can be approximated by the expression;

$$(F(P_g+\Delta P_g,S_o,S_w,T,Y_o,X_4) - F(P_g,S_o,S_w,T,Y_1,X_4))/\Delta P_g \tag{2.75}$$

Similarly, this process can be done for all the variables.

It has been found that for a specific time step, the changes occurring during the iterations are not sufficient to alter the Jacobian in a way which will affect the solution. So, the Jacobian does not need to be calculated at each iteration and one update in the first iteration at each time step has been found to be adequate.

Matrix decomposition is used to solve the matrix. This process involves the transformation of the matrix into other matrices which are generally easier to operate on and then using these transformed matrices to obtain the solution. The method attributed to Crout [1941] is used to decompose the matrix to a lower and an upper triangular matrix. This decomposition is followed by a back substitution which computes the answer in two successive substitution steps. The process of decomposition is as follows.

Given:

$$AP = b \tag{2.76}$$

then

$$A = L\,U \tag{2.77}$$

where L is

$$\begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ l_{31} & l_{32} & l_{33} & \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ l_{n1} & l_{n2} & & l_{nn} \end{bmatrix} \tag{2.78}$$

and U is

$$\begin{bmatrix} 1 & u_{12} & u_{13} & & u_{1n} \\ & 1 & u_{23} & & \\ & & 1 & & \\ & & & & \cdot \\ & & & & \cdot \\ & & & & \cdot \\ & & & & 1 \end{bmatrix} \tag{2.79}$$

From these two triangular matrices we can compute the solution vector P because, by Eq.(2.77),

$$A = LU \tag{2.80}$$

Then after the decomposition:

$$LUP = b \tag{2.81}$$

Let us call the product UP the vector y. Then from Eq.(2.81):

$$Ly = b \tag{2.82}$$

and

$$UP = y \tag{2.83}$$

Solving Eqs.(2.82) and (2.83) successively for y and b respectively:

$$y = L^{-1}b \tag{2.84}$$

and

$$P = U^{-1}y \tag{2.85}$$

The algorithm for this process is as follows.

The elements of the lower triangular matrix are l_{ij} and $l_{ij} = 0$ if $i<j$, and the elements of the corresponding upper matrix are u_{ij} with $u_{ii} = 1$ and $u_{ij} = 0$ if $i>j$. Then:

$$l_{im} = a_{im} - \sum_{k=1}^{m-1} l_{ik}\, u_{km} \qquad \text{for } i = m, m+1, \ldots, n$$

$$m = 1, 2, \ldots, n \qquad (2.86)$$

$$u_{mj} = \frac{1}{l_{mm}} \left(a_{mj} - \sum_{k=1}^{m-1} l_{mk}\, u_{kj} \right) \qquad \text{for } j = m, m+1, \ldots, n$$

$$m = 1, 2, \ldots, n \qquad (2.87)$$

Back substitution may then be made to give;

$$y_i = \frac{b_i - \sum_{k=1}^{i-1} l_{ik}\, y_k}{l_{ii}} \qquad \text{for } i = 1, 2, \ldots, n \qquad (2.88)$$

and

$$P_i = y_i - \sum_{k=i+1}^{n} u_{ik}\, P_k \qquad \text{for } i = n, n-1, \ldots, 1 \qquad (2.89)$$

Note that in the lower decomposition process in Eq.(2.86) and (2.87) the evaluation of the first summation produces:

$$l_{11} = a_{11} - \sum_{k=1}^{0} l_{11}\, u_{11} \qquad (2.90)$$

The summation

$$\sum_{k=1}^{0} l_{11}\, u_{11} = 0 \qquad (2.91)$$

is defined as zero; thus, the first element of the lower matrix is a_{11}, and this calculation will be bypassed in the program.

Underrelaxation of the process is done by multiplying the right hand side of Newton's equation by a damping factor ω ($0<\omega<1$) in order to reduce any exaggerated change which may be predicted by the first couple of iterations.

Newton's iteration continues until all the dependent variables in all grid blocks satisfy a convergence criteria or until a maximum iteration count is exceeded. The convergence criteria are met if the old and current iterated values agree within a specified number of digits. If convergence is not achieved within the specified number of iterations or if any the values of any dependent variable lie outside a specified physical limit, the calculations are repeated with a halved time step. If the time-step becomes less than a specified minimum value because of the halvings, the simulation is aborted.

Time-step size adjustment is required because the time rate of change in the primary variables varies significantly during a typical simulation. The method used for selecting the time-step size is the same as that of ISCOM, which is

$$\Delta t_i^{n+1} = \Delta t^n \; \frac{2\Delta\Psi_i}{\Delta\Psi_i + \delta\Psi_i} \tag{2.92}$$

Where $\Delta\Psi_i$ is the largest change expected in the dependent variable i at a single time step and is read from input.

$\delta\Psi_i$ is the change that occurred.

The time-step chosen is the minimum of all Δt_i's provided that $\Delta t_{min} < \Delta t_i < \Delta t_{max}$. Δt_{min} and Δt_{max} are read from input.

Finally, a complete description and listing of the program as well as the computers used is given in Appendix C.

Chapter Three

MODEL APPLICATIONS

3.1 Introduction

In order to test the validity, robustness, and efficiency of the simulator, a one dimensional simulation was carried out using data constructed from Crookston et al [1979], Grabowski et al [1979], and Vaughn [1986]. This case is used because of its historic use as a bench mark test for model validation and a complete bed and material characterization is available which allows all the major features of the model to be tested.

The physical validity of the behavior of the main variables is discussed first then the results are compared with results of similar calculations performed using the **ISCOM** simulator. **ISCOM** (Grabowski et al [1979]) is a fully implicit multi-dimensional finite difference general thermal simulator developed by the Computing Modelling Group (**CMG**) at Calgary. The model includes four phases, a variable number of oil components, a variable number of chemical reactions, gravity and capillary pressure terms, and the possibility of modelling emulsions.

Finally, the grid block size problem in reservoir simulation is discussed and the research on the methods used to tackle this problem are reviewed. Also, an optimization for the number of grid blocks for the bench mark problem is carried out.

3.2 Bench Mark Simulation

This simulation is carried out in order to test the physical validity of the behavior of the main variables. It is not intended to model the forward combustion of the actual field test because of the unrealistic characterization of the light and heavy oil.

The simulation is one dimensional in which the permeable formation is 50m long, 35m wide, and 6.4m high. It was partitioned into ten grid blocks. Oxygen was injected at a constant rate of 1.36 10^5 gmol/day into grid block No.1 at the reservoir temperature. Production was from grid block No.10, with the well deliverable against 410 kPa back pressure. The complete set of data used for the run is shown in Appendix D.

Figures 3.1a-d, 3.2a-d, and 3.3a-d present the time variations of the temperature, coke concentration, oil and water saturations, and pressure at grid blocks Nos. 2, 4, and 6 respectively.

It can be seen that as long as liquid water is available in a specific grid block, the block temperature is constrained by the pressure to be at or below the steam temperature. For instance, figure 3.1d shows that the pressure of grid block No.2 after 30 days is 600 kPa. While the saturation temperature at this pressure is found from steam tables to be 432 K, figure 3.1a shows that the temperature of grid block No.2 after 30 days is 422 K which is below the saturation temperature.

Once the water vaporizes the temperature increases rapidly. With the rapid increase in the temperature, the coke oxidation also rapidly increases due to the increase in the rate of the coke oxidation reaction in which temperature is an exponential parameter. Once all coke is consumed the temperature starts decreasing rapidly due to the cooling effect of the injected gas and the heat loss to the surrounding strata.

The coke concentration profiles, figures 3.1b-3.3b, show that there are two peaks for the coke concentration. The first peak is attributed to the slow build up of coke with time. This build up process occurs during the time period when the temperature of the grid block is at or near the steam saturation temperature. From figures 3.1a-3.3a, it can be seen that this period of time increases as the combustion front moves toward the production well. For instance, the temperature of grid block No.2 stays near the steam saturation temperature for ten days (from days 23 to 33), while it stays for 60 days (from days 30 to 90) in grid block No.6. This increase in time explains the increase in the size of the first peak of the coke concentration as a result of the accumulation of more coke.

As the water evaporates and the temperature increases, the accumulated coke starts to be oxidized rapidly. This oxidization results in a further increase in the temperature and consequently the rate of the heavy oil cracking (coke production) increases too, leading to the appearance of the second peak in the coke concentration.

Figures 3.1-3.3c show the appearance of peaks in the water and more noticeable in the oil saturation profiles of grid blocks 2, 4, and 6. These peaks are due to the condensation of the vapour coming from the upstream blocks and also the accumulation of liquid oil and water arriving from the previous blocks having been pushed downstream by the combustion front.

The pressure profile shows fluctuations in the pressure. These fluctuations are due to the water and oil banking which occur downstream of the combustion front. These banks are shown clearly in figure 3.4 and 3.5 which show the water and oil saturation profiles over the reservoir taken at various times.

Figure 3.6a shows a typical oil saturation, water saturation, and temperature profiles over the reservoir taken after 60.5 days. The following zones can be recognized from this figure:

Zone A: A burn-out zone in which the temperature increases continuously from the inlet face temperature (the initial reservoir temperature). Oil and water saturation are zero, so only gas is present in this zone.

Zone B: A combustion zone where the temperature reaches the maximum.

Zone C: Having been pushed downstream by the combustion front, the oil accumulates in this zone and oil saturation reaches peak. Since the temperature is still high in this zone, no water is available.

Zone D: A steam plateau zone. This zone is recognized by the low temperature gradient. The temperature in this zone corresponds to the steam saturation temperature at the prevailing partial pressure of water. Also oil saturation drops in this zone due to the vaporization of some of the oil.

Zone E: A zone of decreasing temperature where the temperature drops slowly from the steam plateau temperature. In this zone the oil vapour coming from the previous zone condenses which leads to the increase in the oil saturation.

Zone F: In this zone, the temperature continues to decrease to reach the reservoir original temperature.

The temperature profile over the reservoir is shown in figure 3.6b at different times (each time represent the time when the temperature reached the maximum in a specific grid block). It can be seen that the peak temperature varies from one block to the other due to the fuel availability in that specific block.

Figure 3.6b also shows that the combustion front propagates at virtually constant velocity. The average velocity of the combustion front is calculated as $(0.7*50)/(t_9 - t_2)$, where t_2 and t_9 are the time in days of the temperature peak in grid block 2 and 9 respectively, and 50 is the reservoir length. The average velocity is found to be 0.38 m/day.

Figure 3.7 shows the cumulative oil and water recovery for the run expressed as a percentage from the original oil and water in place. It also shows the light and heavy oil recoveries. The light oil produced was created entirely from the cracking reaction since a light component was not present initially in any significant quantity.

Cumulative water recovery exceeds 100% of the original water in place because of the water formed in the oxidation reactions.

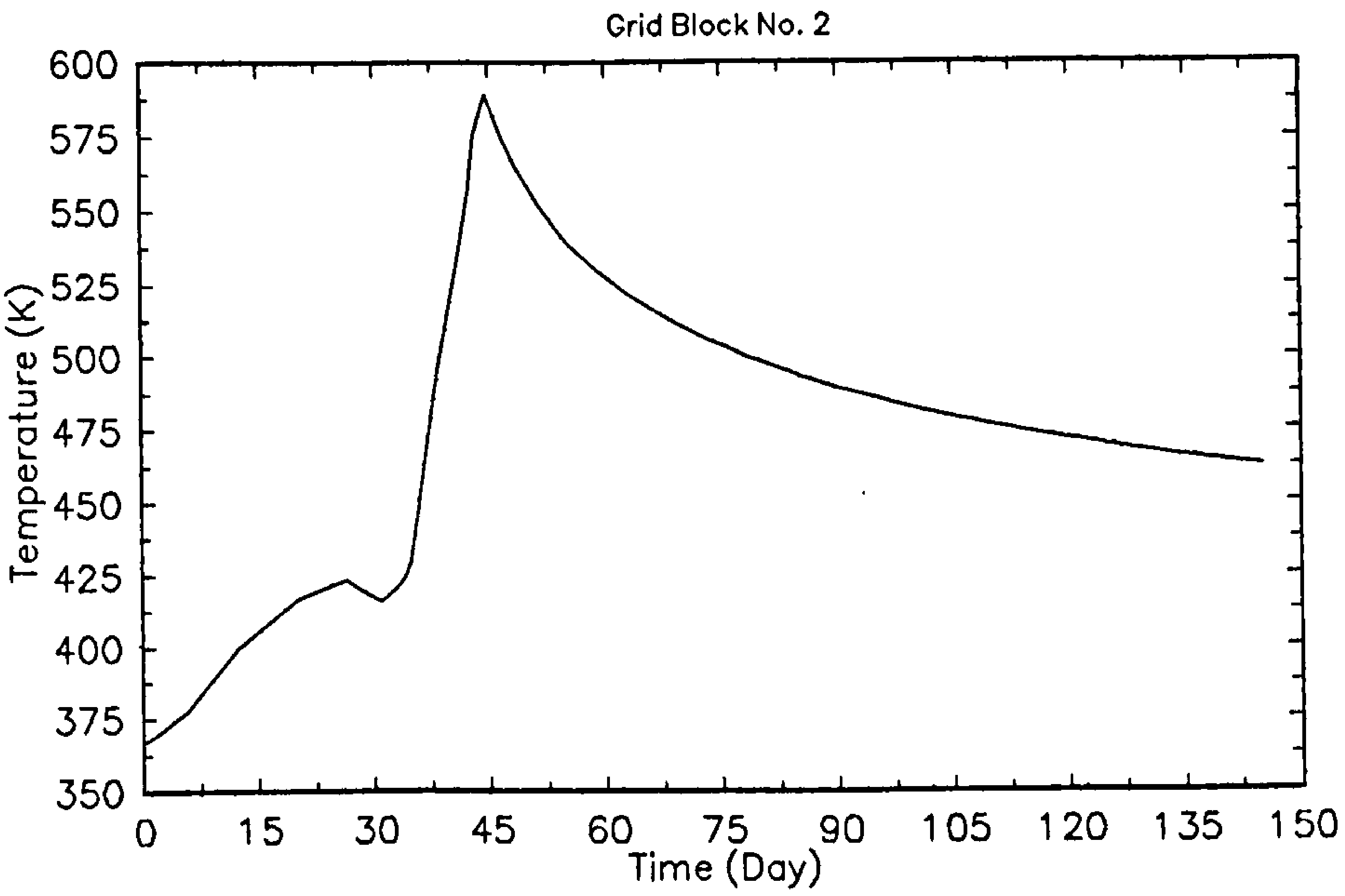


Figure 3.1 a Temperature Profile Of Grid Block 2

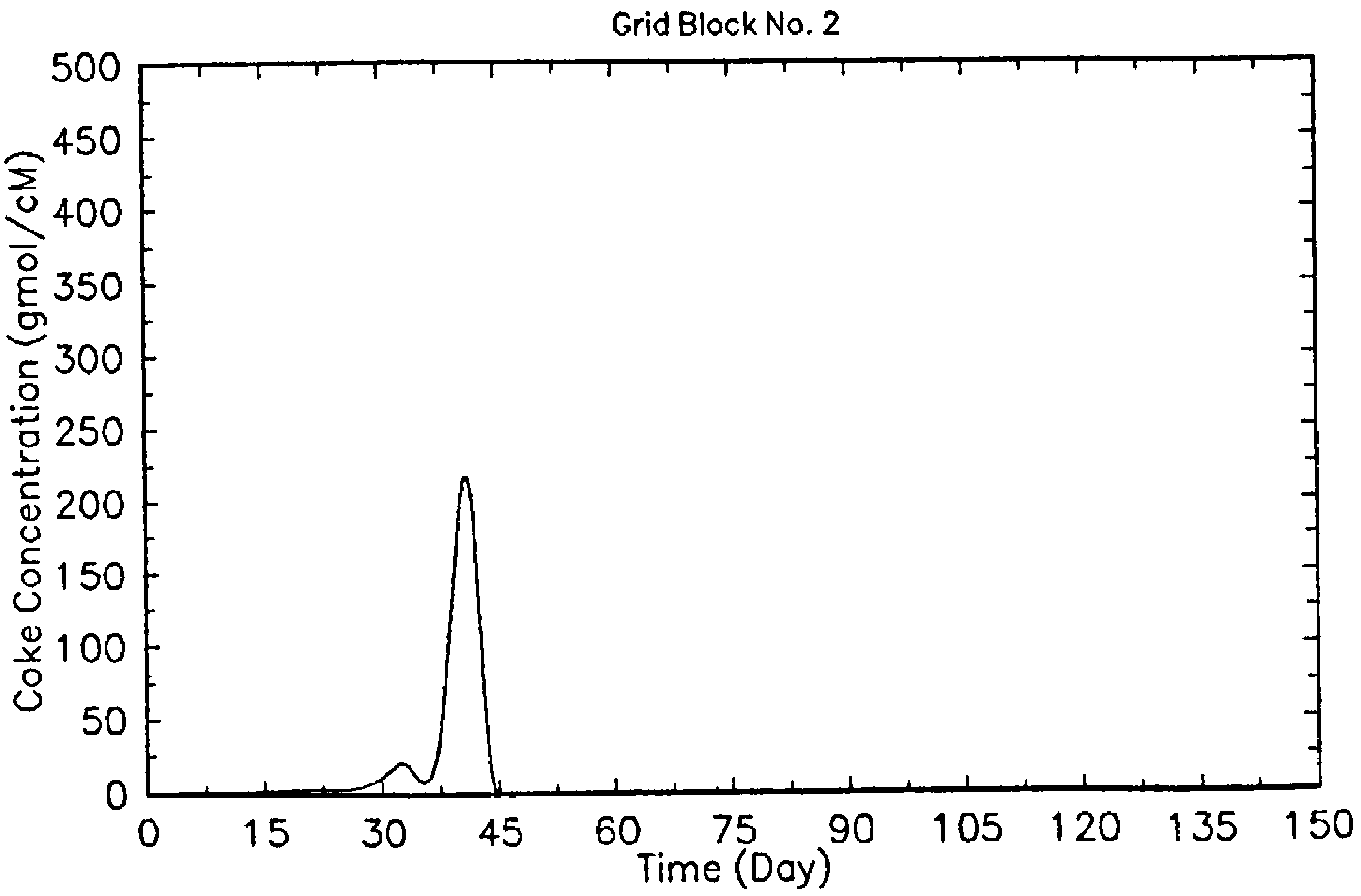


Figure 3.1 b Coke Concentration Profile Of Grid Block 2

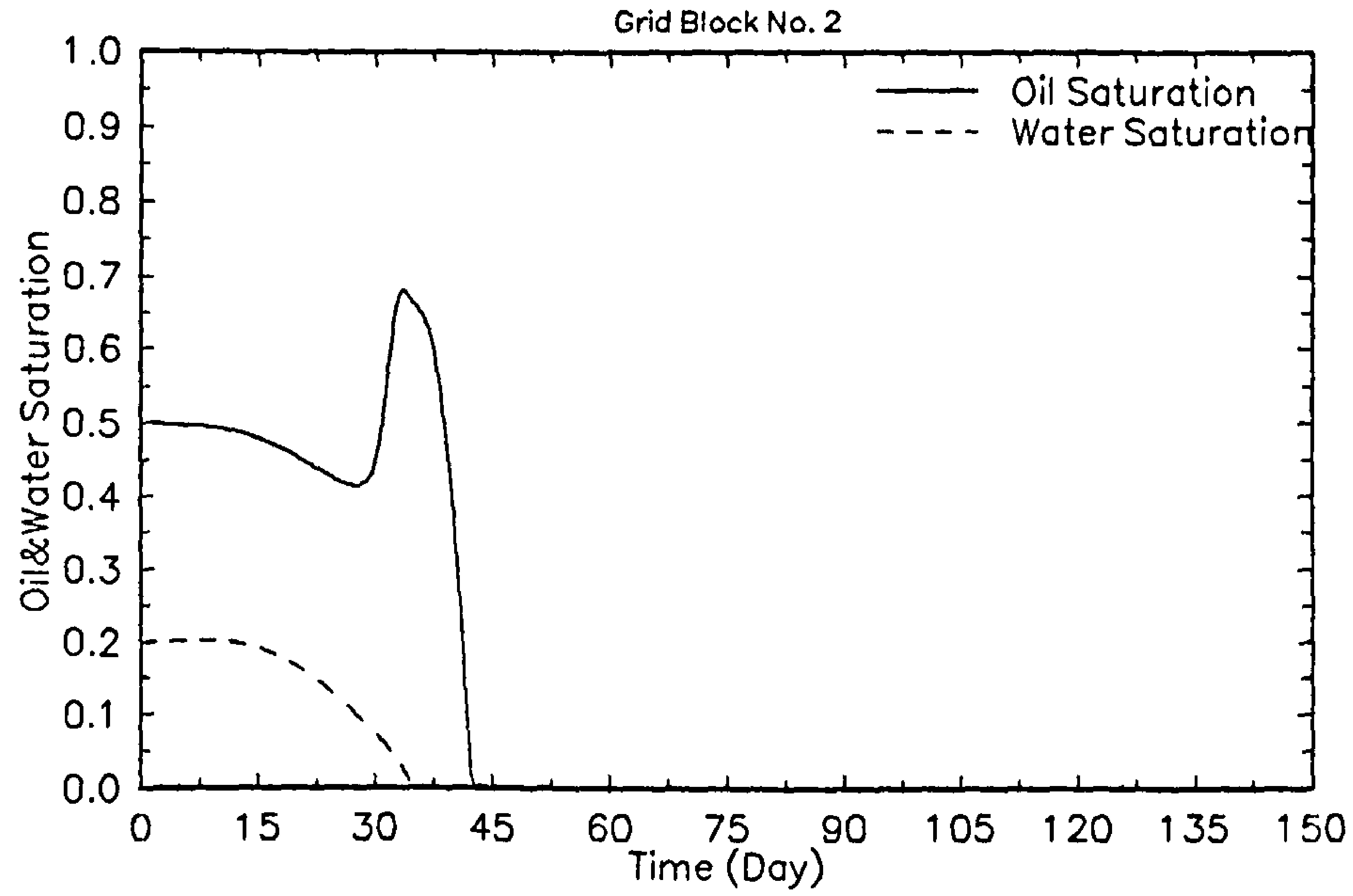


Figure 3.1 c Oil And Water Saturation Profile of Grid Block 2

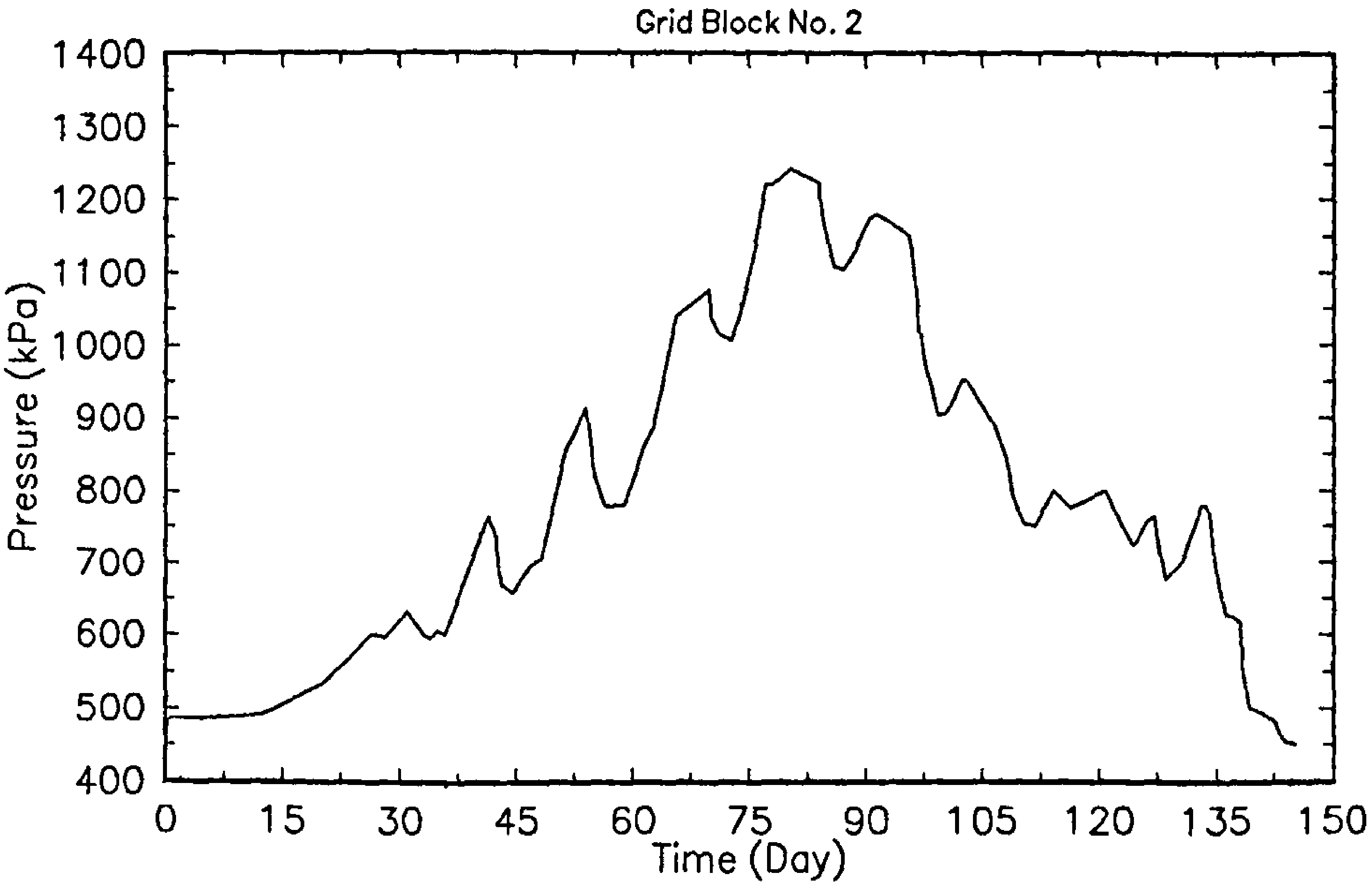


Figure 3.1 d Pressure Profile Of Grid Block 2

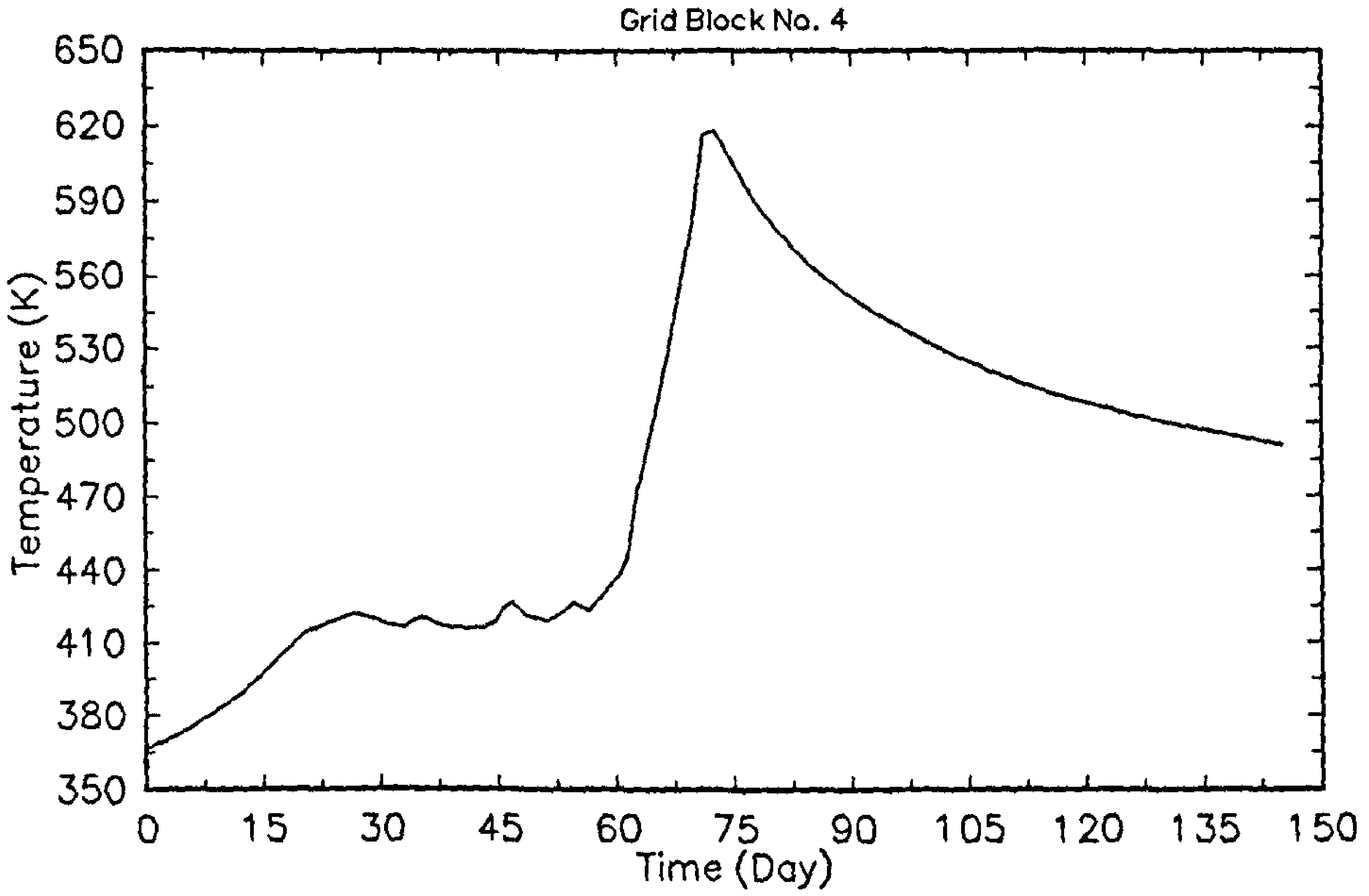


Figure 3.2a Temperature Profile Of Grid Block 4

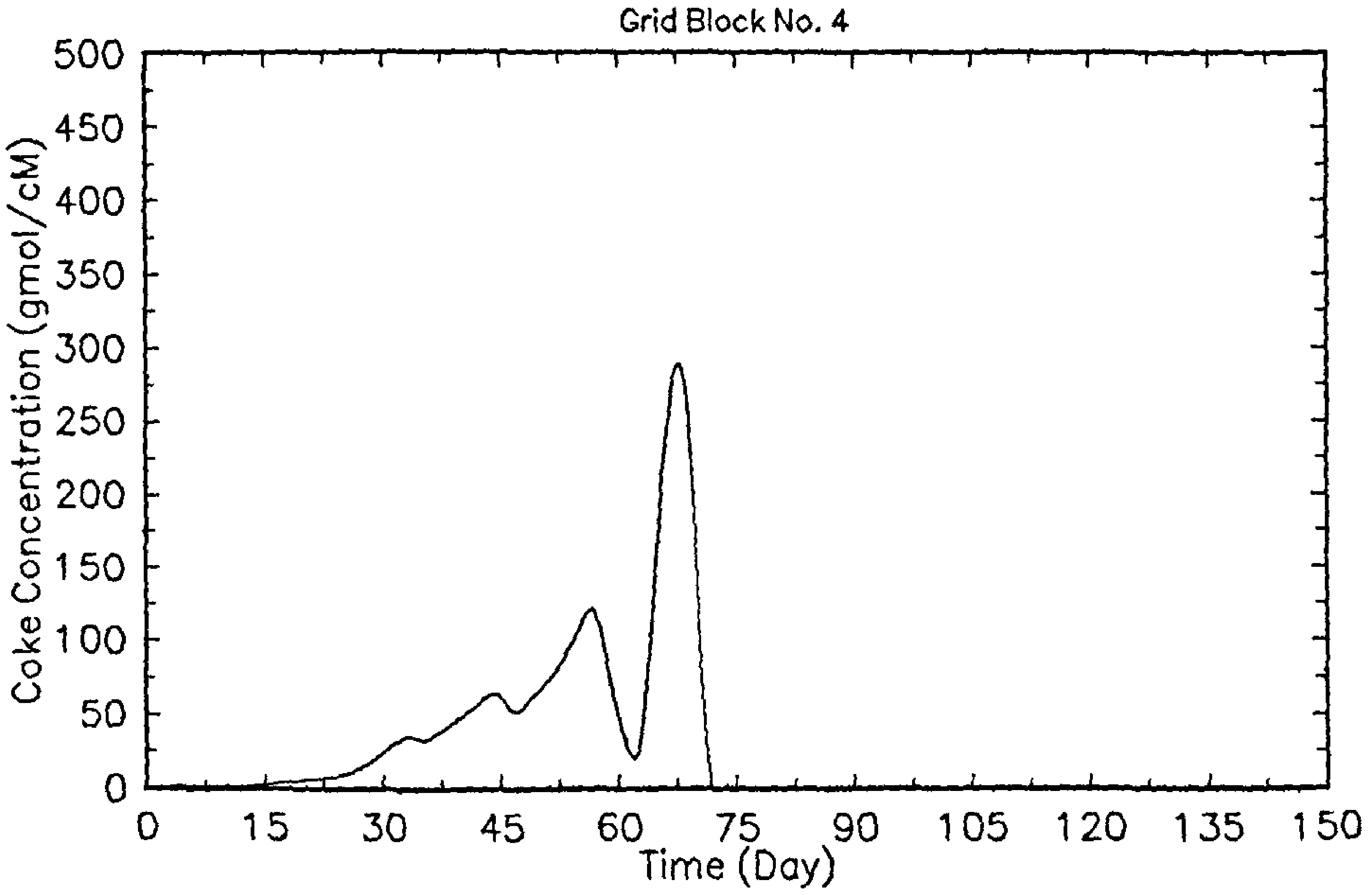


Figure 3.2b Coke Concentration Profile Of Grid Block 4

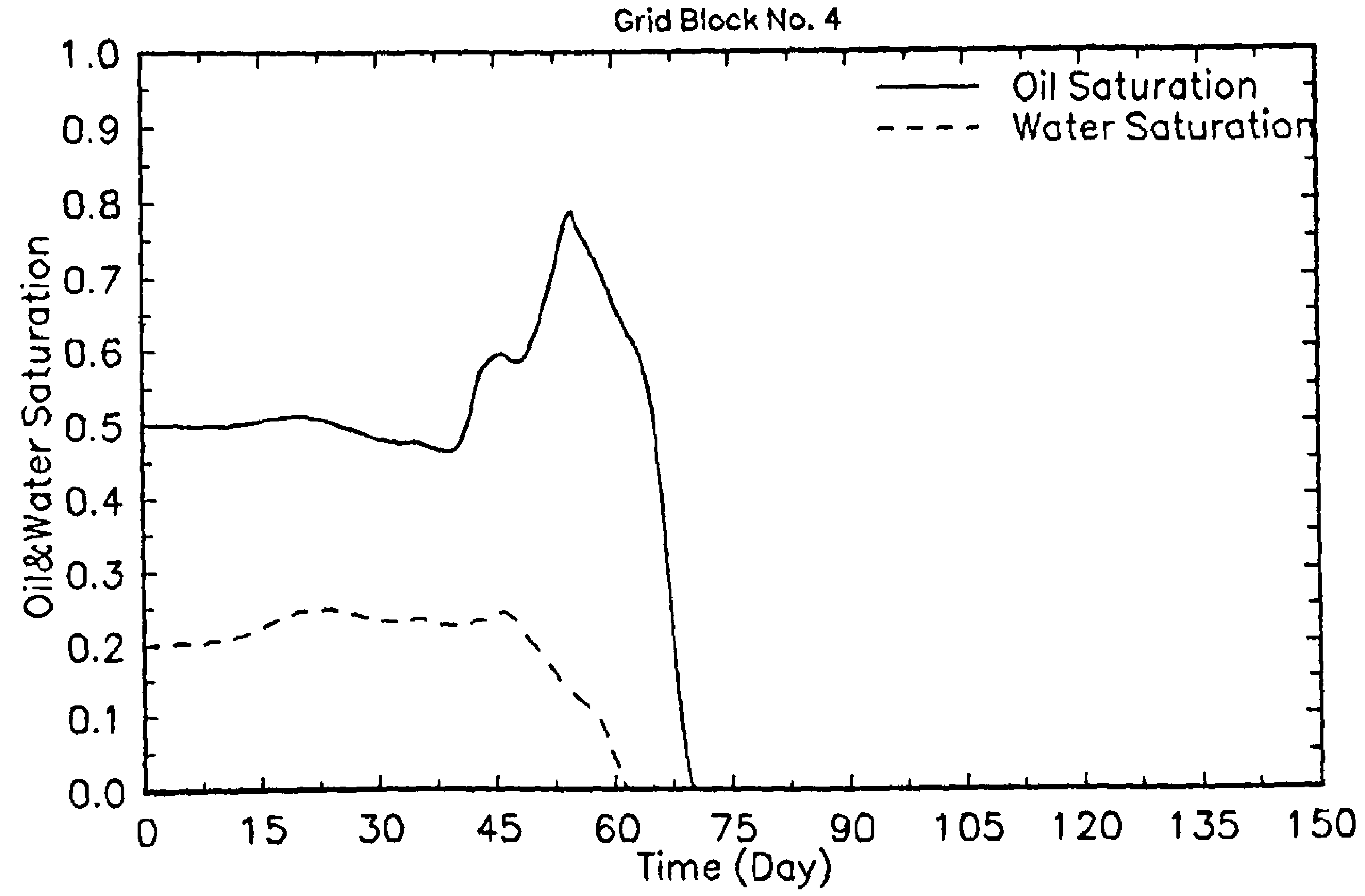


Figure 3.2c Oil And Water Saturation Profile Of Grid Block 4

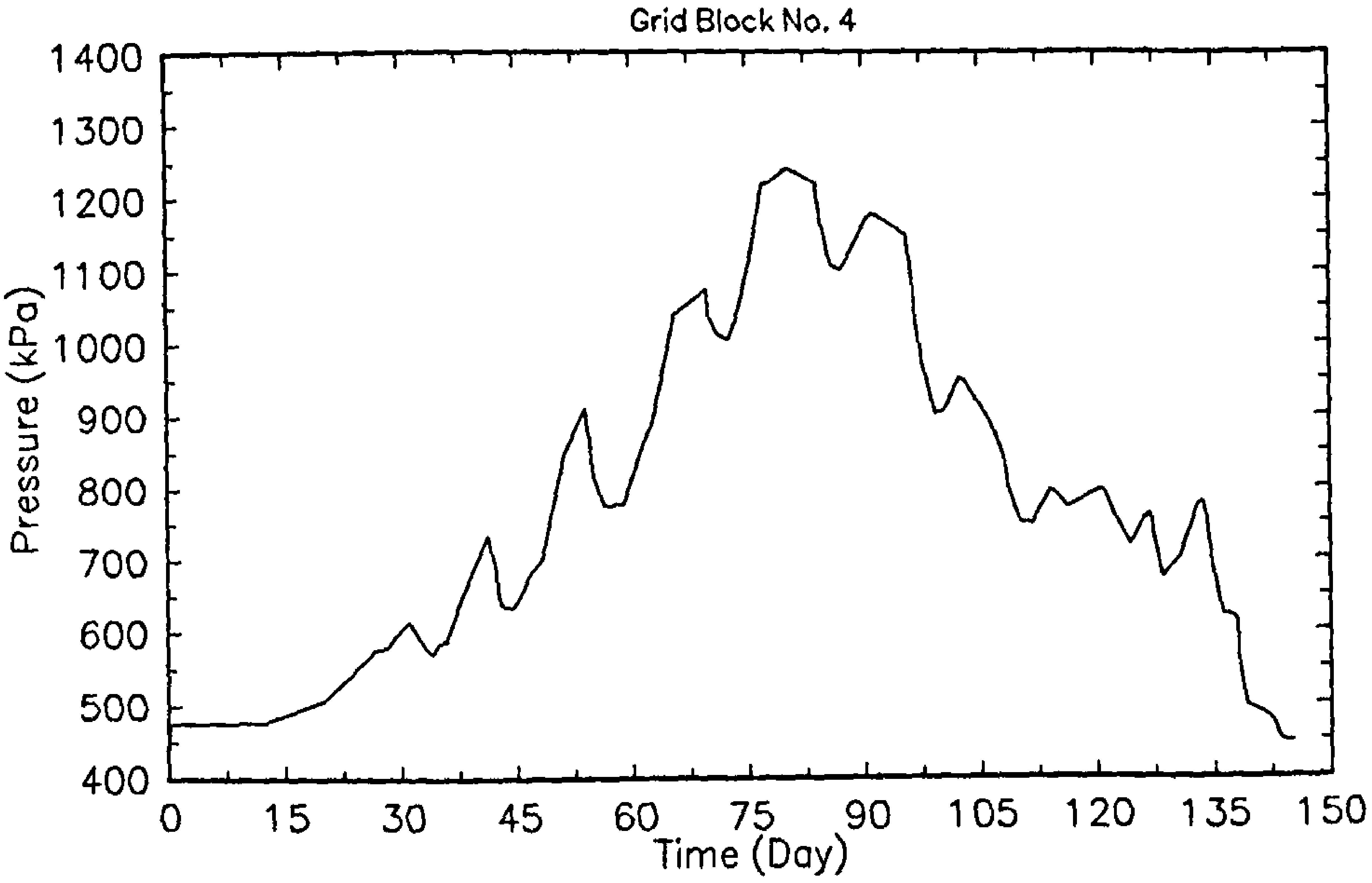


Figure 3.2d Pressure Profile Of Grid Block 4

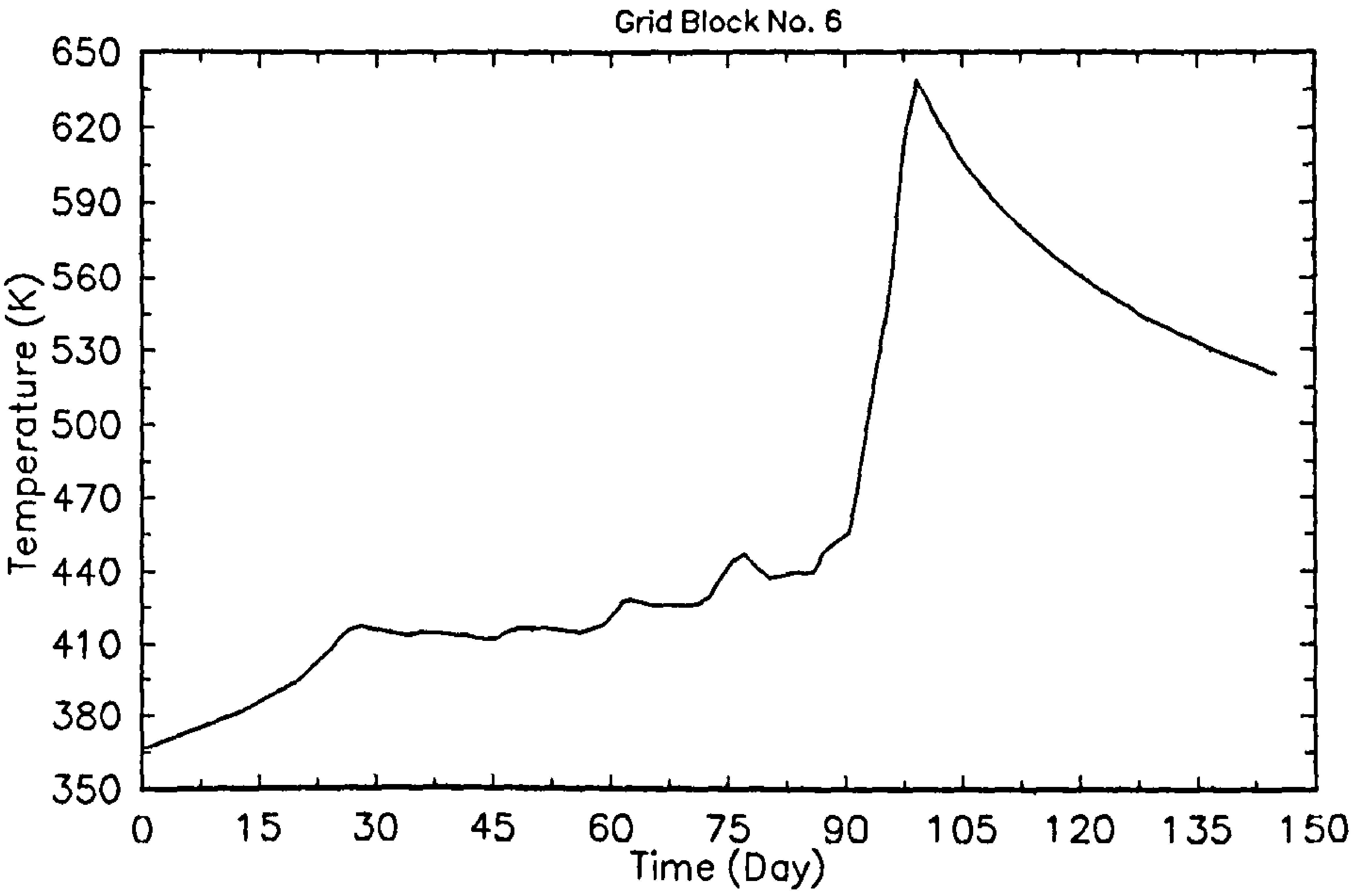


Figure 3.3a Temperature Profile Of Grid Block 6

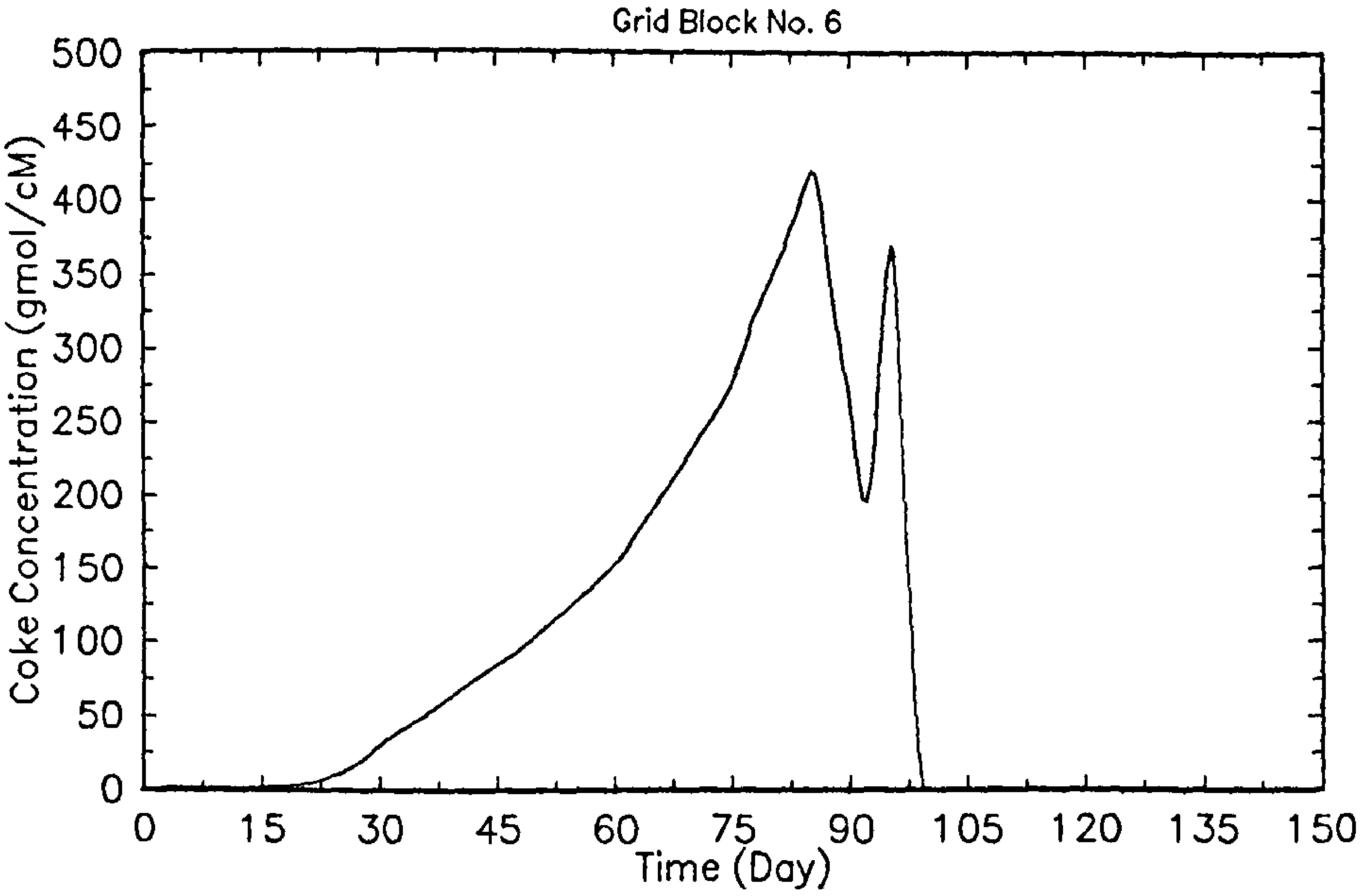


Figure 3.3b Coke Concentration Profile Of Grid Block 6

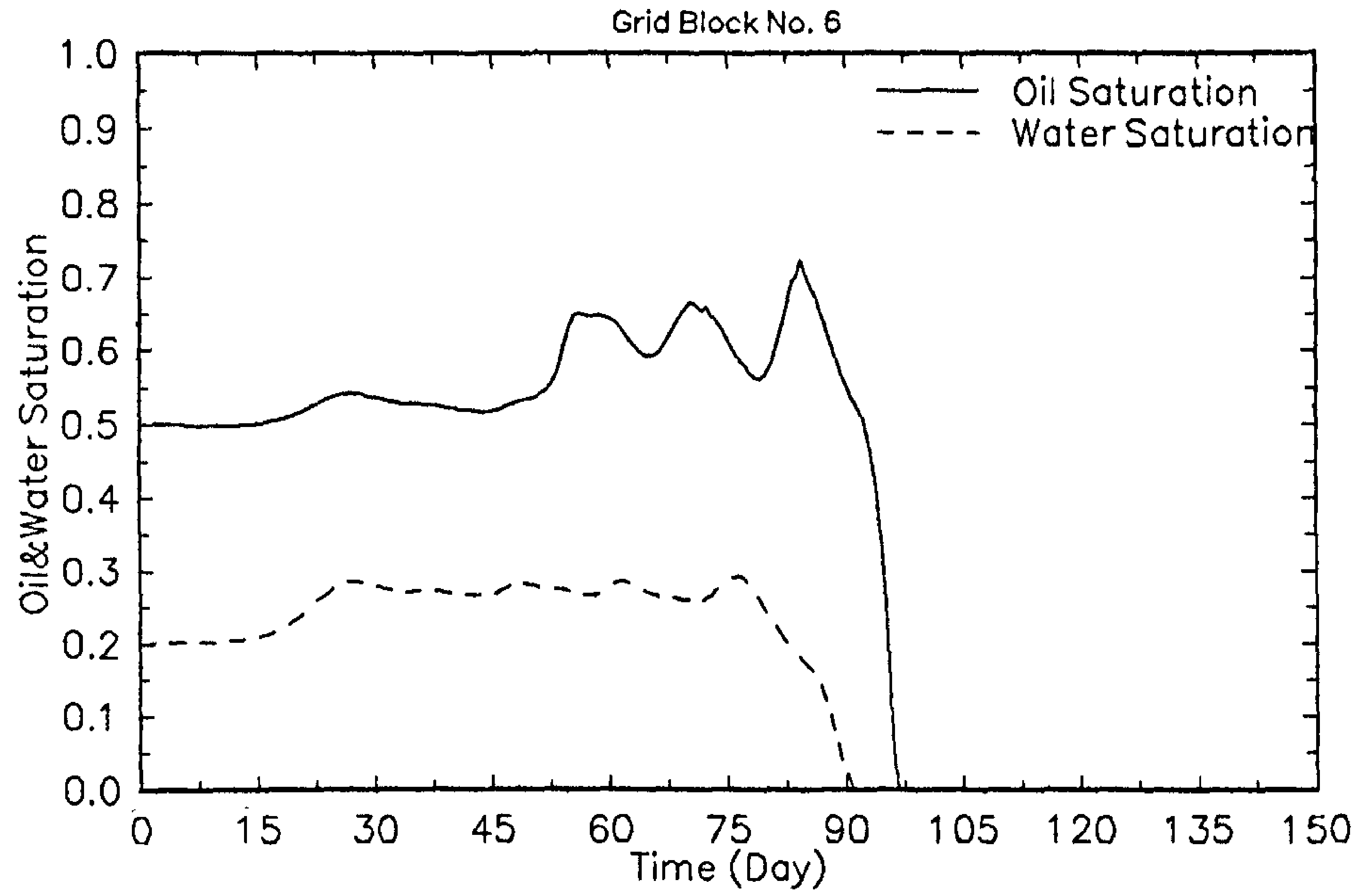


Figure 3.3c Oil And Water Saturation Profile Of Grid Block 6

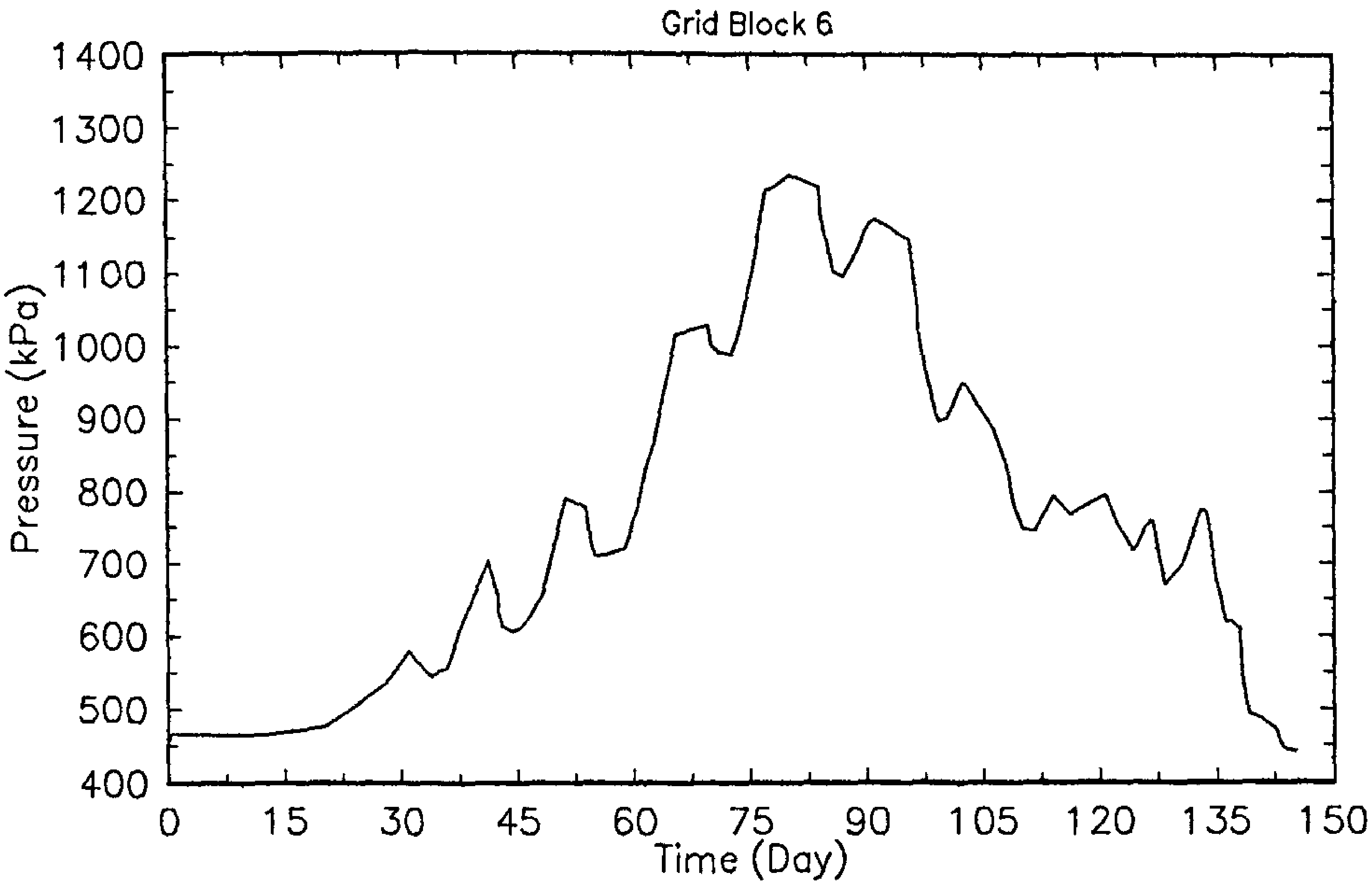


Figure 3.3d Pressure Profile Of Grid Block 6

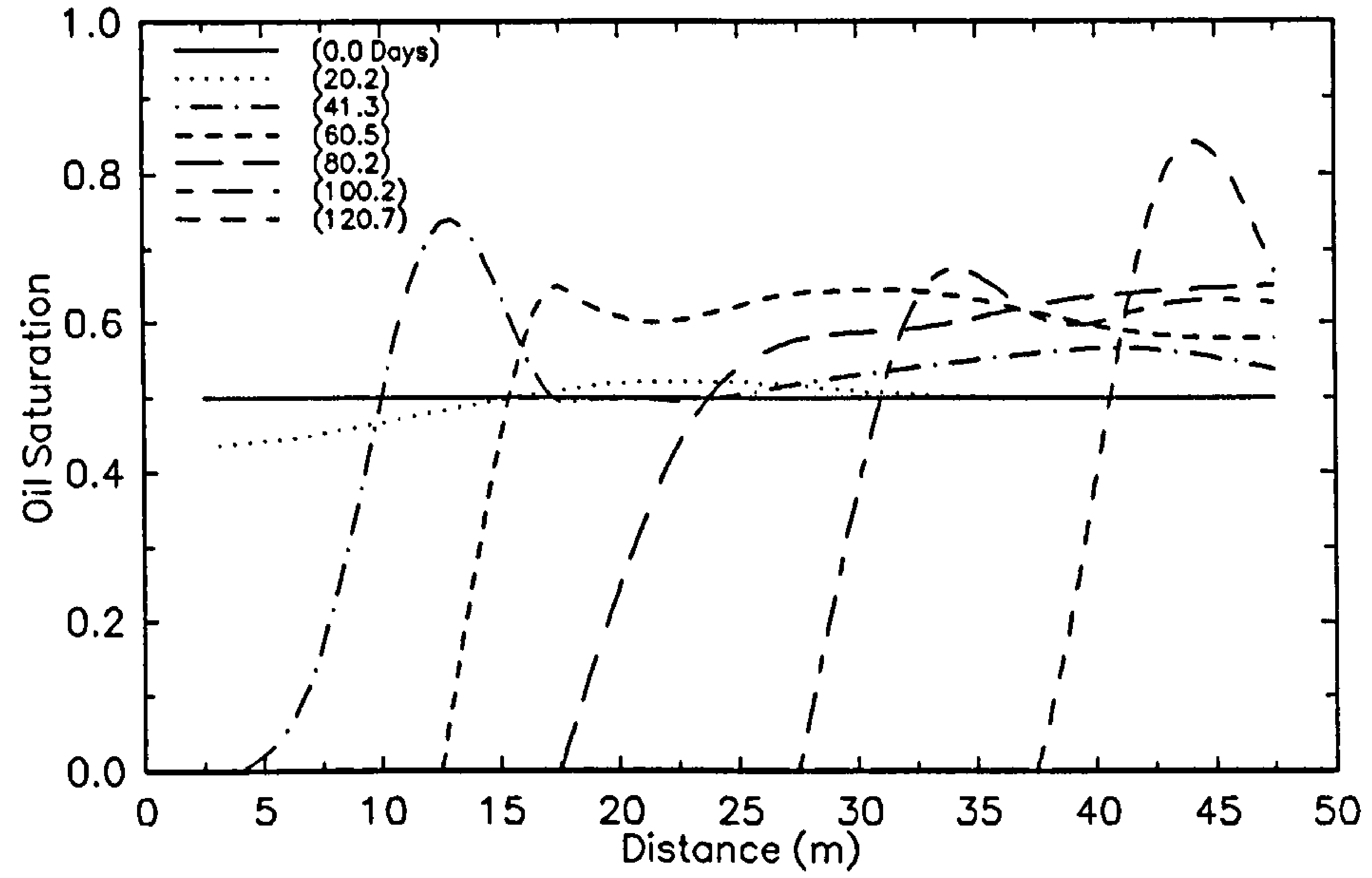


Figure 3.4 Oil Saturation Profile Over The Reservoir At Various Times

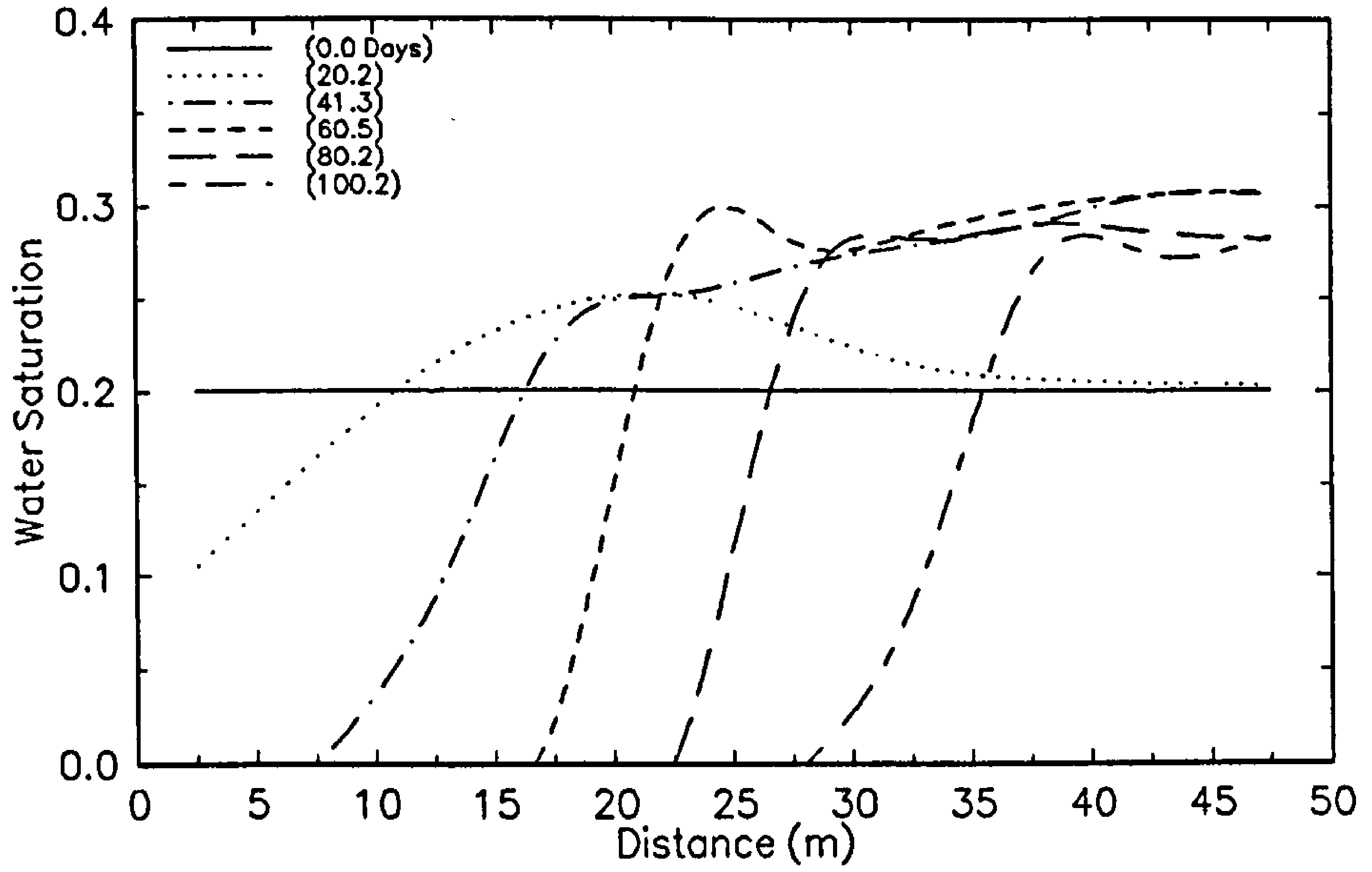


Figure 3.5 Water Saturation Profile Over The Reservoir At Various Times

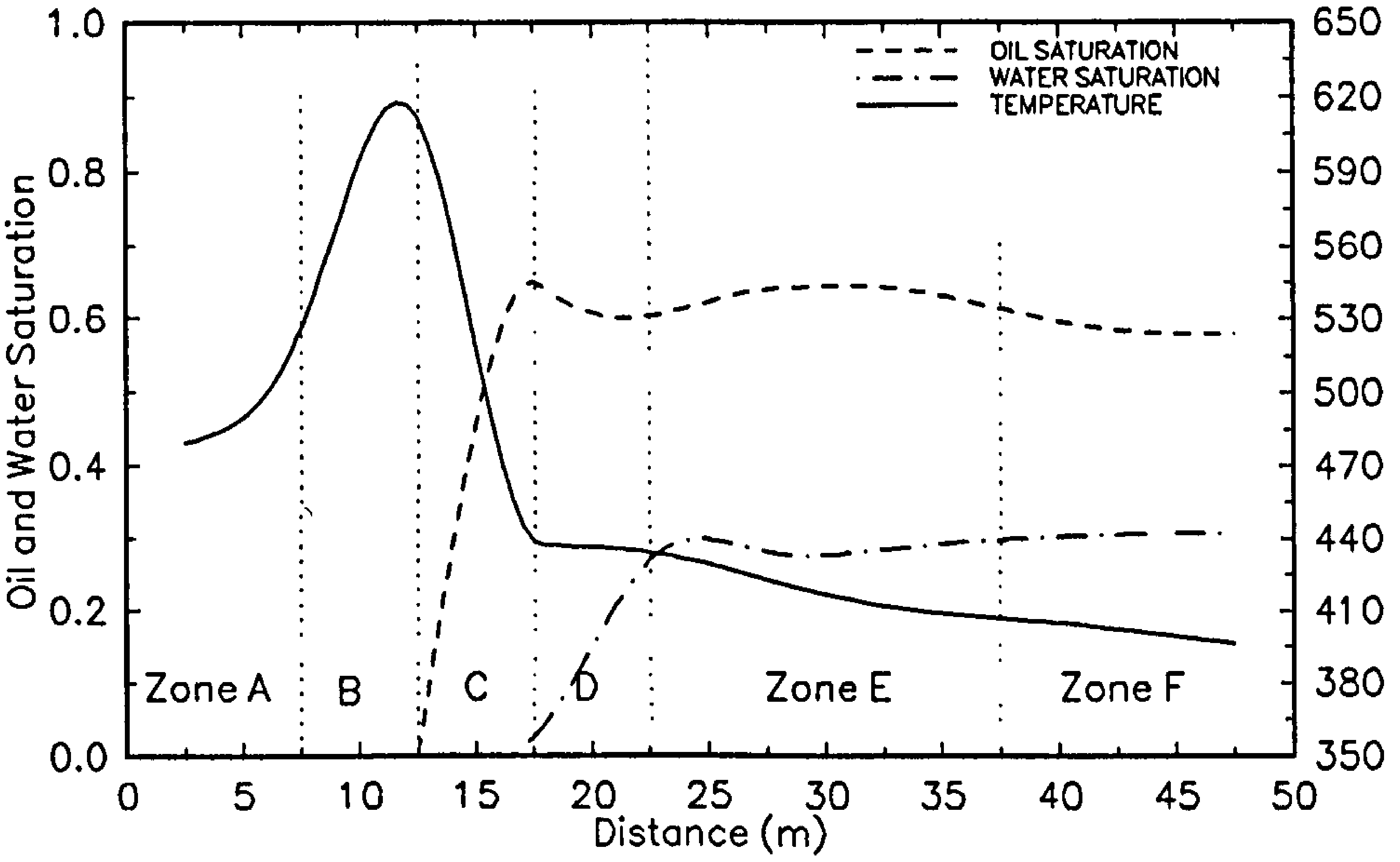


Figure 3.6a Temperature and Saturation Profiles Over The Reservoir After (60.5 day)

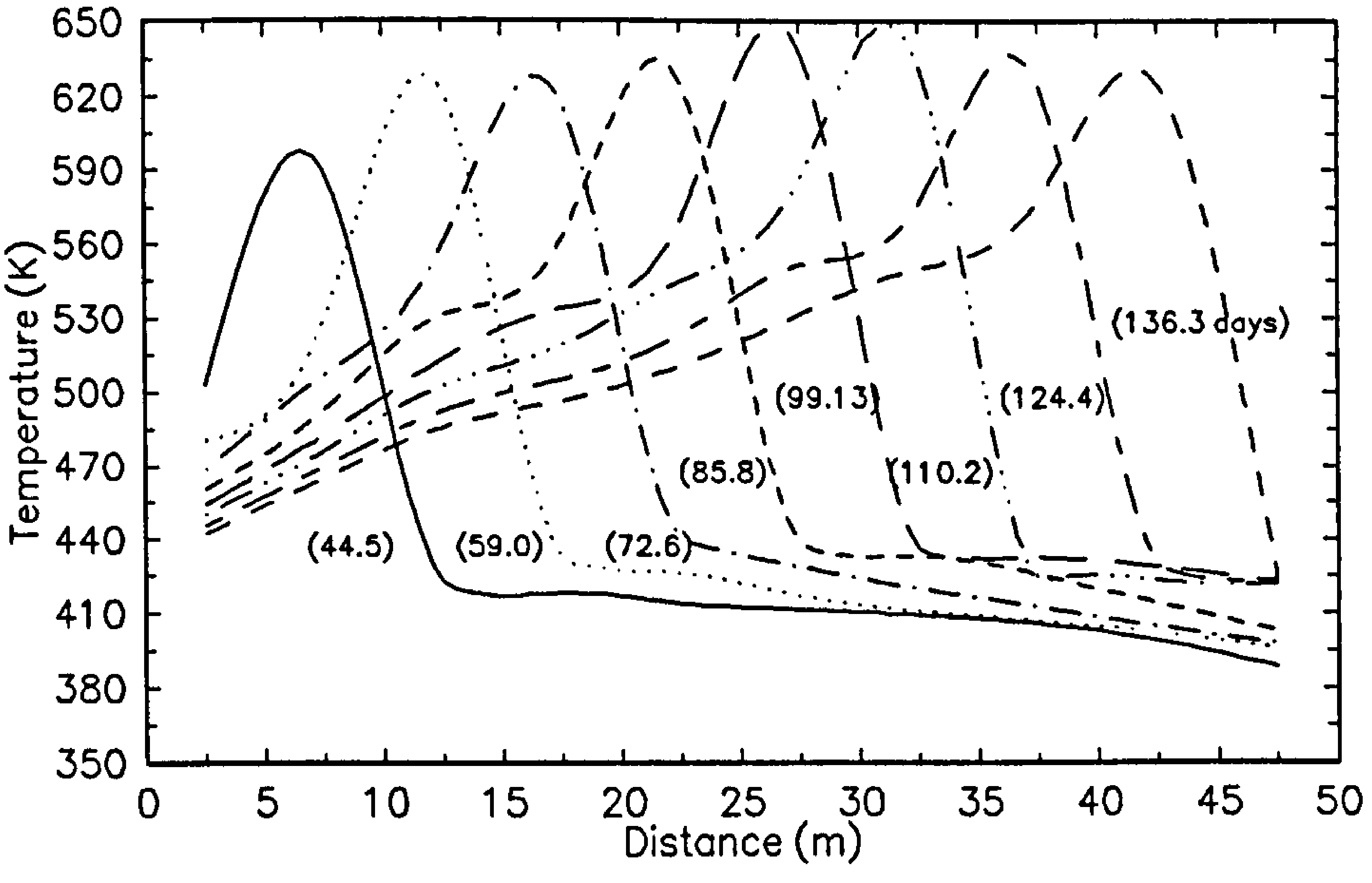


Figure 3.6b Temperature profile Over The Reservoir

(Brackets represent the time when the temperature reached maximum at the specific grid block)

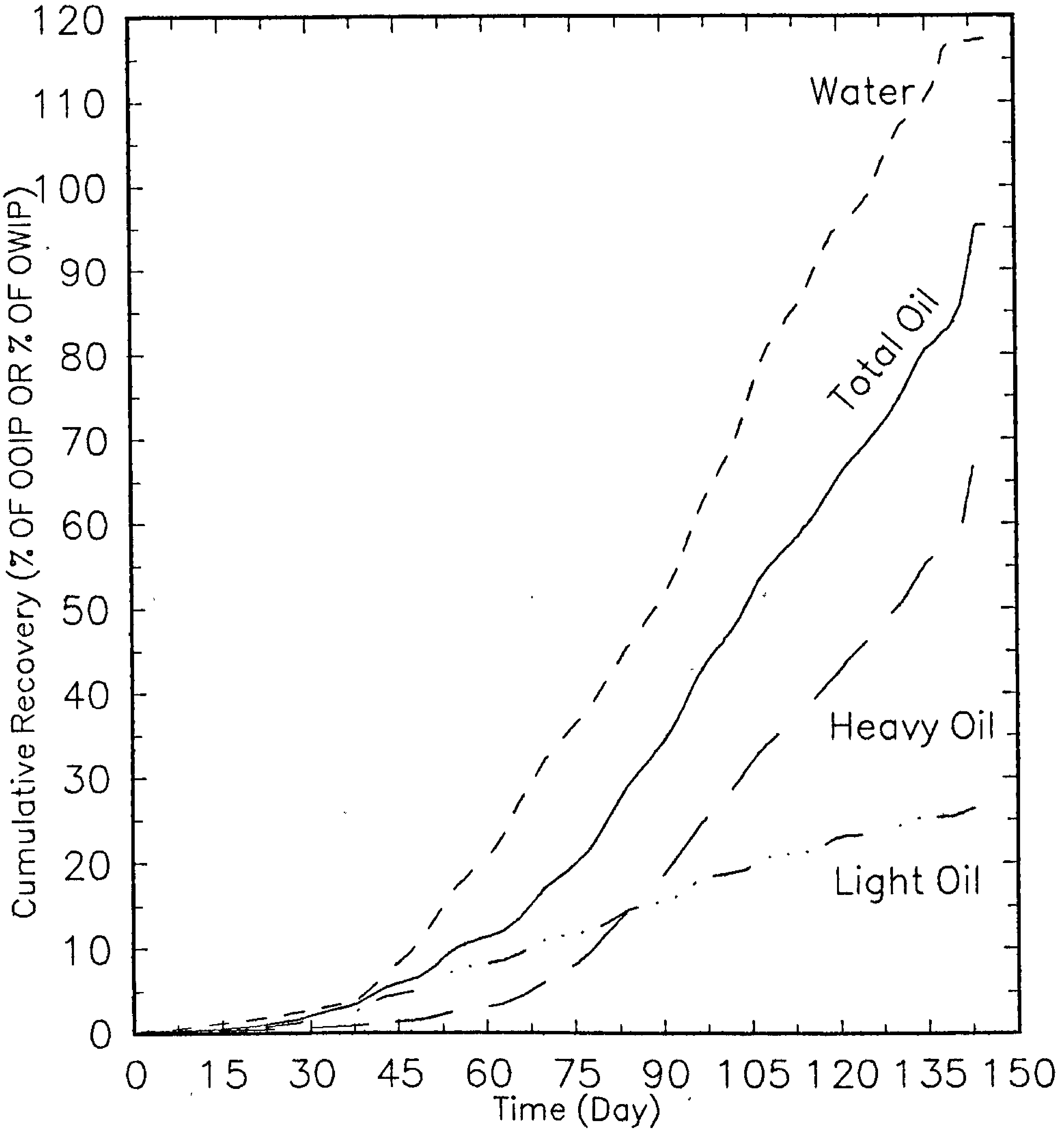


Figure 3.7 Cumulative Oil And Water Recovery

3.3 Validation With ISCOM Thermal Simulator

Before a numerical simulator can be used to design a field project, it is important to establish confidence in its predictive capability for the particular field. One way of achieving this is to compare the results with results of similar calculations taken from a more general and well tested model.

The **ISCOM** simulator (Grabowski, Vinsome, Lin, Behie, and Rubin [1979] and Rubin, and Buchannan [1983]) is used to validate this model. **ISCOM** is one of a number of multi-dimensional in-situ combustion simulators that have been developed in recent years (others are those of Crookston, Culham, and Chen [1979], Youngren [1980], and Coats [1980]). It is a compositional simulator in that it solves a molar conservation equation, including the contribution from chemical reactions, for each component. The conservation equations are solved in conjunction with a single energy equation which includes the effects of conduction, convection, heats of reaction, heat injection and production, and losses to surrounding rocks. Phase equilibrium is described by empirical physical correlations. Also each chemical reaction is described by a stoichiometric equation, and each reaction rate is described by an empirical equation. Other physical properties are also described by correlations or tabulations.

The main differences between the ISCOM simulator and this project simulator which is designated as (SUISCOM) can be summarized as follows:

1. ISCOM can accept any number of components whereas the SUISCOM can

only accept the six components described in chapter two.

2. Any number and form of reactions can be used in ISCOM whereas only the four reactions described in chapter 2 can be used in SUISCOM.

3. ISCOM uses a modified form of Gaussian elimination to solve the Jacobian matrix whereas SUISCOM uses the LU decomposition method to solve the matrix.

The bench mark simulation data have been used to perform the comparison between the two simulators. Figures 3.8a-c, and 3.9a-c show the temperature, oil saturation, and water saturation profiles over the reservoir after 21.00 (before any ignition to take place) and 60.5 days respectively. Figures 3.10a-c present the water, oil, and light oil recovery profiles for the two simulators. These figures illustrate good agreement between the simulators and the discrepancies shown in the recovery profiles are due to the fact that a slight increase in the coke lay down occurs in the case of the ISCOM simulator. This increase which results from ignoring the term $[1.0 - (C_c/C_{max})^5]$ in the coke production reaction rate equation in the ISCOM program which leads to a faster combustion front because of the extra heat generated from burning that coke.

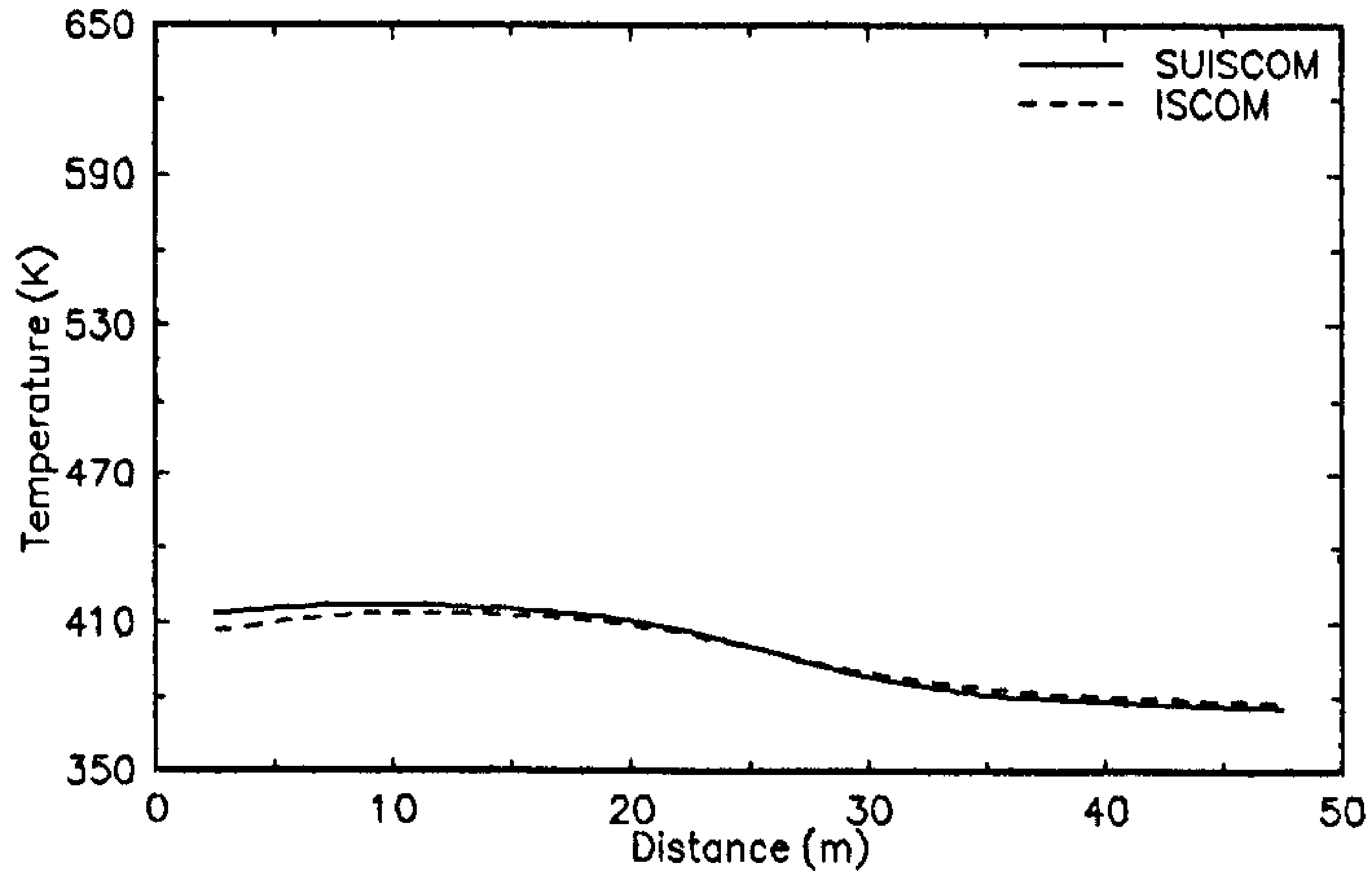


Figure 3.8a Temperature Profile Over The Reservoir (After 21.0 Days)

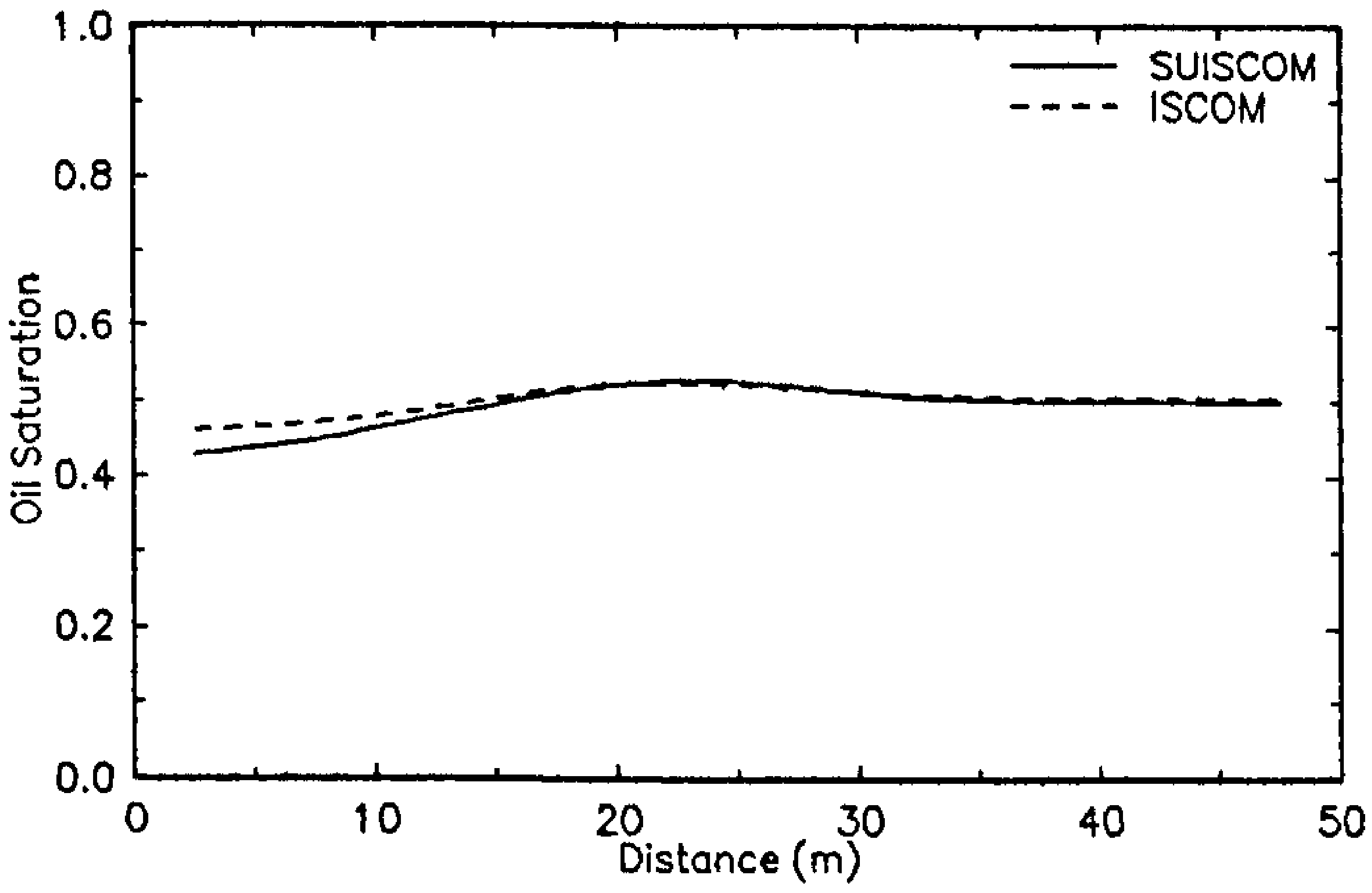


Figure 3.8b Oil Saturation Profile Over The Reservoir (After 21.0 Days)

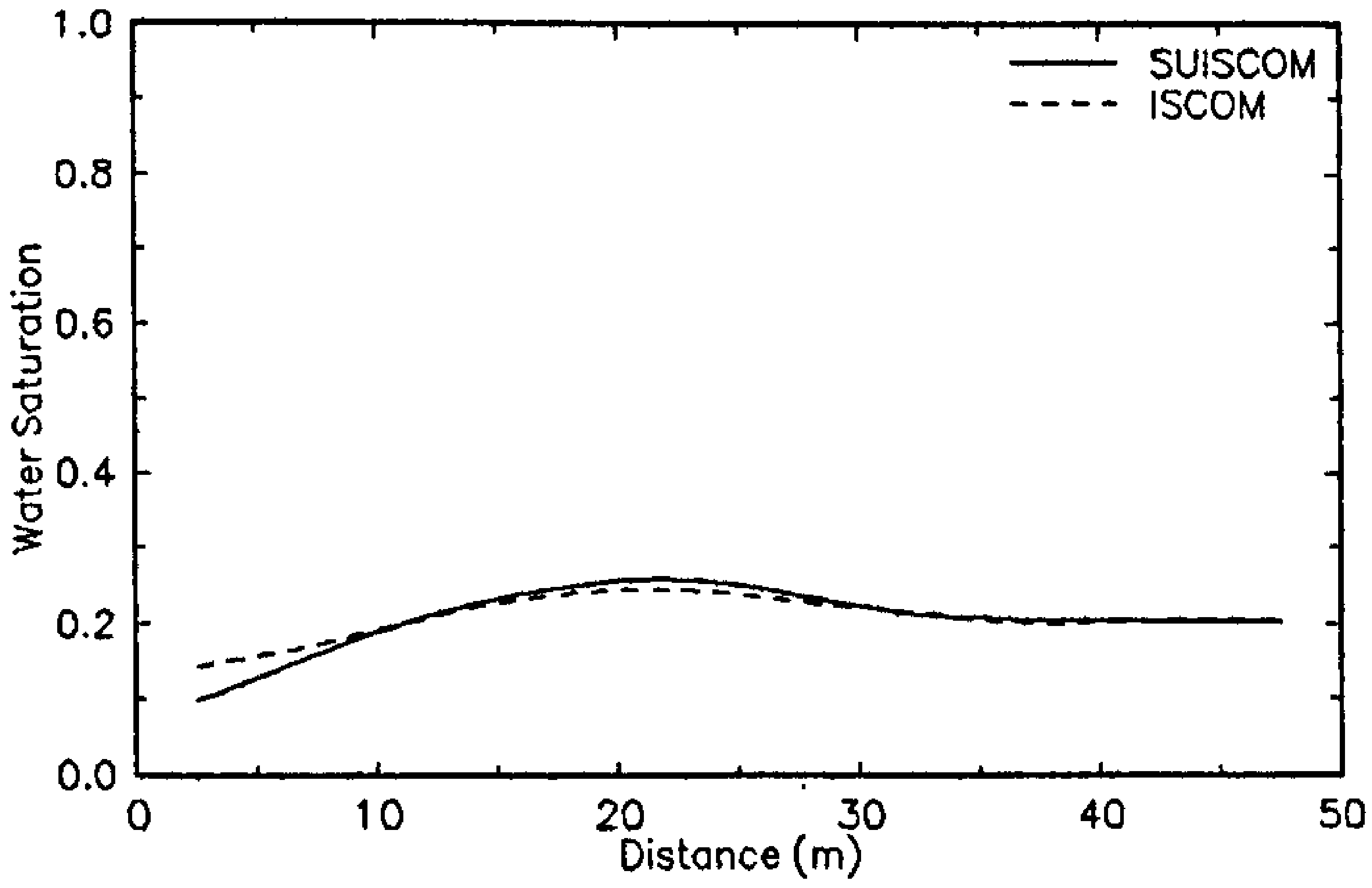


Figure 3.8c Water Saturation Profile Over The Reservoir (After 21.0 Days)

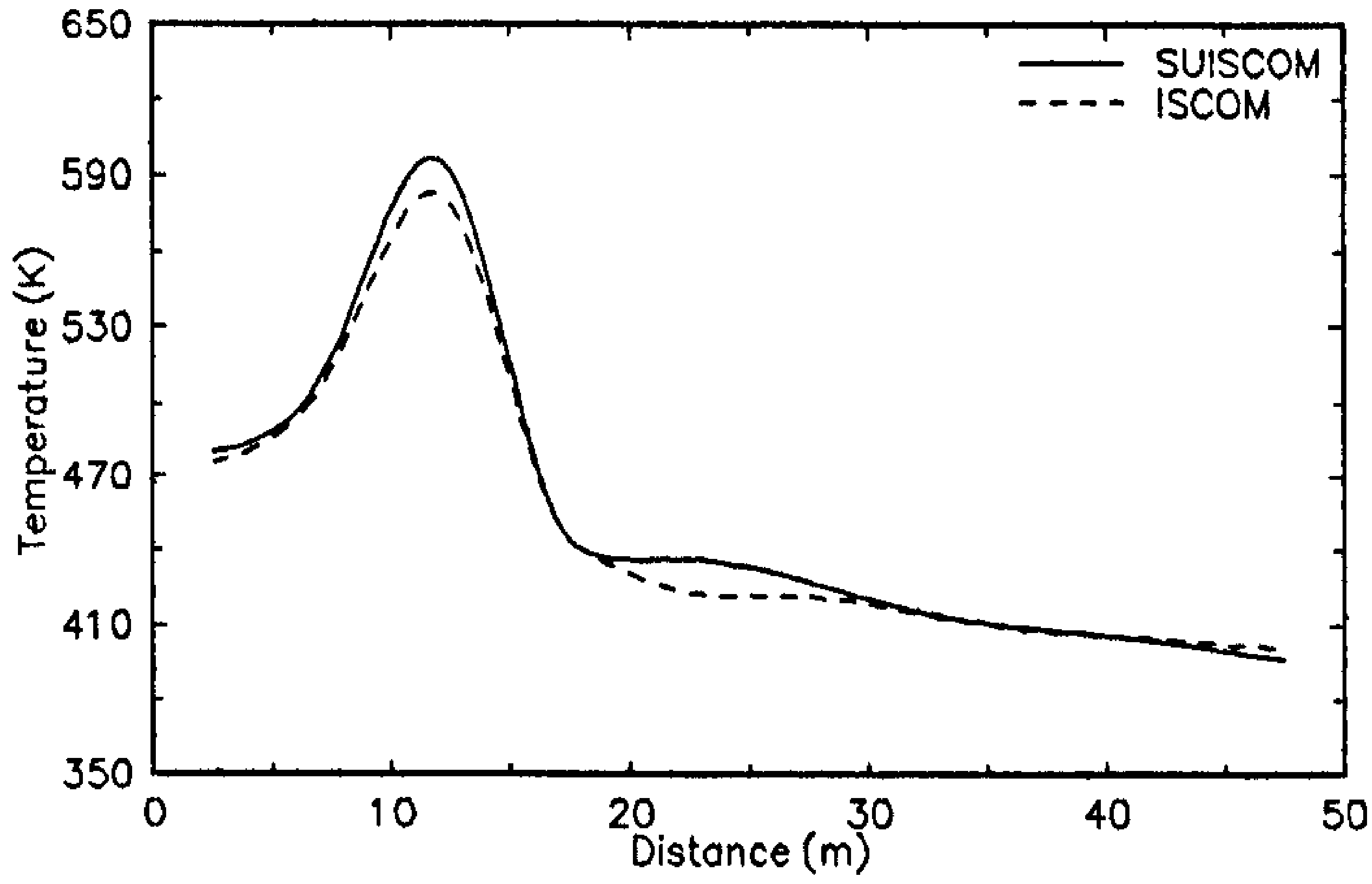


Figure 3.9a Temperature Profile Over The Reservoir (After 60.5 Days)

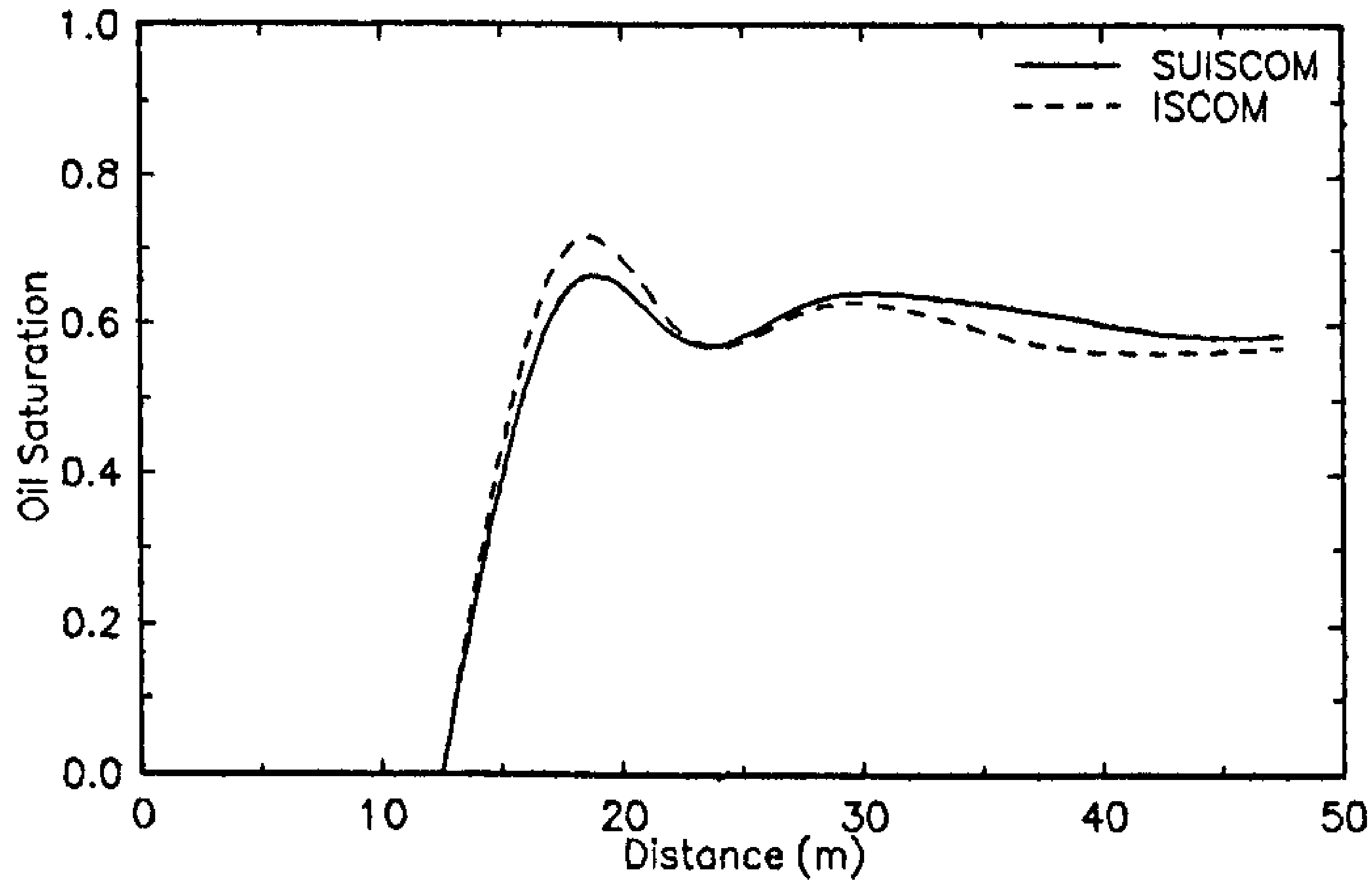


Figure 3.9b Oil Saturation Profile Over The Reservoir (After 60.5 Days)

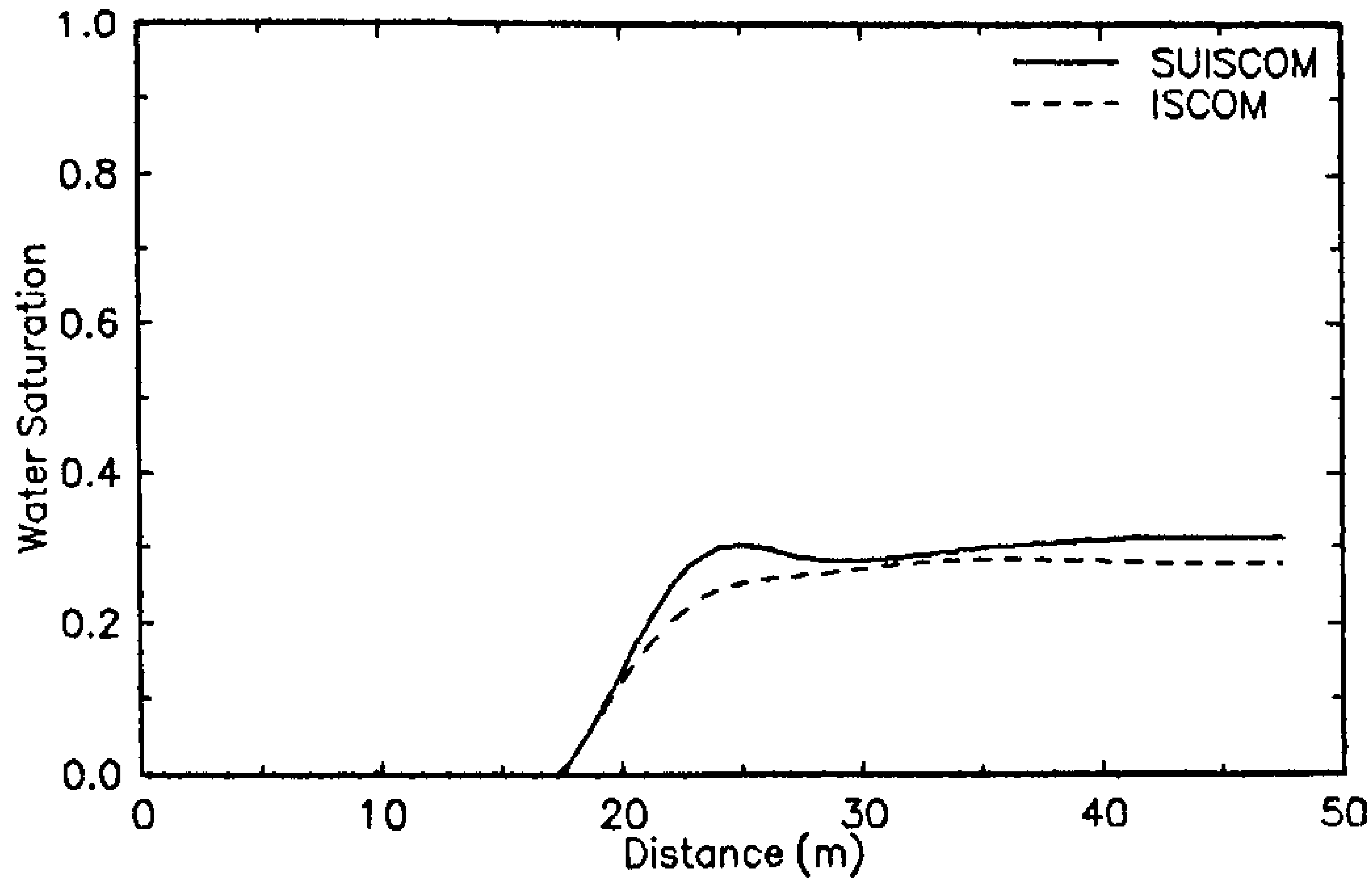


Figure 3.9c Water Saturation Profile Over The Reservoir (After 60.5 Days)

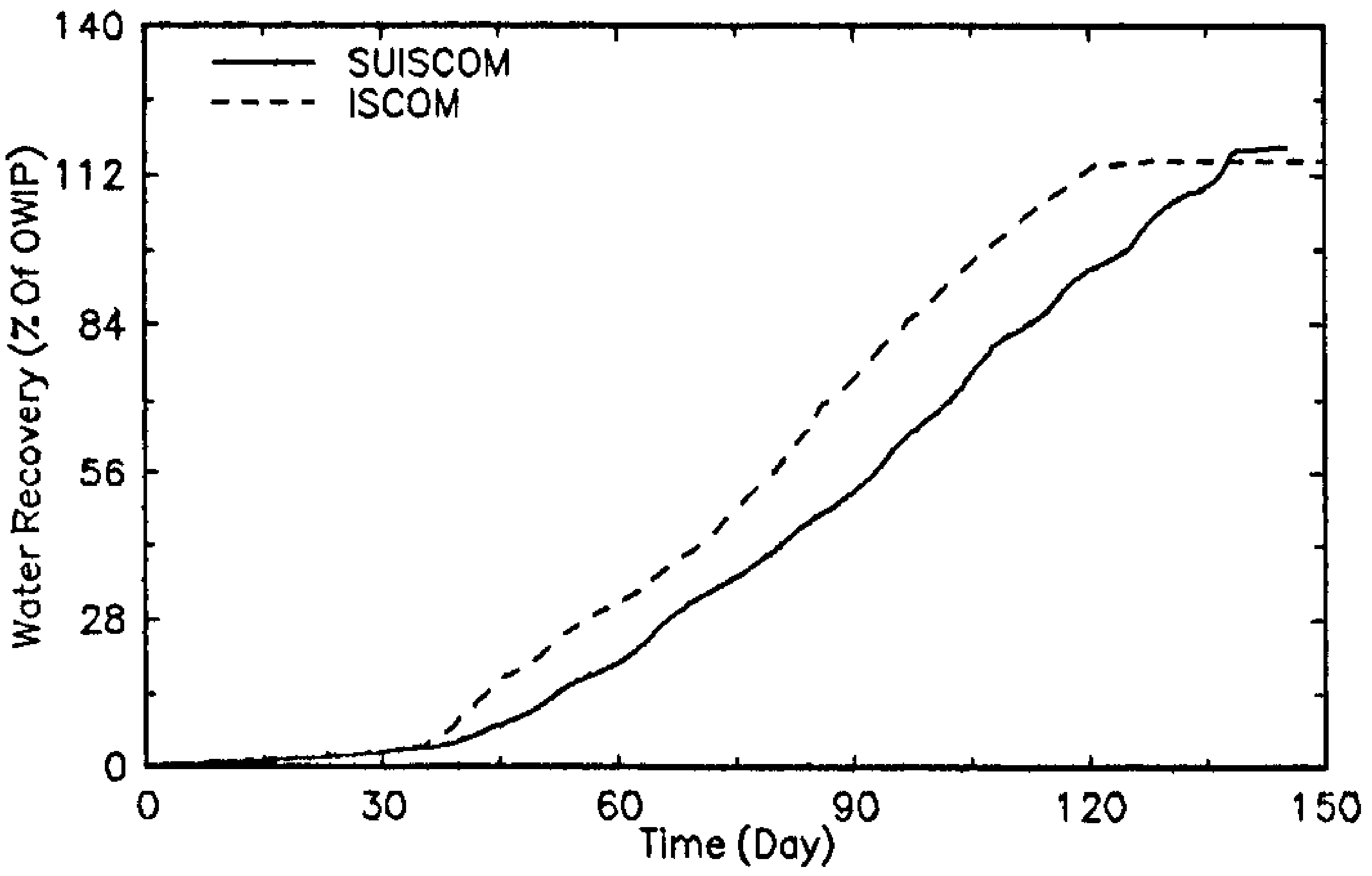


Figure 3.10a Cumulative Water Recovery Profile

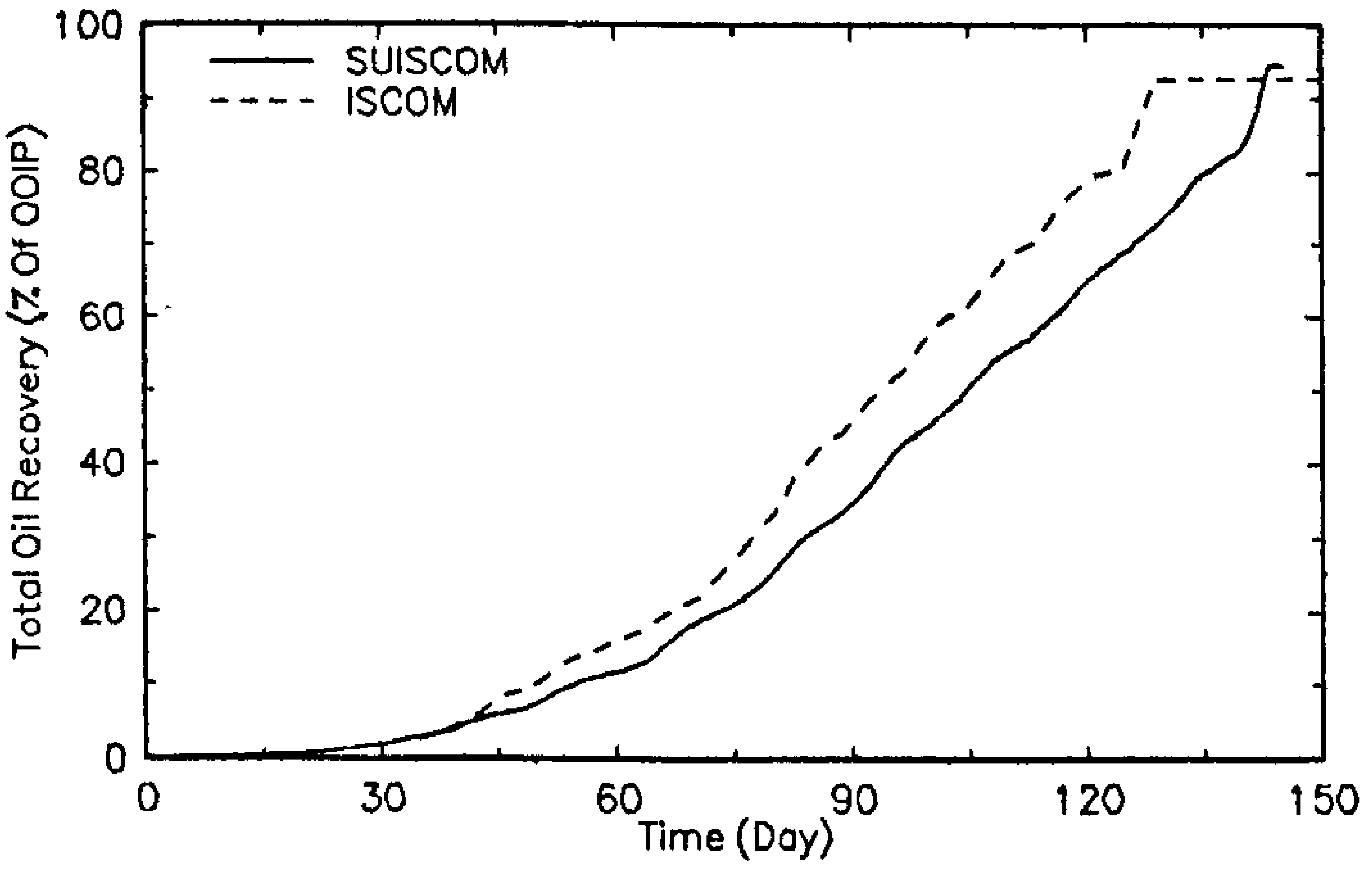


Figure 3.10b Cumulative Oil Recovery Profile

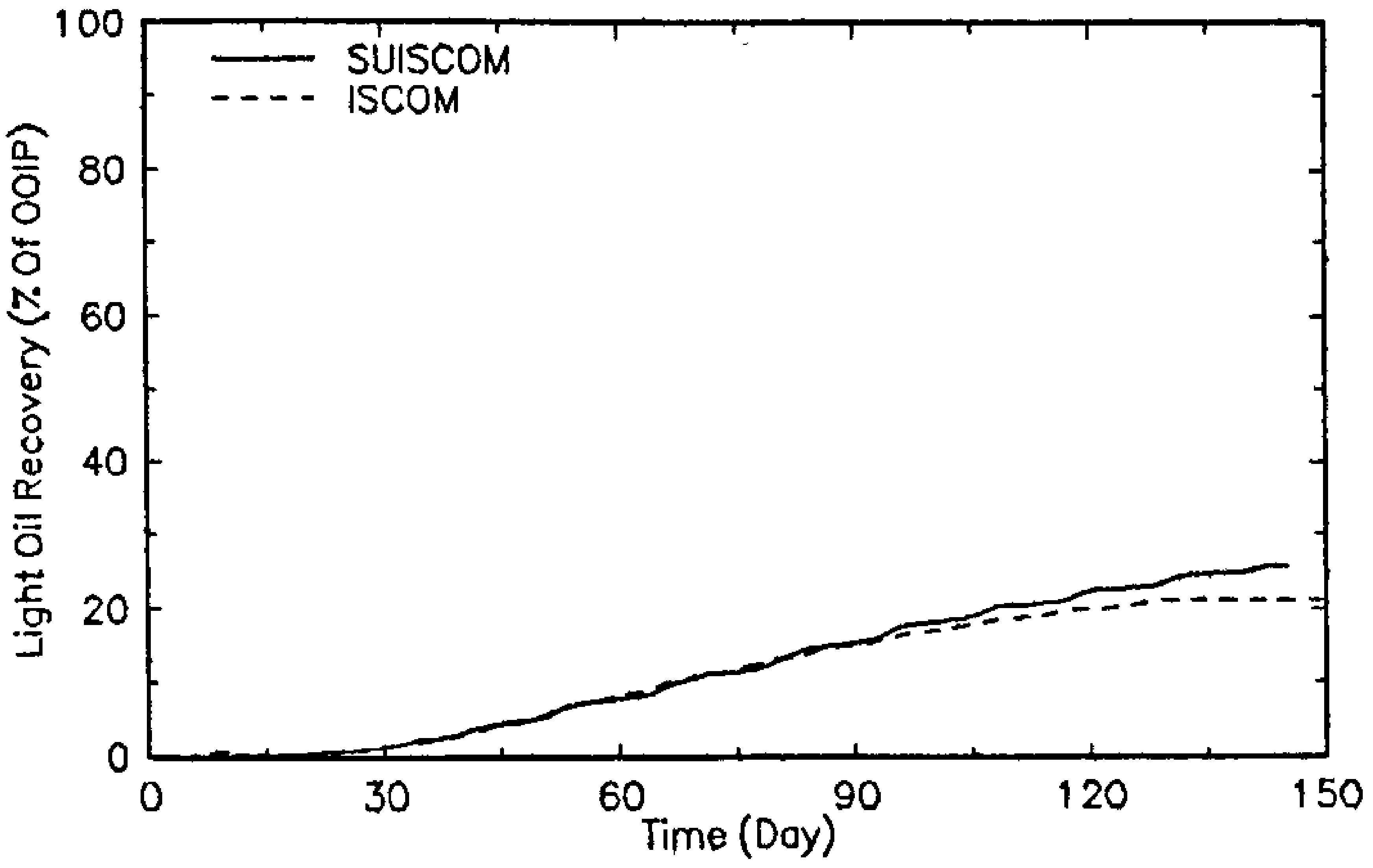


Figure 3.10c Cumulative Light Oil Recovery Profile

3.4. The Evaluation Of The Inter-Block Mobility Term and Optimization for Number of Grid Blocks

Single-point upstream weighting is used to evaluate the mobility terms in the finite difference equations. This scheme leads to numerically stable finite difference schemes if they are evaluated implicitly. This stabilization is a result of moving an off-diagonal term in the Jacobian matrix (the matrix used to solve the problem) onto the diagonal to give more diagonal dominance and a guaranteed solution. However, the local truncation error associated with single-point upstream weighting is first order and the first order terms introduces an artificial diffusion that is known as numerical dispersion. The most noticeable result of numerical dispersion is that sharp fronts in fluid displacement are not modeled accurately but are smoothed out considerably. The magnitude of this error depends upon the grid block size (ΔX).

Figure 3.11 shows an extreme case of this error. It represents the temperature profile of the combustion front in a certain grid block from the reservoir. The width of the front may be very small compared to the grid block size which is economically possible for simulation. The temperature distribution will then be a narrow pulse with very large temperature maximum values. Numerical dispersion will smear the pulse out, increasing the width and drastically decreasing the largest predicted temperature values. If this value is below the temperature in which the burning reaction starts, the physical combustion will cease and no more thermal energy will be generated.

The case study is run with 5, 10, 20, and 40 grid blocks to investigate the grid size effects. Figures 3.12a-b and 3.13a-b show the temperature and oil saturation profiles of the reservoir after 60 and 90 days.

When 5 grid blocks are used, these figures show a sharp decrease in the peak temperature as well as a change in its location. Also, the oil saturation profile shows an overall decrease. Using 10 grid blocks, these figures show that the temperature and oil saturation profiles are close to those of 20 grid blocks. Therefore, considering the large CPU time needed for a full simulation run with 20 grid blocks compared with that needed with 10 grid blocks, it can be concluded that, for this specific data set, 10 grid blocks are acceptable.

Other schemes of calculating the mobility terms have been used to reduce the grid size effect. One of these schemes is the one adapted by Vaughan [1986]. He used an explicit form of the two-point upstream weighting. This scheme is also one of the schemes adapted by ISCOM although ISCOM's users have a choice of other schemes too. Wheatley [1979] adapted an implicit form of the two-point upstream weighting scheme and claimed that this version is more accurate than the single-point upstream.

It is not within the scope of this project to discuss each method comprehensively. However, these schemes have been added to the program for future investigation. Furthermore, the simulator was run using each of those schemes with 5, 10, 20, and 40 grid blocks and figures 3.14a-b to 3.17a-b show the temperature and oil saturation profiles of the reservoir after 60 and 90 days.

It has been found that Vaughn's scheme leads to an oscillatory behavior in the iteration scheme which causes an increase in the CPU time needed for the run. Also, no improvement in the accuracy is gained from this scheme. On the other hand, using Wheatley's scheme leads to a higher peak temperature when 10 grid blocks is used as shown in figure 3.16a as well as no improvement in accuracy is obtained when using 5 grid blocks.

The problem of grid block size has been addressed by many researchers. The methods and techniques developed to tackle this problem are summarized in Table 3.1.

It is not within the scope of this project to adapt any of these or develop a new approach. However, it is recommended the use of the Ewing and Lazarov [1988] approach to be added to the model in order to tackle the grid size problem in field scale simulation.

Table 3.1 Techniques For Solving The Grid Size Problem

Technequ e	References
Front tracking	Hwang, Jones, and Odeh [1980] Davies [1988]
Activation temperature	Coats [1983]
Pseudo kinetics scheme	Ito, and Chow [1988]
Grid refinement	Heinnemann, Gerken, von Hantelmann [1983] Han, Han, Yan, Peng [1987] Ewing [1986] Ewing and Lazarov [1988]

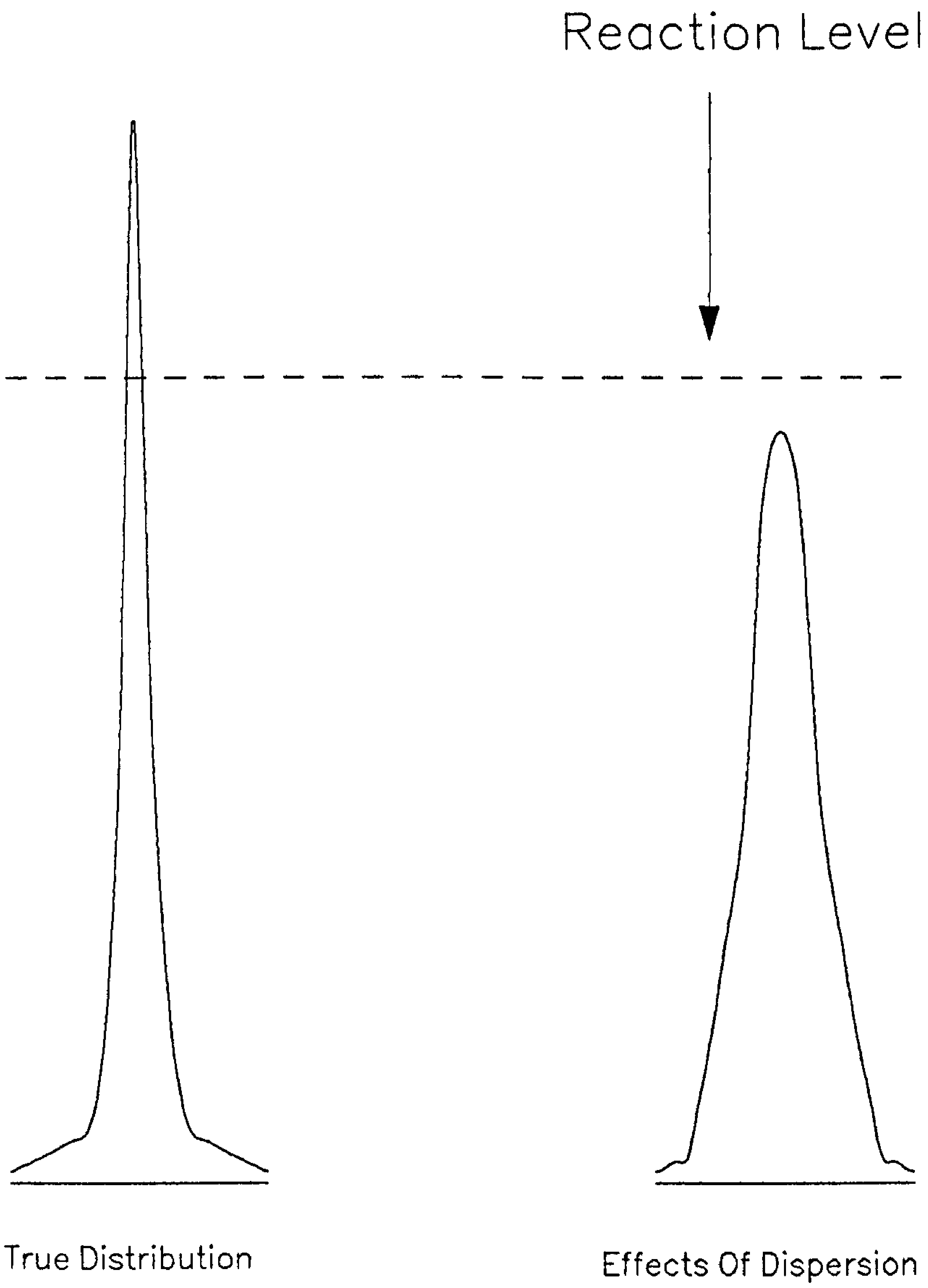


Figure 3.11 Combustion Front Resolution

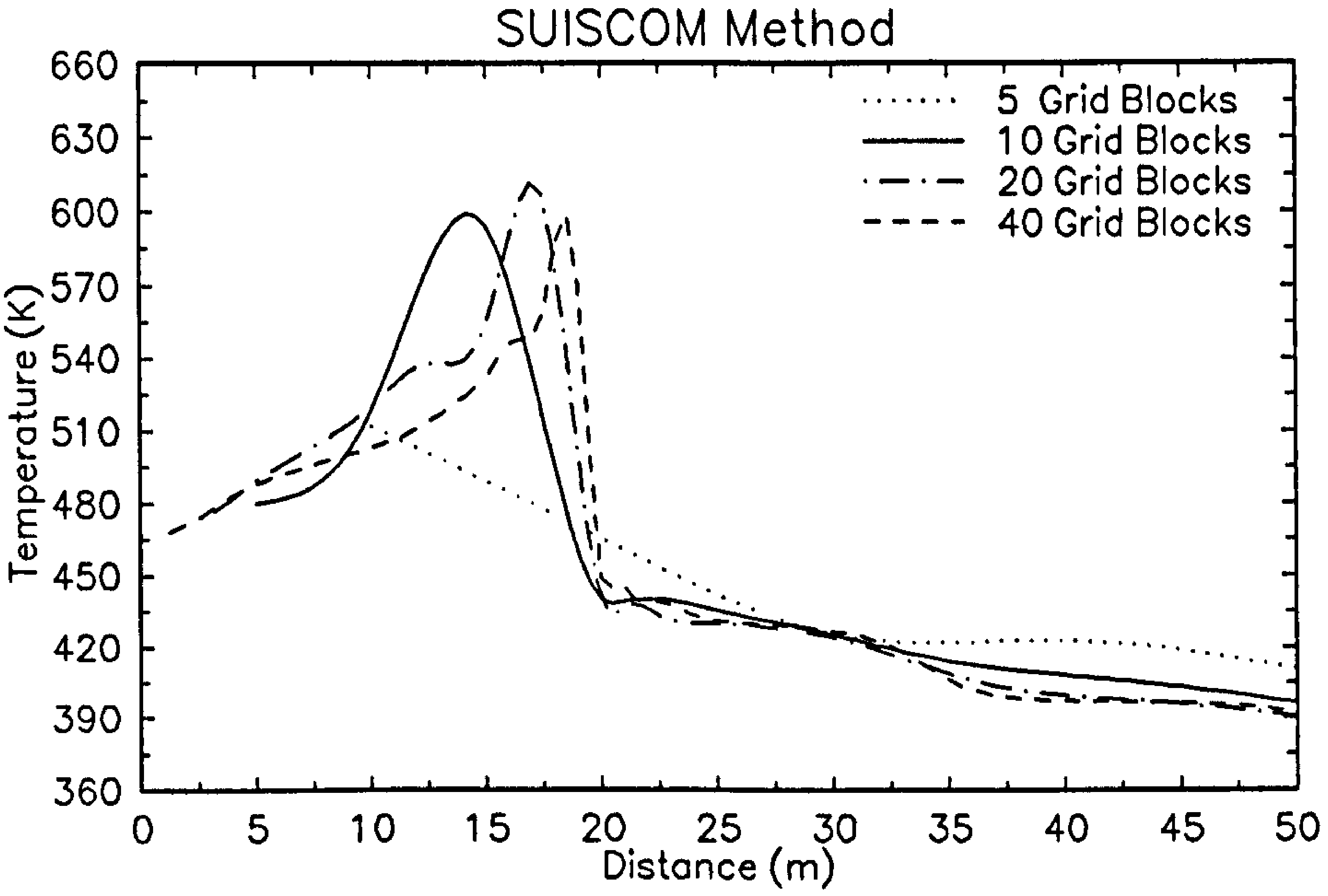


Figure 3.12a Temperature Profile Over The Reservoir (After 60.0 Days)

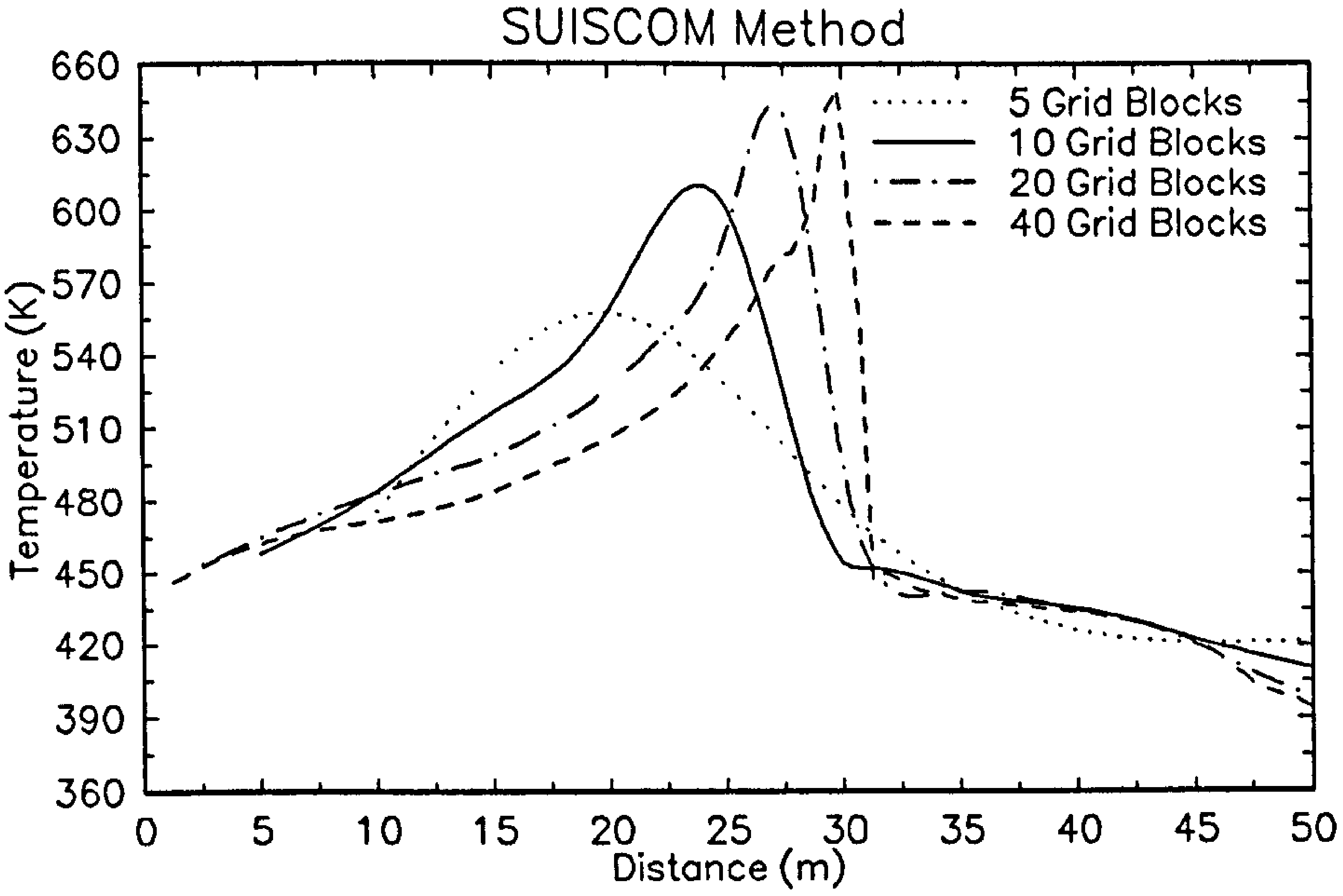
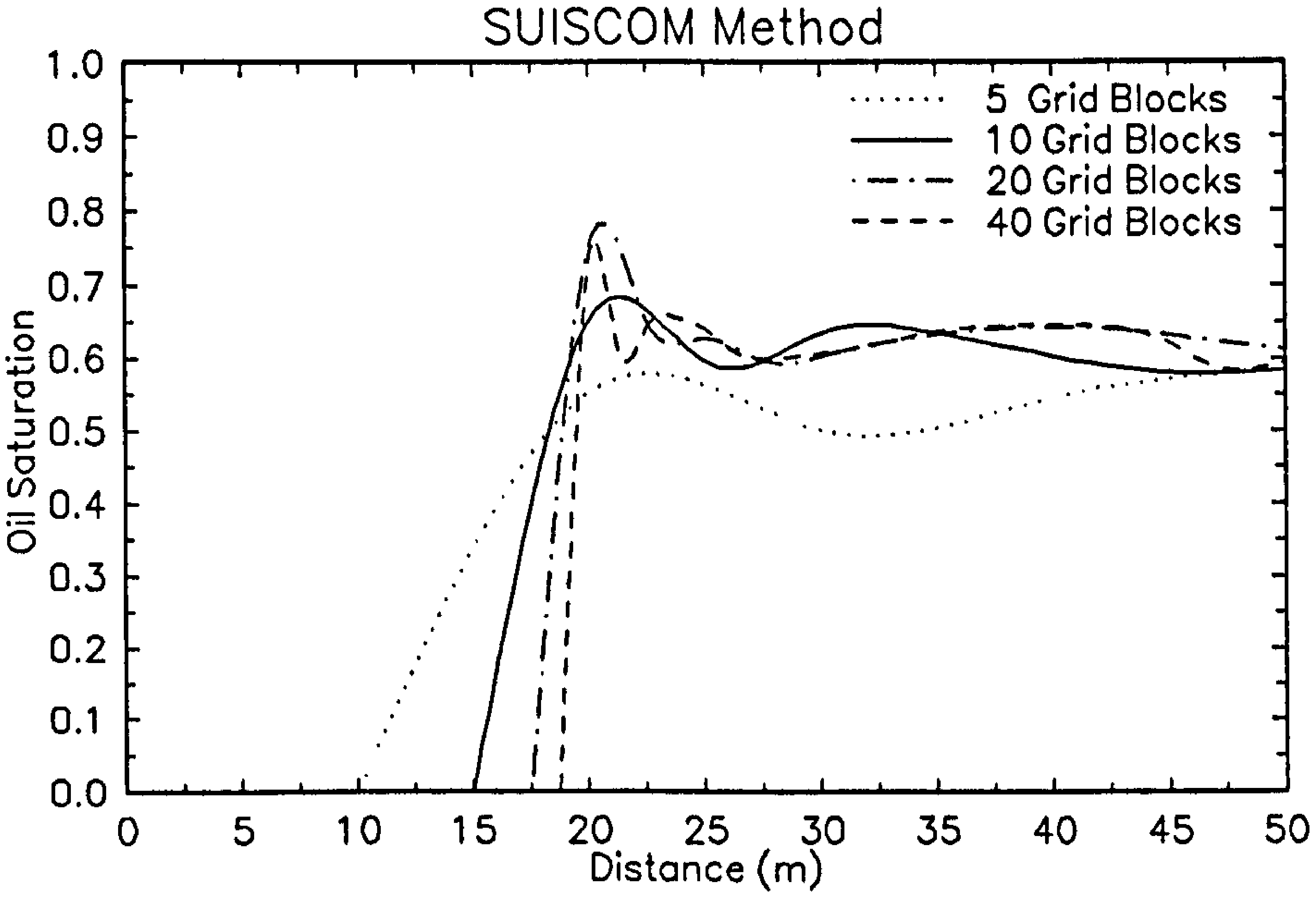
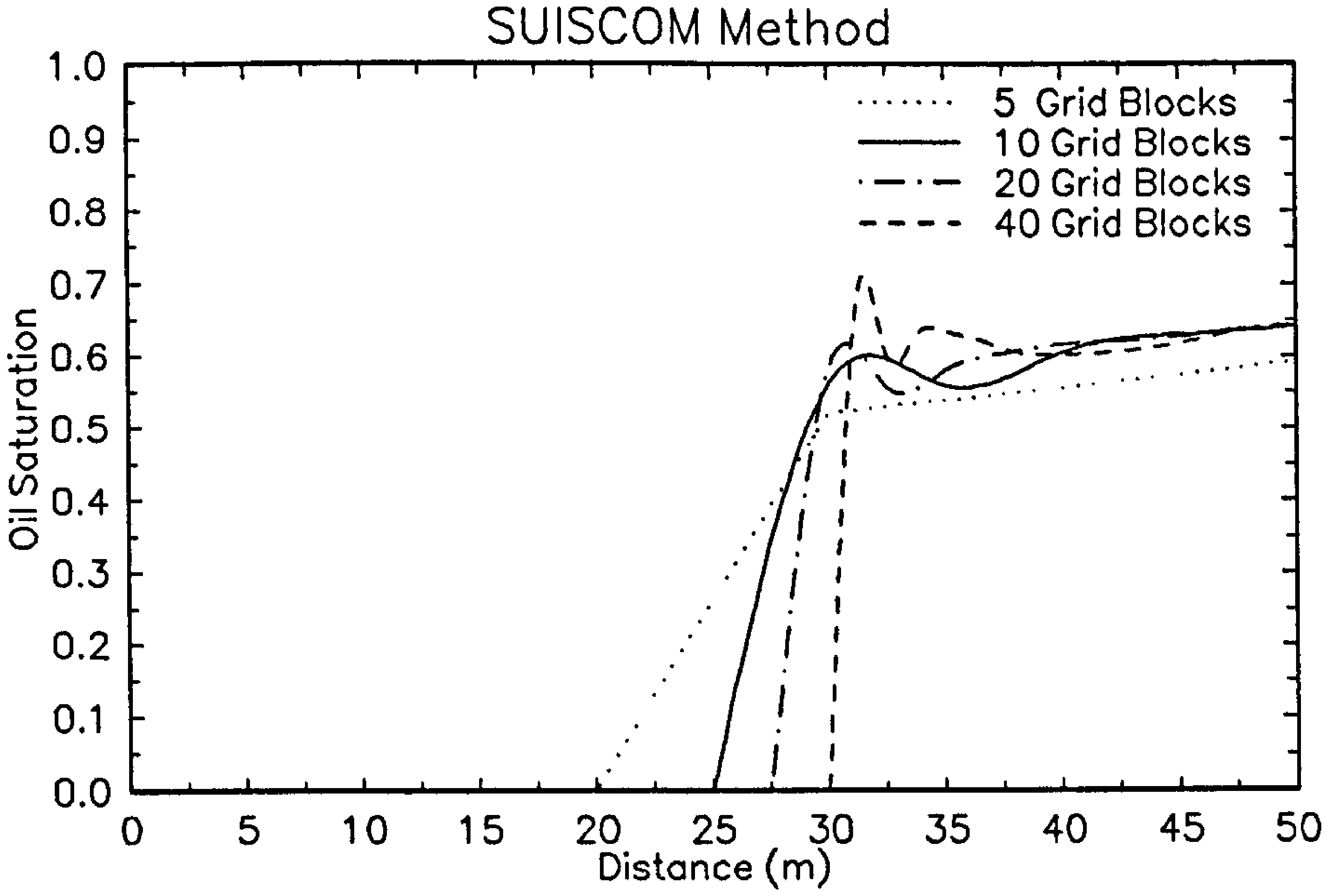


Figure 3.12b Temperature Profile Over The Reservoir (After 90.0 Days)



SUISCOM Method

Figure 3.13a Oil Saturation Profile Over The Reservoir (After 60.0 Days)



SUISCOM Method

Figure 3.13b Oil Saturation Profile Over The Reservoir (After 90.0 Days)

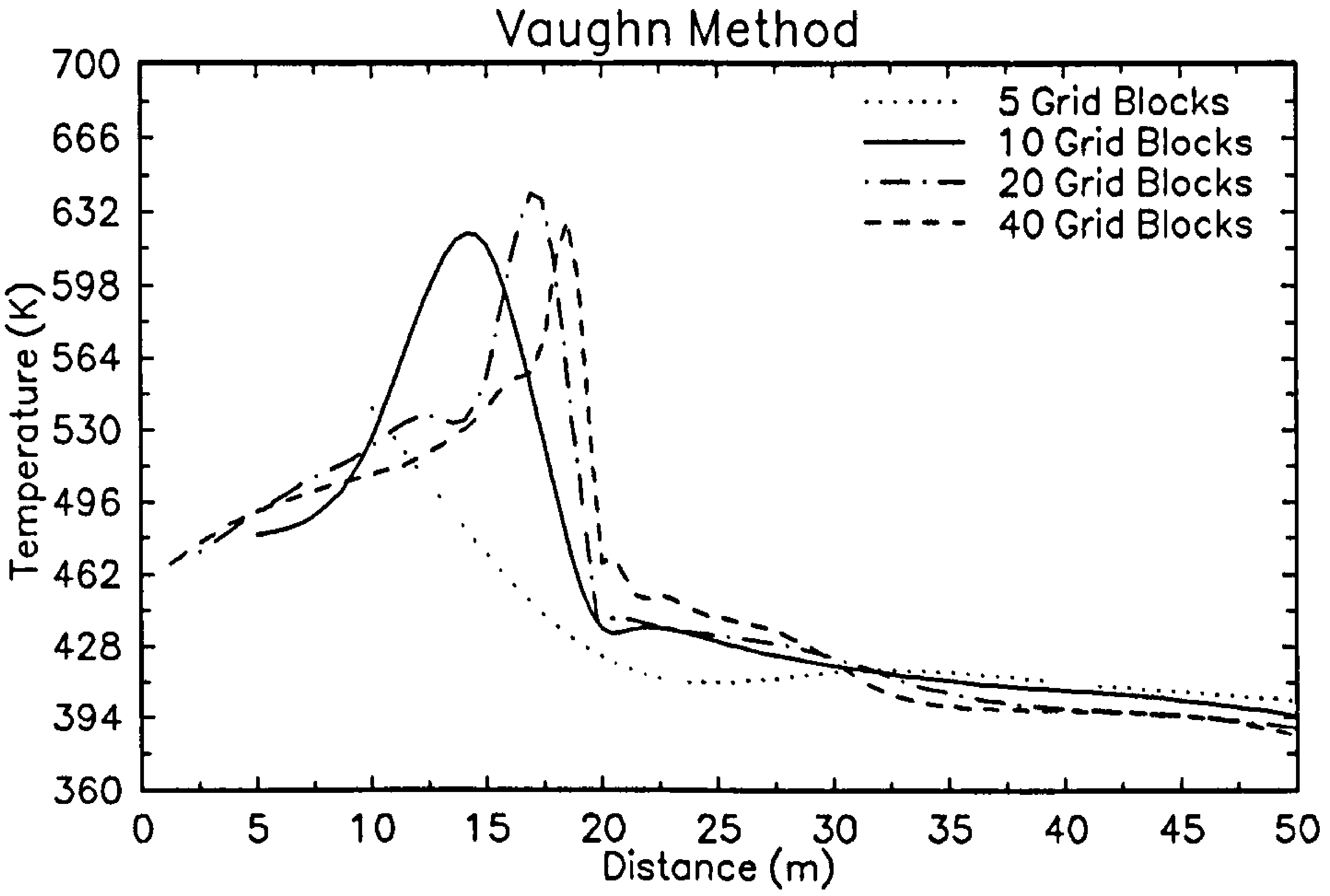


Figure 3.14a Temperature Profile Over The Reservoir (After 60.0 Days)

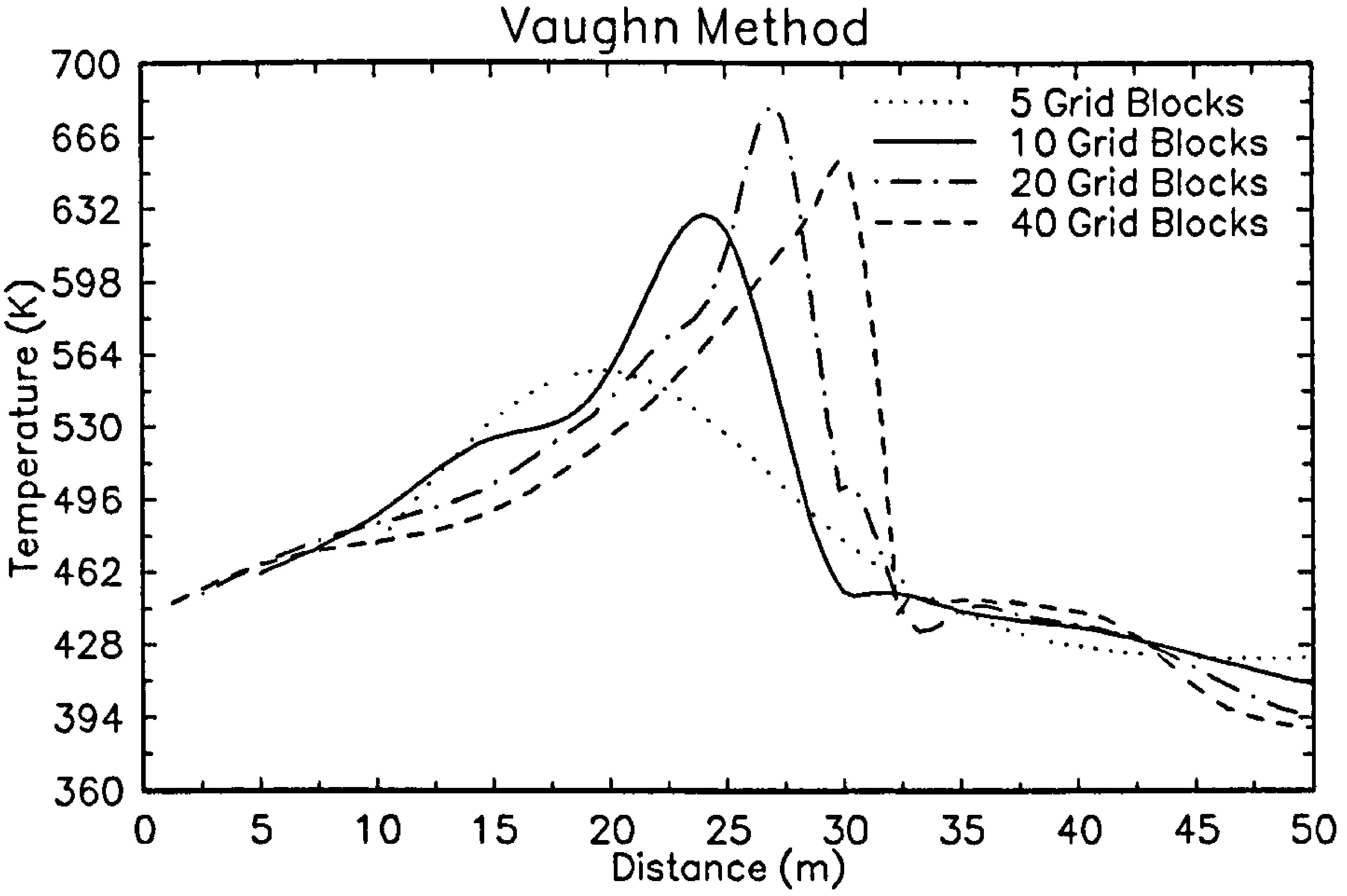


Figure 3.14b Temperature Profile Over The Reservoir (After 90.0 Days)

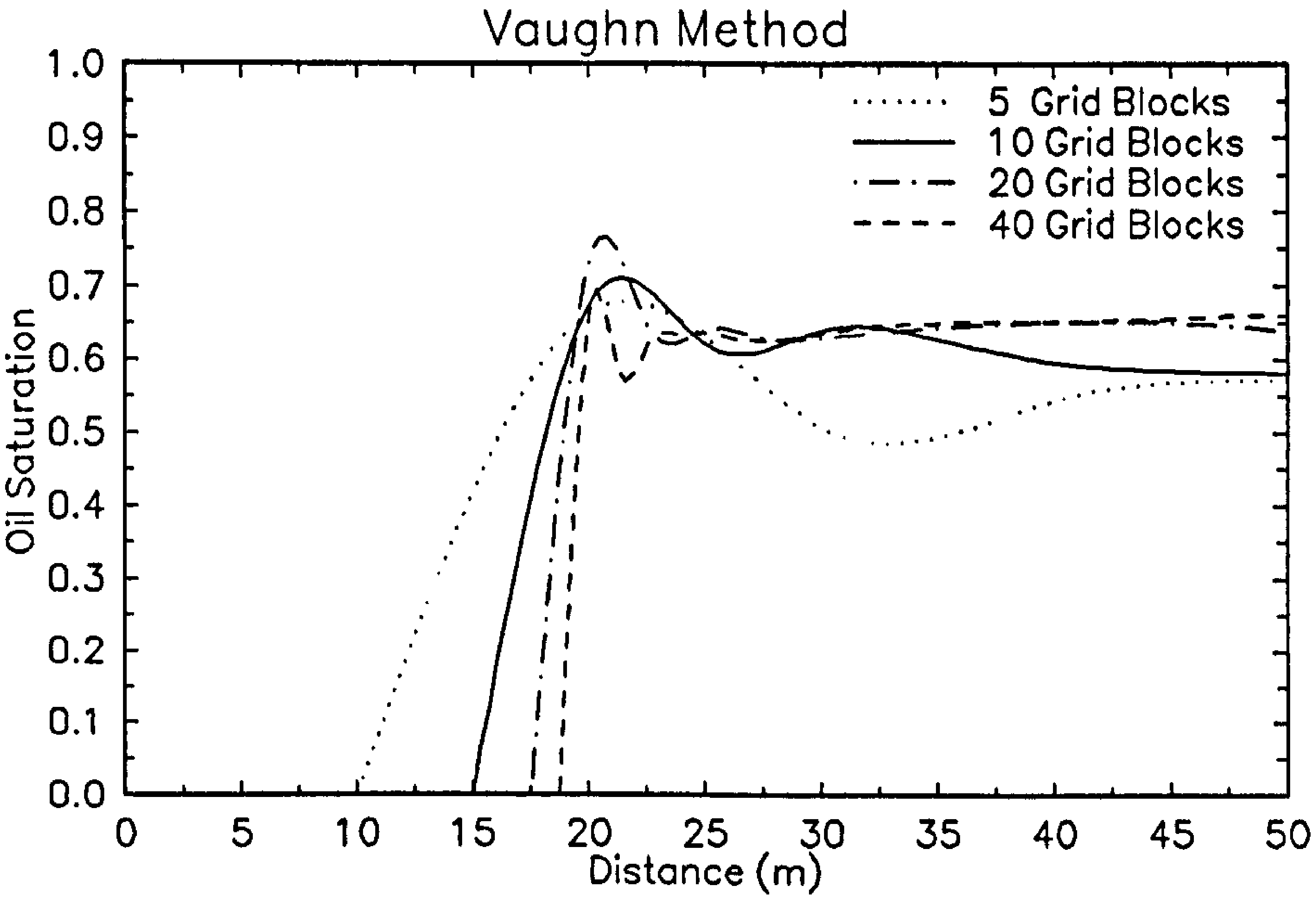


Figure 3.15a Oil Saturation Profile Over The Reservoir (After 60.0 Days)

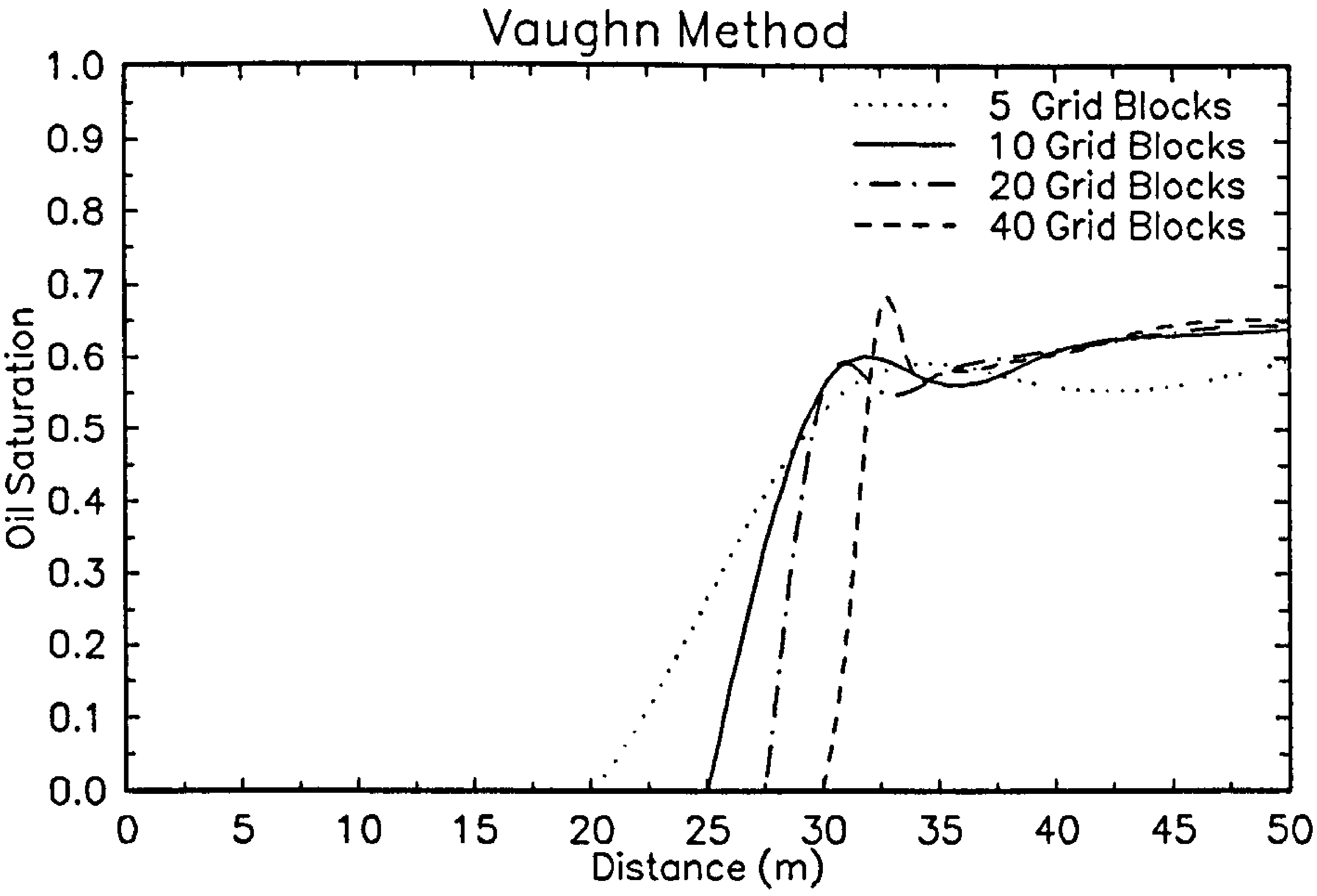


Figure 3.15b Oil Saturation Profile Over The Reservoir (After 90.0 Days)

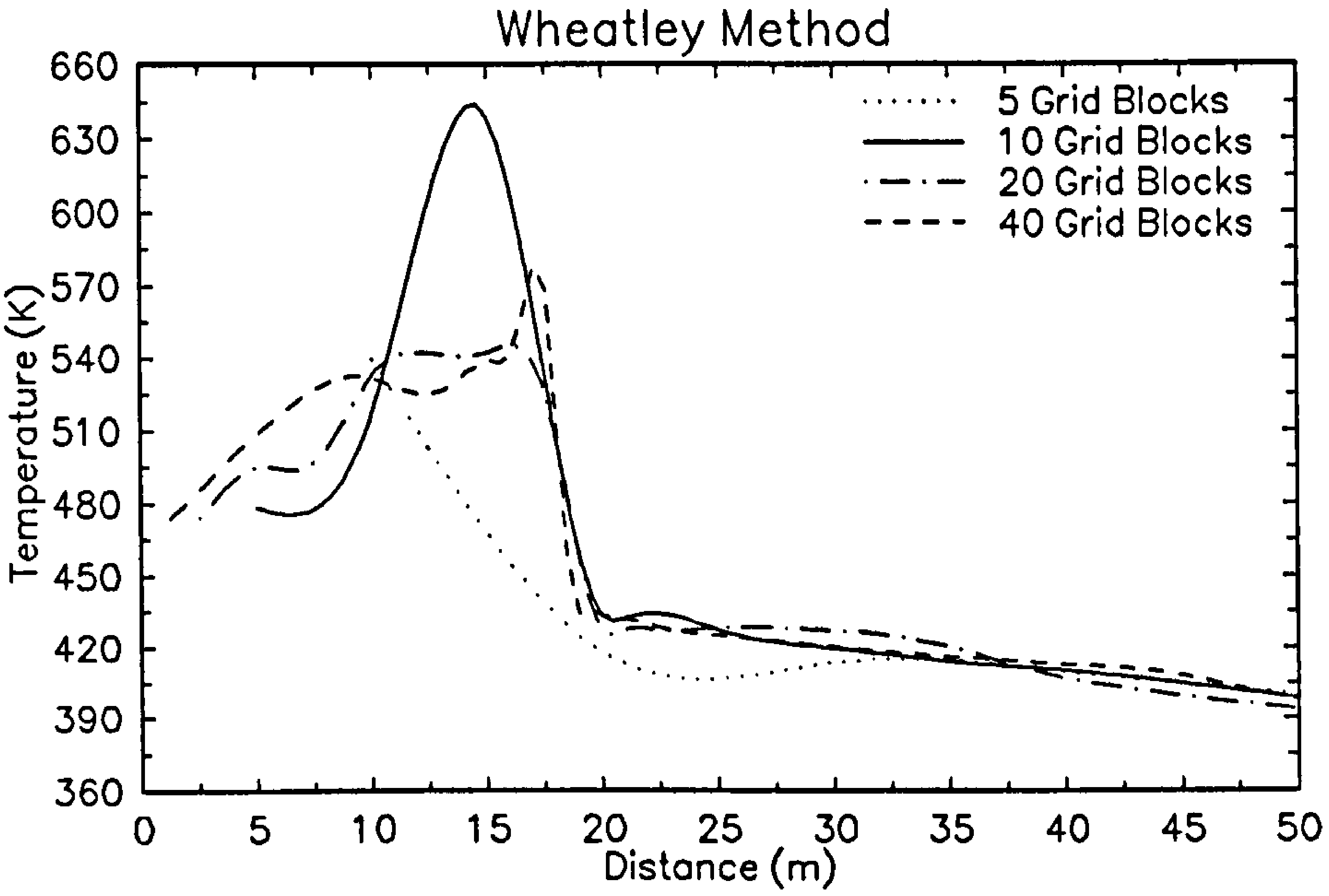

Wheatley Method

Figure 3.16a Temperature Profile Over The Reservoir (After 60.0 Days)

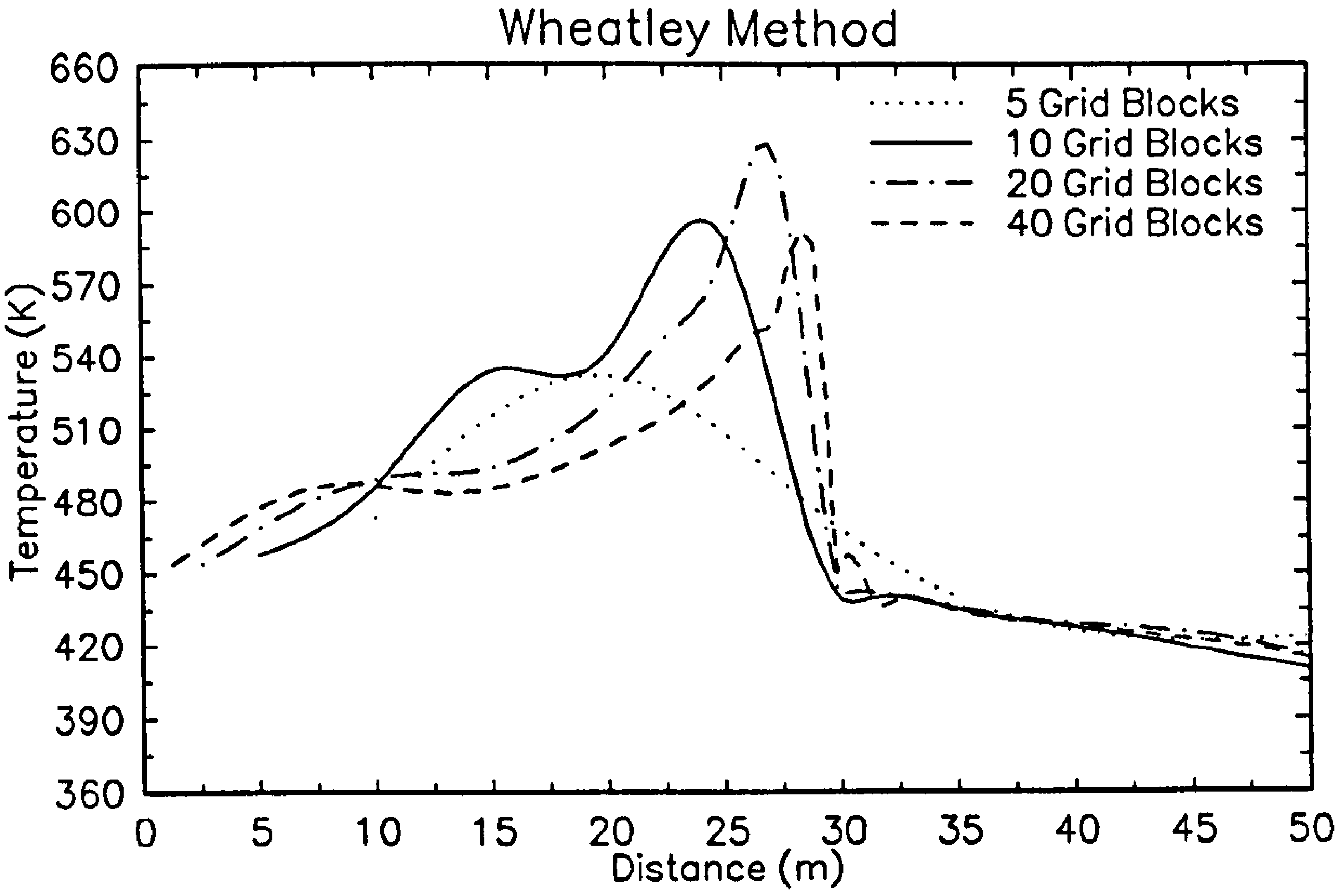

Wheatley Method

Figure 3.16b Temperature Profile Over The Reservoir (After 90.0 Days)

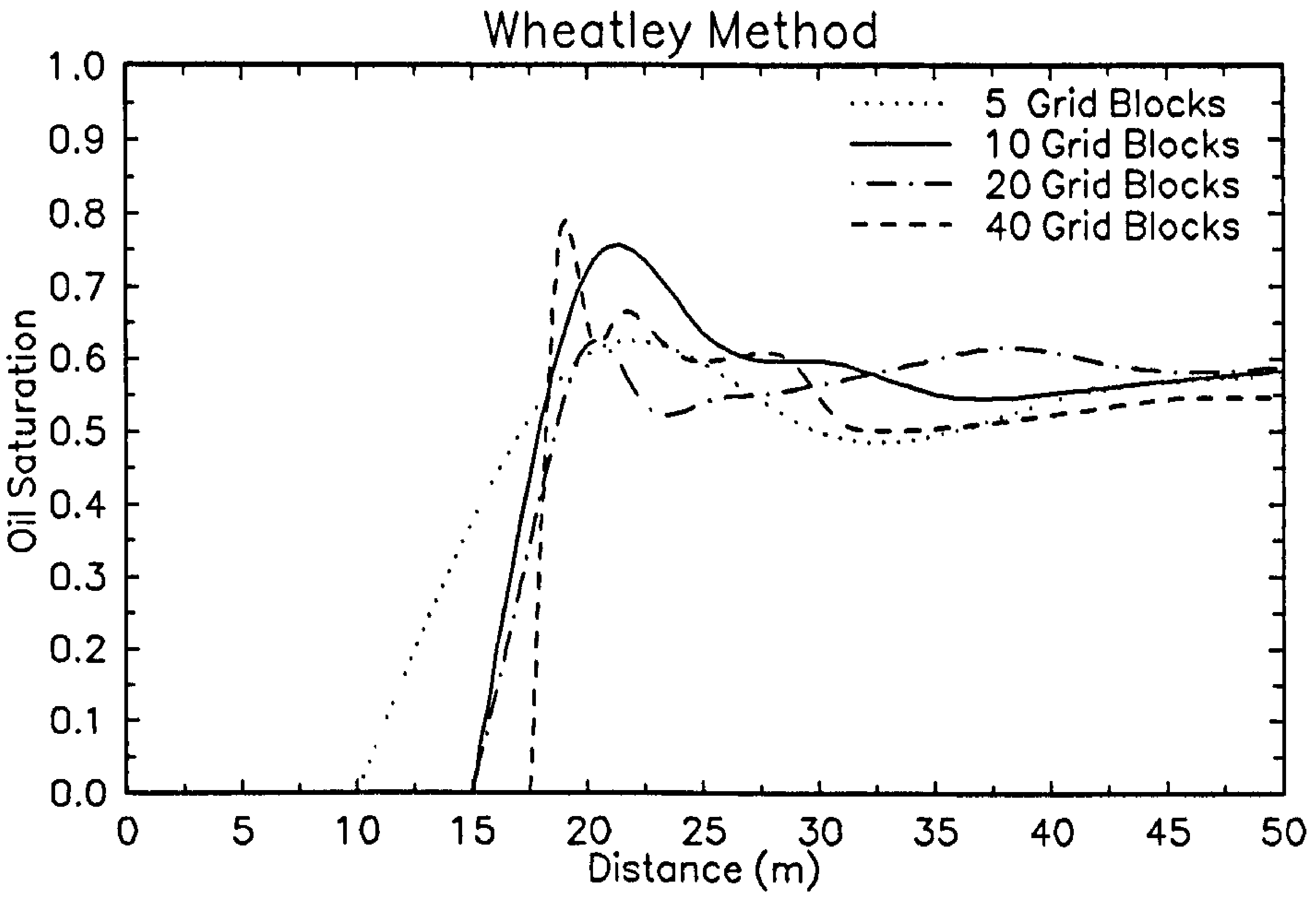


Figure 3.17a Oil Saturation Profile Over The Reservoir (After 60.0 Days)

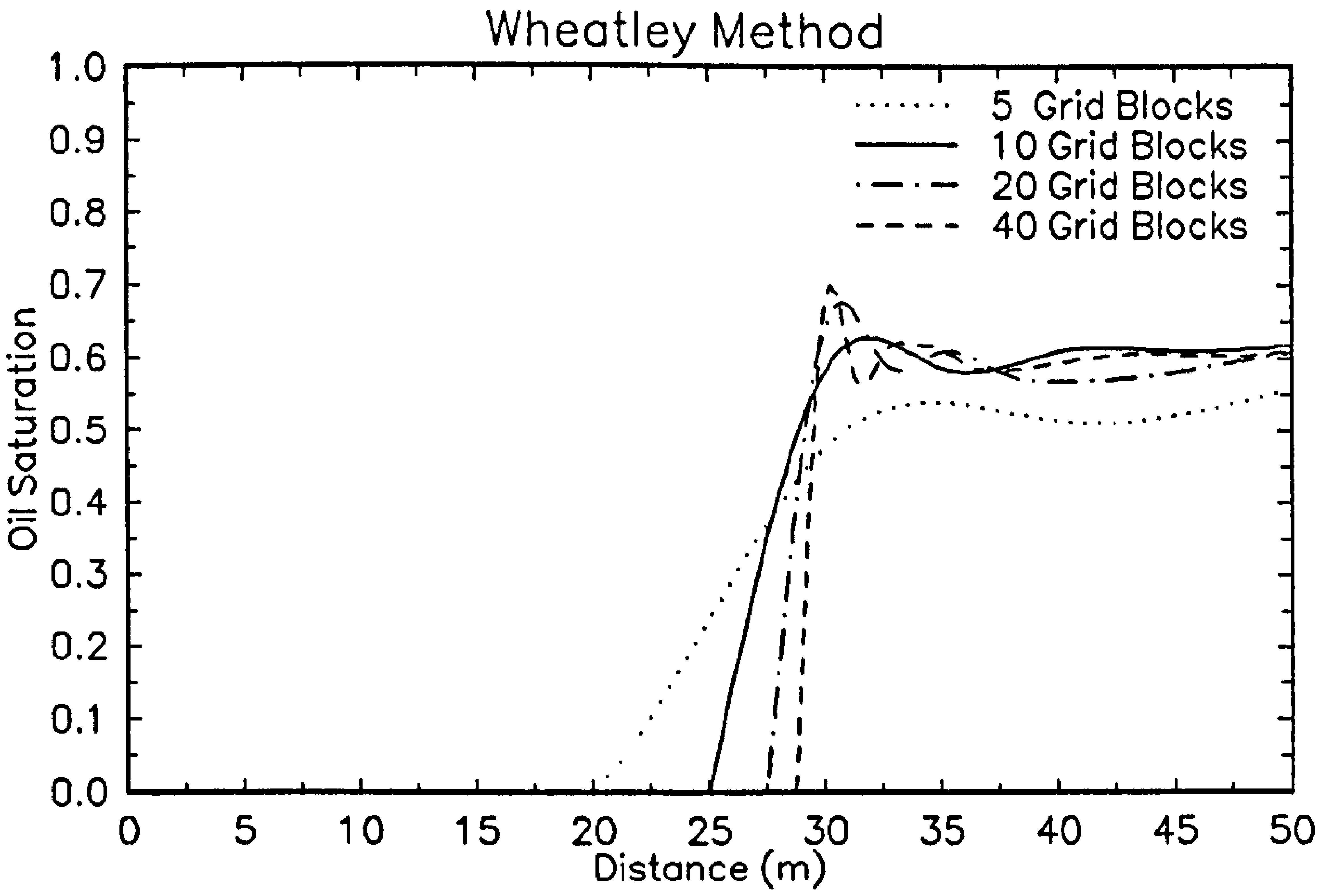


Figure 3.17b Oil Saturation Profile Over The Reservoir (After 90.0 Days)

Chapter Four

A SENSITIVITY STUDY OF THE EFFECT OF PARAMETERS ON THE PERFORMANCE CHARACTERISTICS OF THE IN-SITU COMBUSTION PROCESS

4.1 Introduction

This study has been made for two reasons; firstly, to indicate which data are ‘important’ in the sense of having a significant influence on the performance characteristics of the process and secondly, to test the robustness of the program to the changes in the input parameters (these are exaggerated in some cases). This study is necessary before making any judgment regarding the simulator.

The parameters investigated were water recovery, oil recovery, light oil recovery, the pressure, the average peak temperature, the total production time, and the heat losses to the surrounding strata.

Water recovery is a volumetric water recovery expressed as a percentage of the original water in place (OWIP). Also, oil and light oil recovery are volumetric recoveries expressed as a percentage of the original oil in place (OOIP). Stock tank oil densities were taken as 4274.0 and 8612.0 gmol/m^3 for heavy and light oil, respectively, at 298 K and 450 kPa. Initial water and oil in place are 839.49 and 1906.10 m^3 respectively. The first and last grid block peak temperatures were not included in the calculation of the average peak temperature and the heat loss to the over and under burden strata is expressed as a percentage of the total heat of reaction.

Finally, it should be emphasized that all conclusions made in this chapter are only valid for this one set of data and should not be generalized without a more comprehensive study.

4.2 Effects Of Operating Parameters

4.2.1 The Effect Of The Injection Gas Flow Rate

Tables 4.1, 4.2, and 4.3 show the effects of the changes in the injection gas flow rate at different oxygen concentrations. Inspection of the results shows an increase of reservoir pressures with increase in the injection flow rate. This increase in the pressure is also shown in figures 4.1, 4.2, and 4.3 which present the pressure profiles of a typical grid block from the reservoir (grid block three). The fluctuations shown in the pressure profiles are due to the water and oil banking which occur downstream of the combustion front.

Since the peak temperature is a function of the amount of the fuel available, it can be seen from tables 4.1-4.3 that an increase in the injection flow rates did not have a significant effect on the temperature; this is because the fuel production reaction rate (heavy oil cracking) is not a function of the pressure.

Since the heat loss terms shown in tables 4.1-4.3 represent the ratio between the heat loss and the total heat generated by the four reactions, the decrease shown in these terms with increase in flow rate is actually a result of the overall increase in the heat generated by the four reactions.

Figures 4.4, 4.5, and 4.6 show the corresponding oil recovery profiles, figures 4.7, 4.8, and 4.9 show the water recovery profiles and figures 4.10, 4.11, and 4.12 show the light oil recovery profiles for the effect of injection flow rate at different oxygen concentrations. A significant increase in the oil and water produced can be seen at a specific time although the total amount of oil and water which can be recovered regardless of the time did not change significantly. Also, these figures show a significant decrease in the time required to achieve maximum recovery at higher flow rates which explains the termination of the higher flow rate curves at an earlier time.

Table 4.1 The effect of the injected gas flow rate

OXYGEN CONCENTRATION = 100%			
VARIABLES	FLOW RATE (MOL/day)		
	$1.36\ 10^5$	$2.04\ 10^5$	$2.72\ 10^5$
WATER RECOVERY (% OF OWIP)	117.4	117.2	117.4
TOTAL OIL RECOVERY (% OOIP)	95.3	93.6	92.8
LIGHT OIL RECOVERY (% OOIP)	26.3	22.53	20.6
AVERAGE PEAK TEMPERATURE (K)	622.0	620.0	621.8
MAXIMUM PRESSURE (kPa)	1242.0	1369.3	1558.3
TOTAL PRODUCTION TIME (day)	145.0	95.1	72.0
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	25.0	22.3	20.6

Table 4.2 The effect of the injected gas flow rate

OXYGEN CONCENTRATION = 50%			
VARIABLES	FLOW RATE (MOL/day)		
	$1.36\ 10^5$	$2.04\ 10^5$	$2.72\ 10^5$
WATER RECOVERY (% OF OWIP)	117.6	117.5	118.0
TOTAL OIL RECOVERY (% OOIP)	95.1	93.0	91.9
LIGHT OIL RECOVERY (% OOIP)	30.0	25.1	22.6
AVERAGE PEAK TEMPERATURE (K)	604.4	594.6	597.5
MAXIMUM PRESSURE (kPa)	833.6	1029.6	1177.9
TOTAL PRODUCTION TIME (day)	309.6	198.2	149.0
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	27.8	24.8	23.0

Table 4.3 The effect of the injected gas flow rate

OXYGEN CONCENTRATION = 20%			
VARIABLES	FLOW RATE (MOL/day)		
	$1.36\ 10^5$	$2.04\ 10^5$	$2.72\ 10^5$
WATER RECOVERY (% OF OWIP)	122.7	122.1	122.4
TOTAL OIL RECOVERY (% OOIP)	96.2	93.8	92.6
LIGHT OIL RECOVERY (% OOIP)	35.8	28.6	23.8
AVERAGE PEAK TEMPERATURE (K)	578.6	568.0	554.0
MAXIMUM PRESSURE (kPa)	696.5	857.4	1013.8
TOTAL PRODUCTION TIME (day)	886.7	553.0	414.6
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	30.2	27.1	25.0

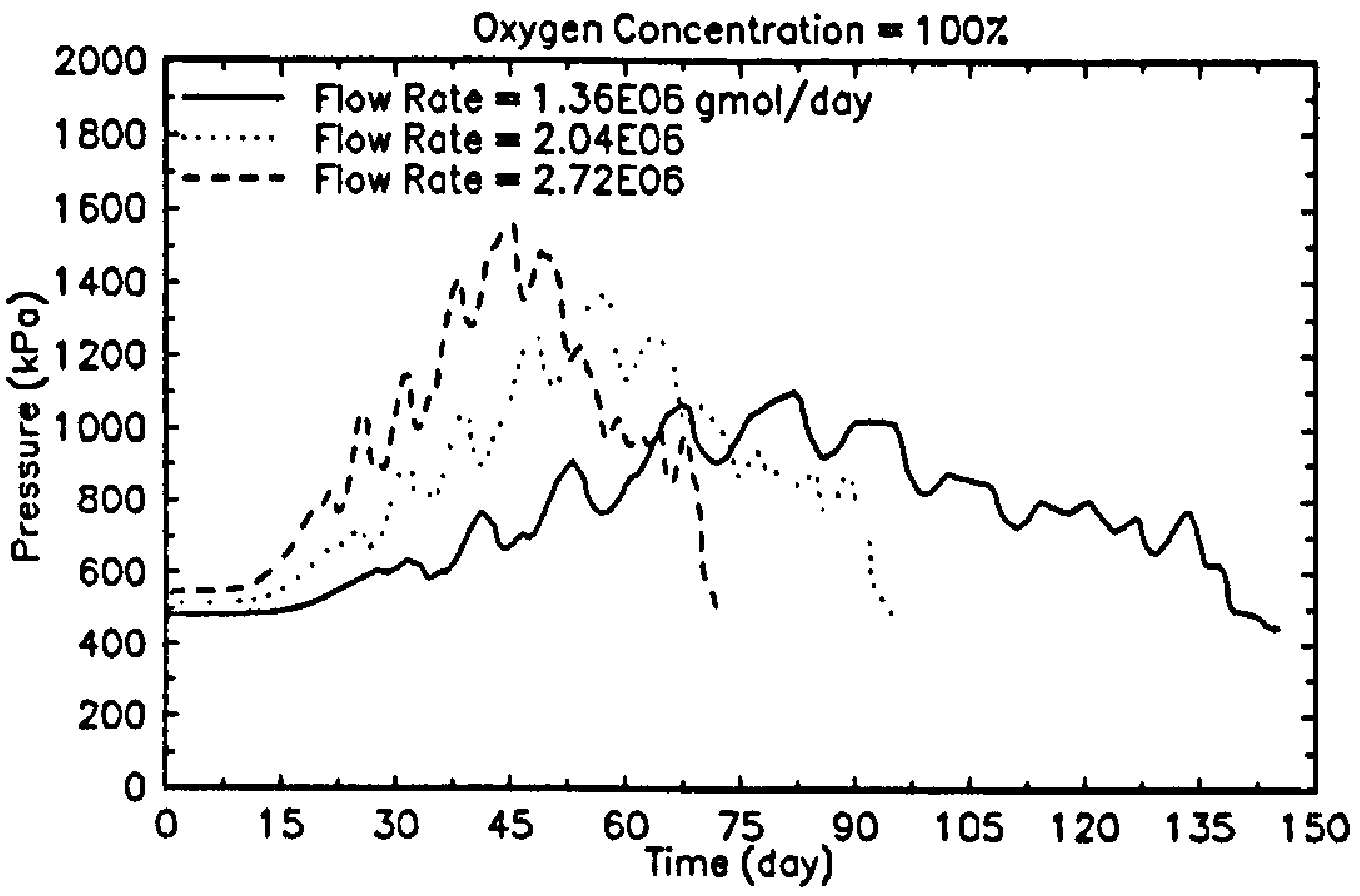


Figure 4.1 Pressure Profile Of Grid Block 3

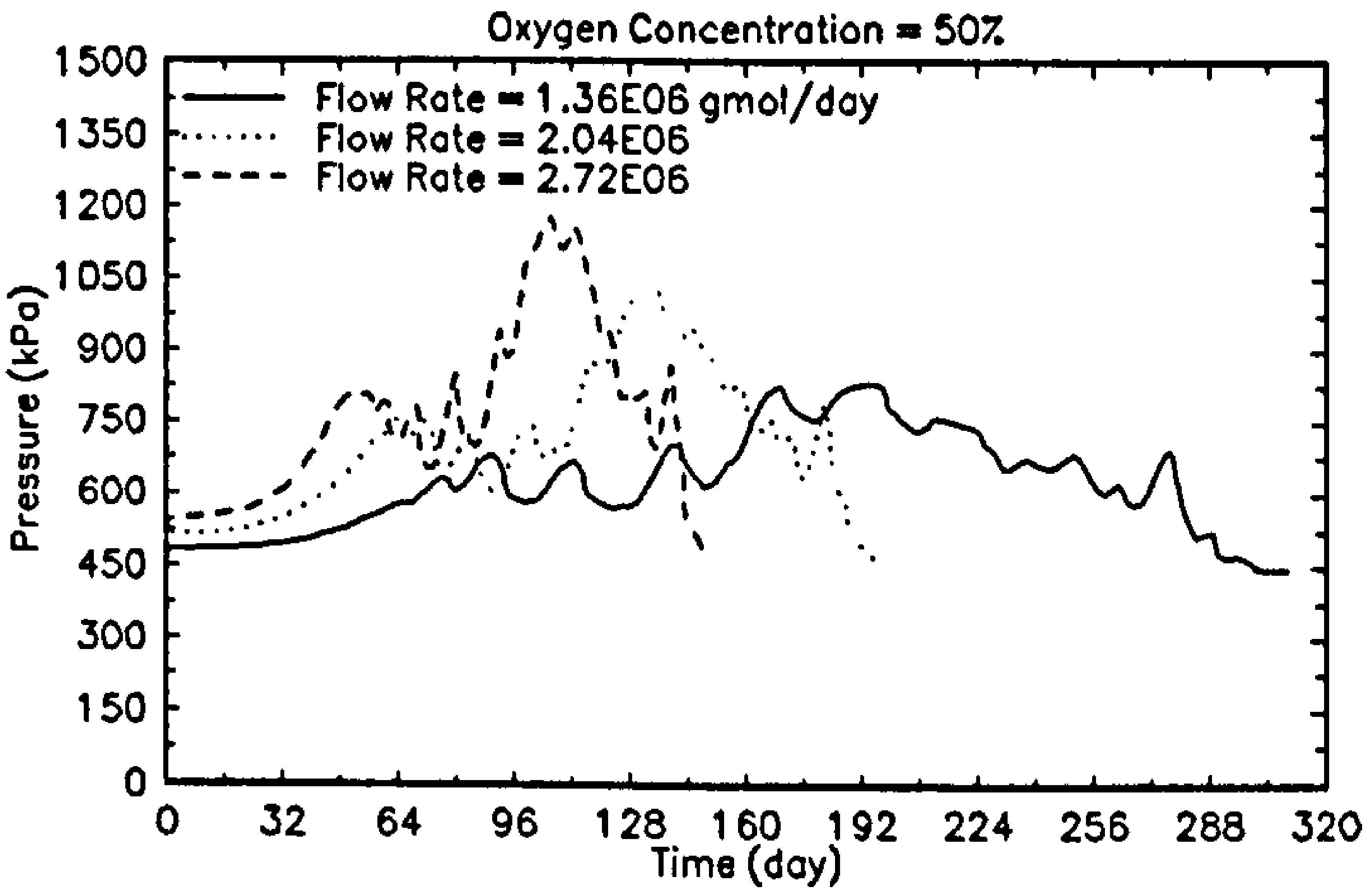


Figure 4.2 Pressure Profile Of Grid Block 3

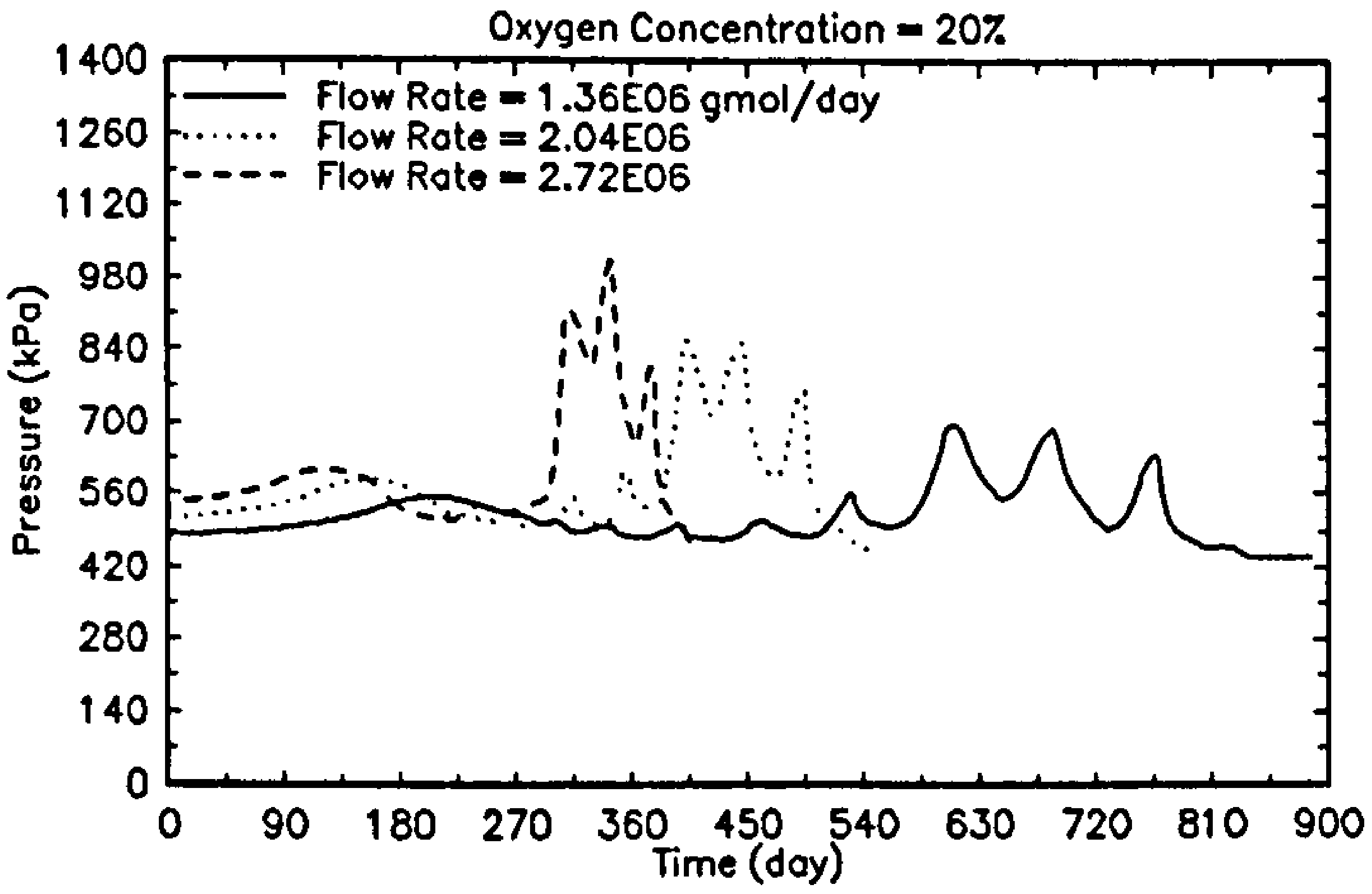


Figure 4.3 Pressure Profile Of Grid Block 3

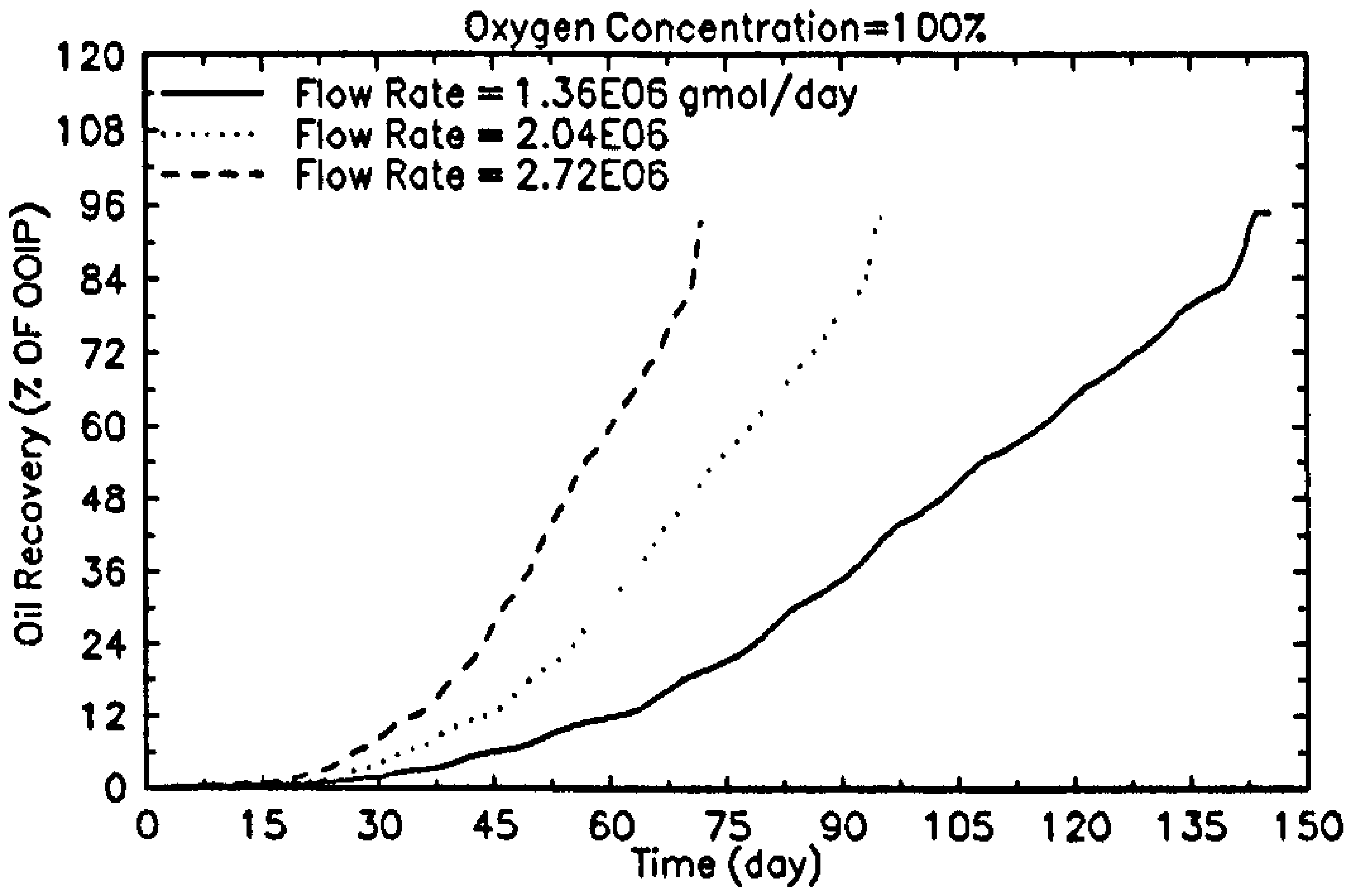


Figure 4.4 Oil Recovery Profile

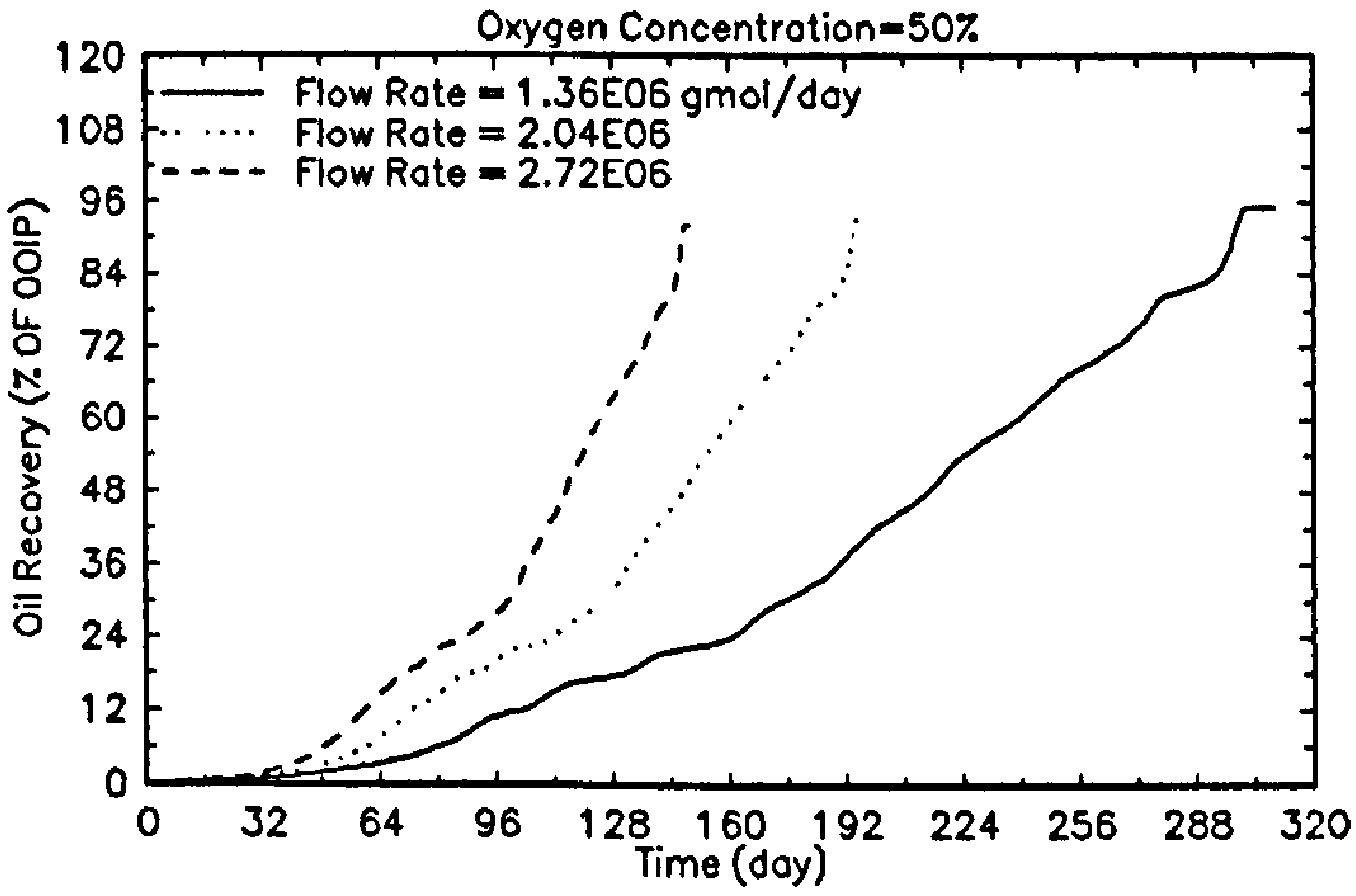


Figure 4.5 Oil Recovery Profile

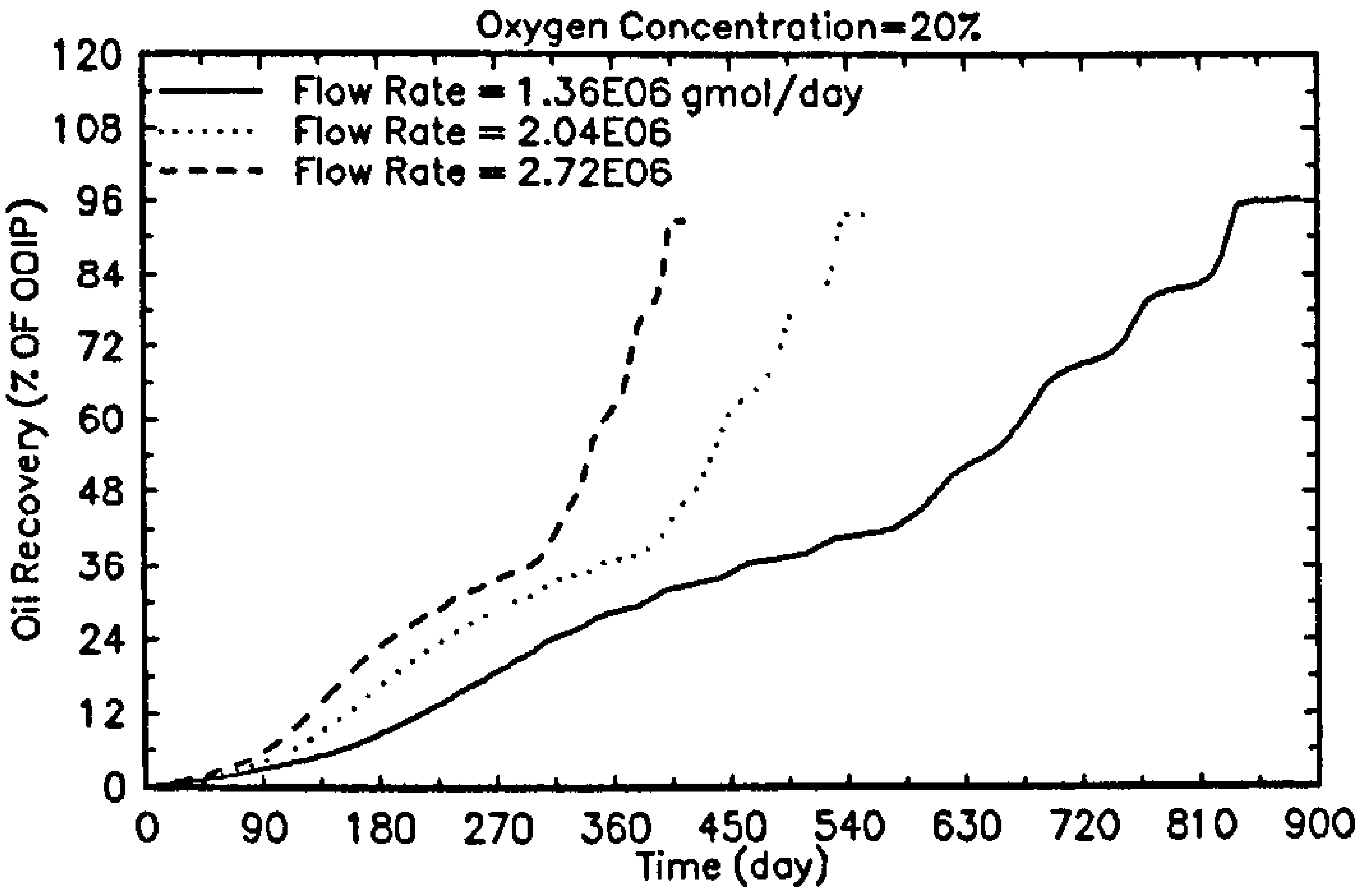


Figure 4.6 Oil Recovery Profile

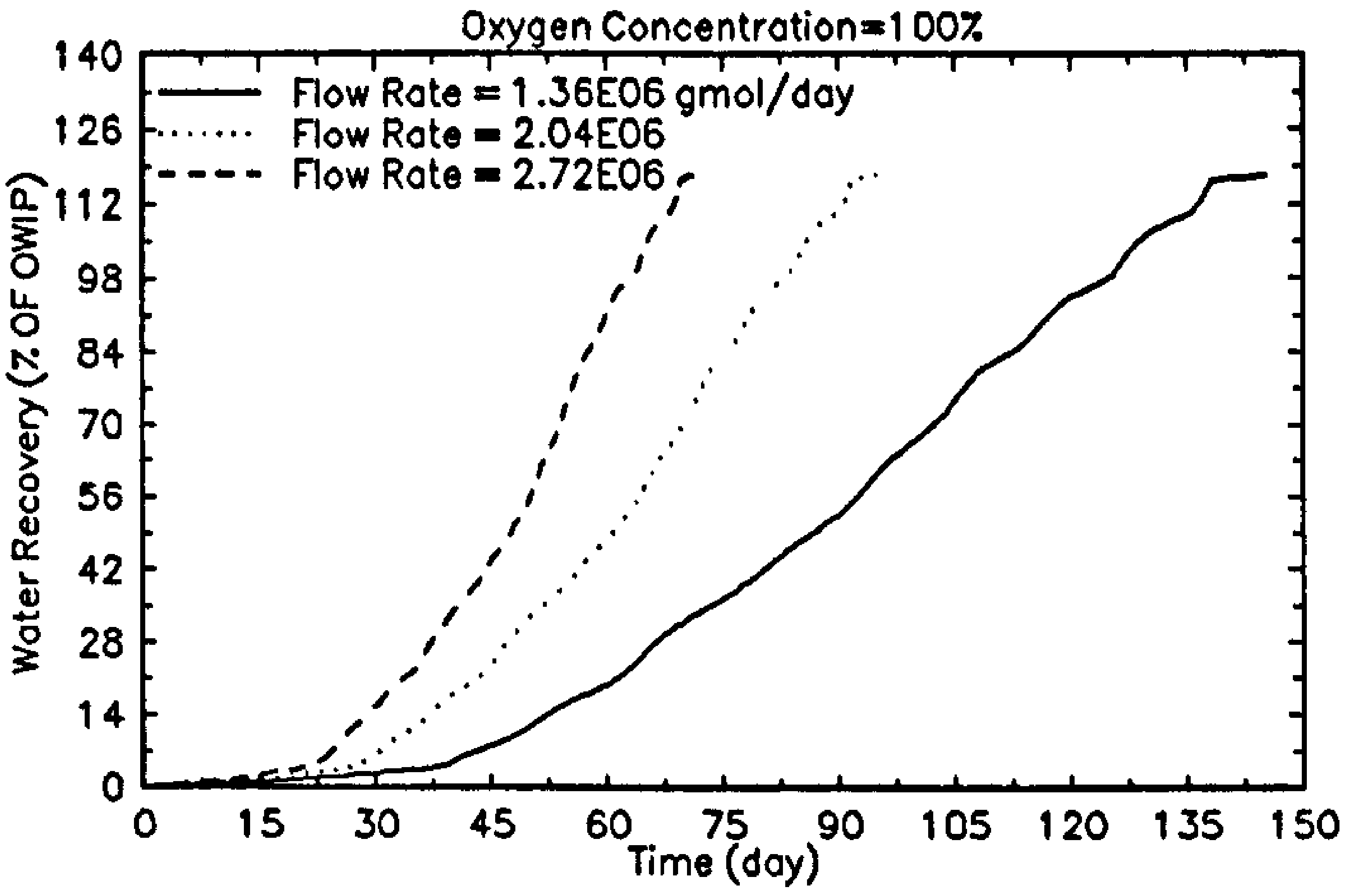


Figure 4.7 Water Recovery Profile

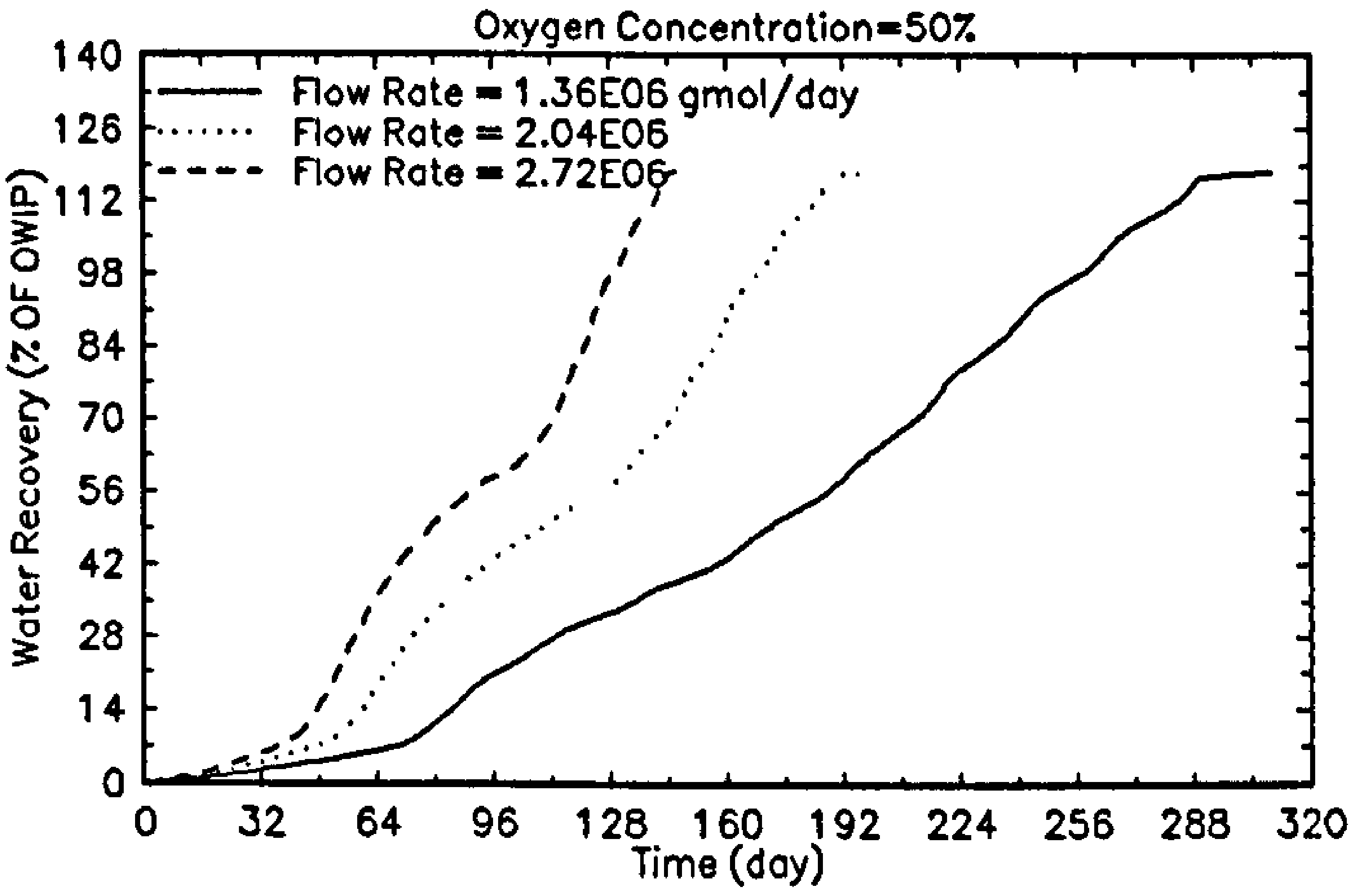


Figure 4.8 Water Recovery Profile

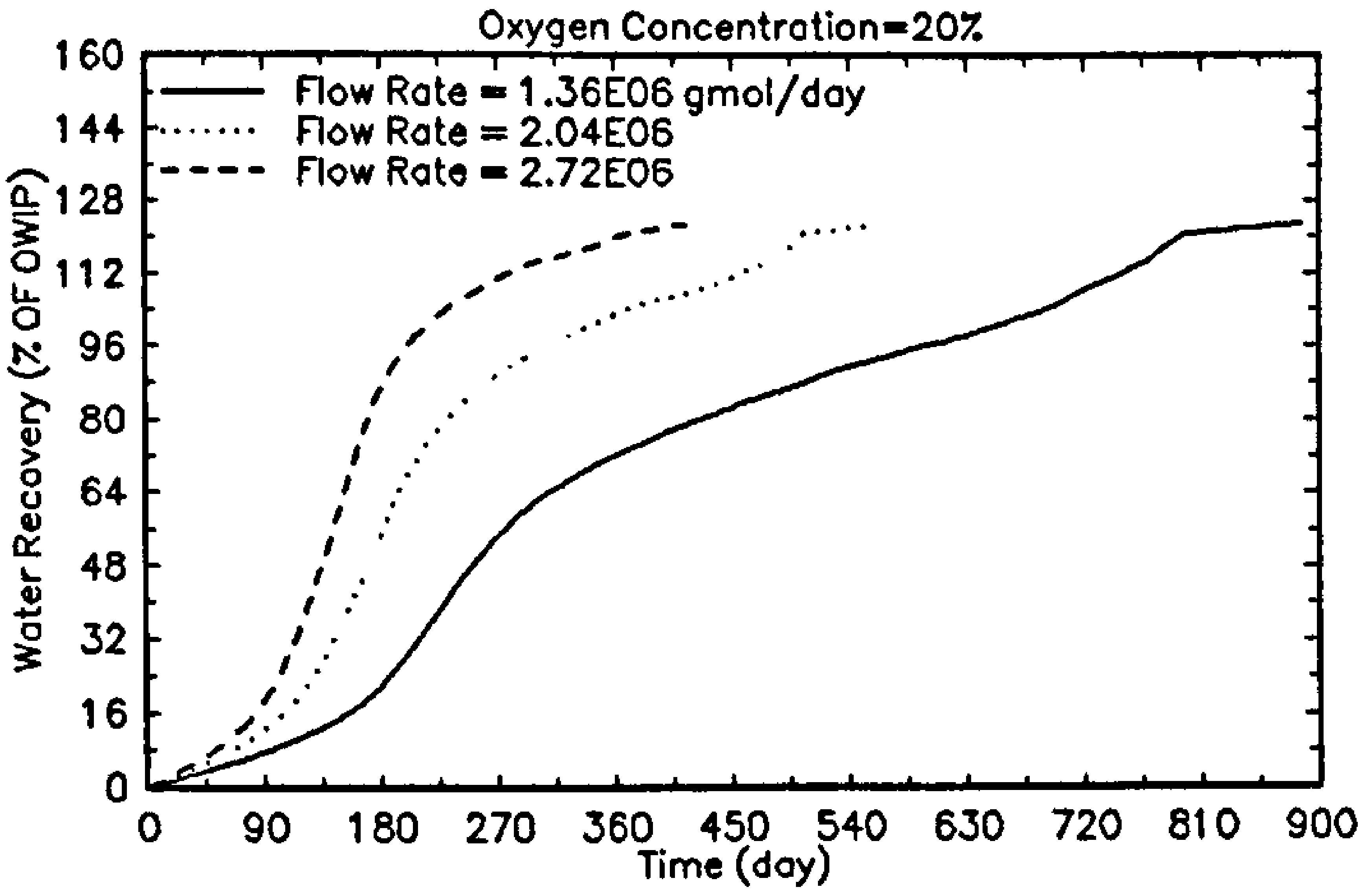


Figure 4.9 Water Recovery Profile

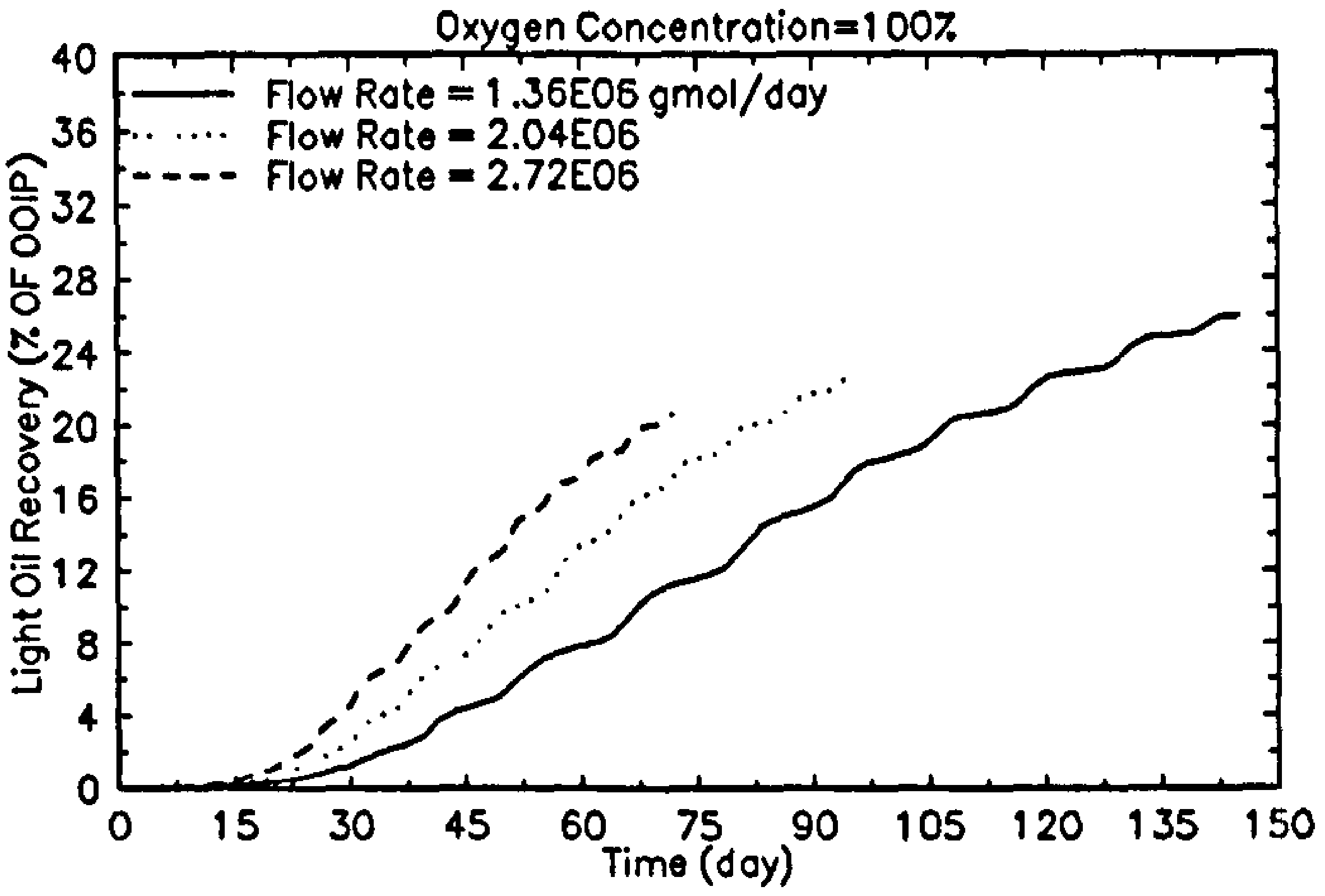


Figure 4.10 Light Oil Recovery Profile

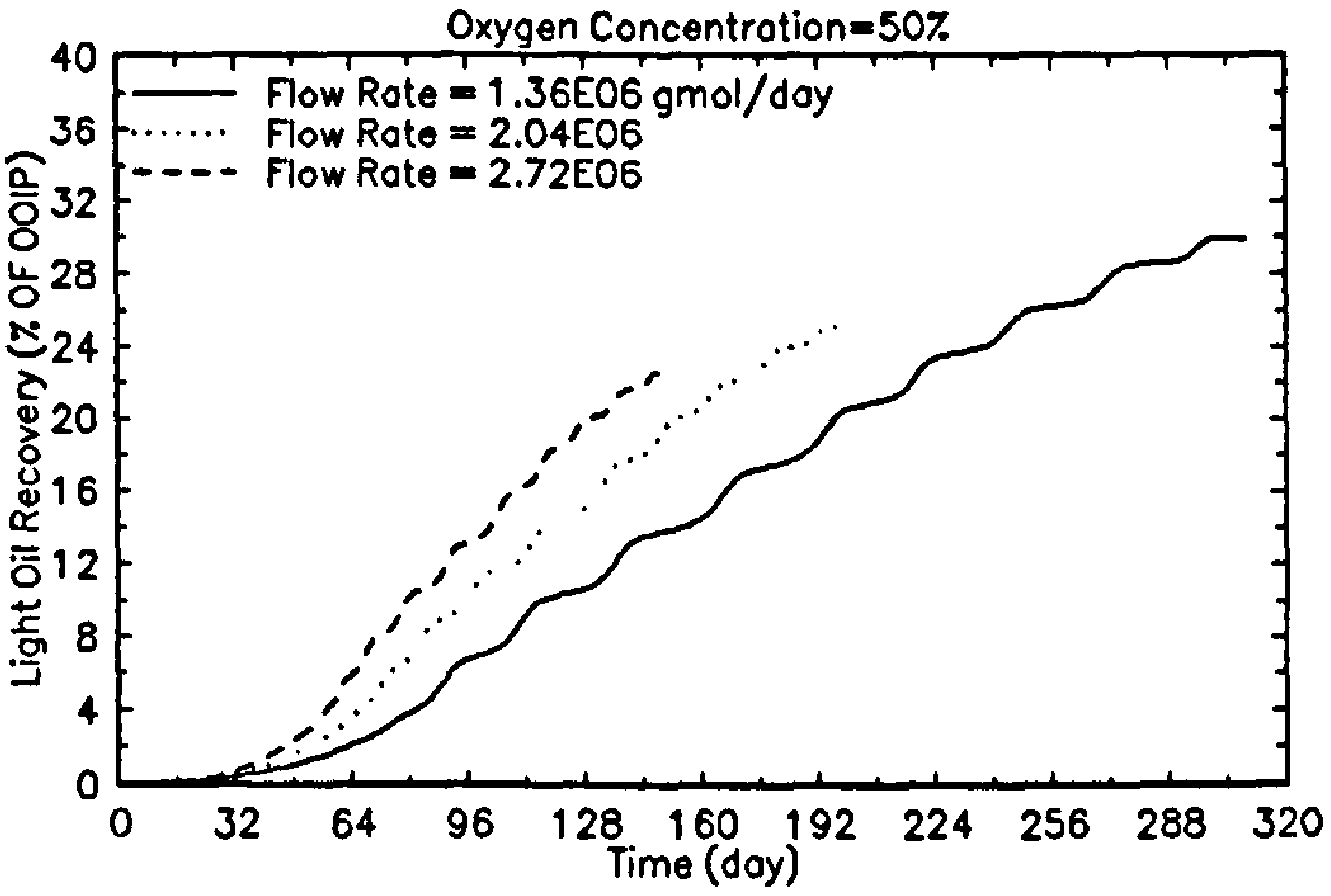


Figure 4.11 Light Oil Recovery Profile

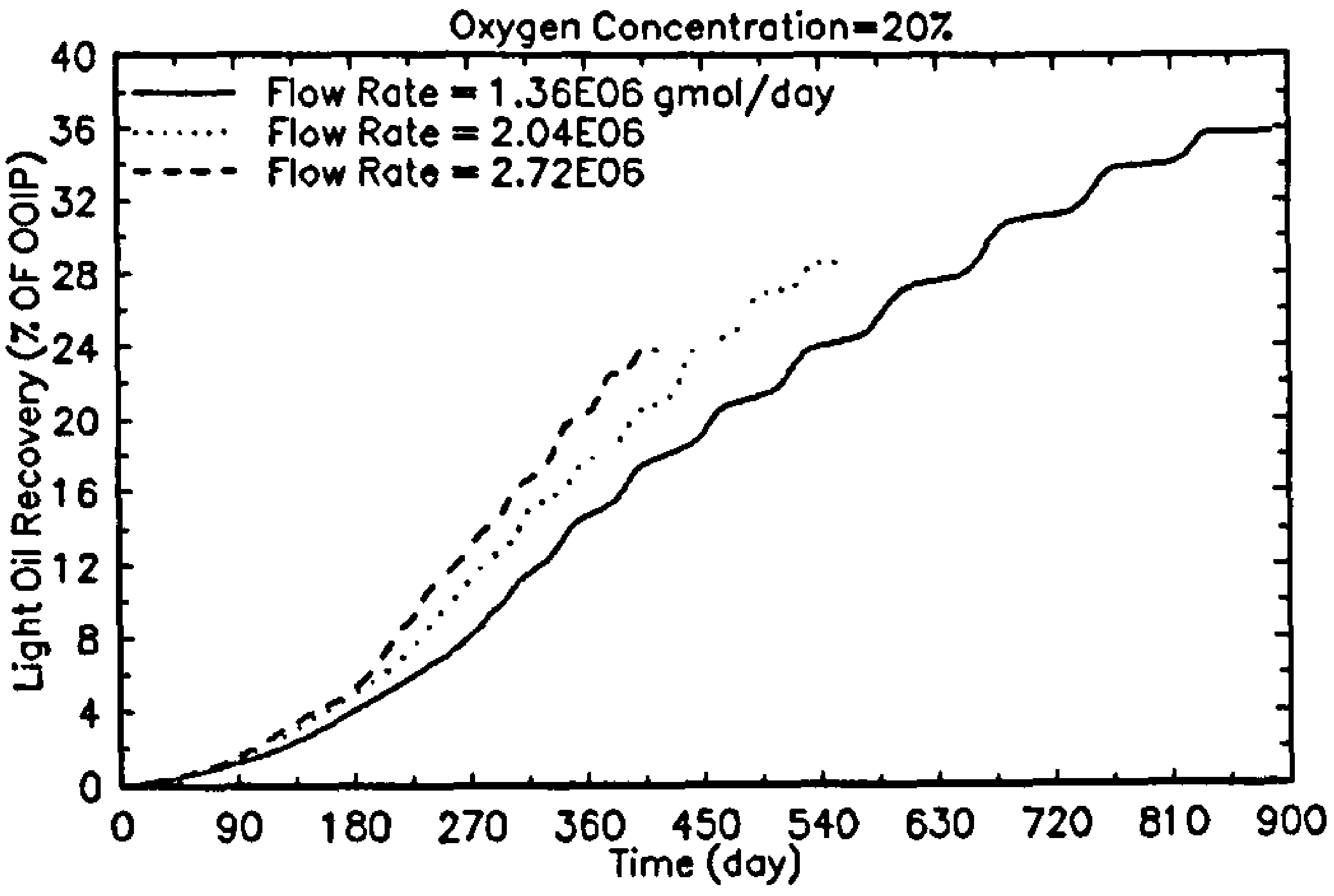


Figure 4.12 Light Oil Recovery Profile

4.2.2 The Effect Of The Oxygen Concentration

Tables 4.4, 4.5, and 4.6 show the effects of the changes in the oxygen concentration in the injection gas at three different total flow rates.

It can be seen that the production time increases significantly with decreasing oxygen concentration. This is due to the lower reaction rates resulting from the decrease in the oxygen partial pressure.

Figures 4.13, 4.14, and 4.15 and figures 4.16, 4.17, and 4.18 show the oil and water recovery profiles respectively. A significant decrease in both the oil and water produced can be seen at any specific time with decreasing oxygen content although the total amount of oil and water which can be recovered irrespective of the time did not change significantly.

In figures 4.19, 4.20, and 4.21 the light oil recovery is shown. It can be seen that the total amount of light oil produced regardless of time, increases with decreasing oxygen concentration of the injected gas. This is because of the lower light oil oxidation reaction rate resulting from the lower oxygen partial pressure.

Figures 4.22, 4.23, and 4.24 show a lower peak temperature with a lower oxygen concentration for different flow rates. This can be attributed to the overall decrease in the reaction rates resulting from the lower oxygen partial pressure. These figures also show a slower combustion front which is also a result of the lower rates of the oxidation reactions.

The pressure distribution of a typical grid block in the reservoir (grid block three) is shown in figures 4.25, 4.26, and 4.27. A lower pressure results from a lower oxygen concentration. This is because of the lower oil and water vapour pressure resulting from the lower temperature.

Tables 4.4, 4.5 and 4.6 also show how the heat loss varies with the oxygen concentration. The increase in heat loss with deceased oxygen concentration is a consequence of the longer production time. This also occurs because the heat loss term is presented as a percentage of the total heat of reaction which decreases with lower oxygen concentrations.

Table 4.4 The effect of the oxygen concentration

FLOW RATE = 1.36 10^5 gMOL/day			
VARIABLES	OXYGEN CONCENTRATION		
	100%	50%	20%
WATER RECOVERY (% OF OWIP)	117.4	117.6	122.6
TOTAL OIL RECOVERY (% OOIP)	95.3	95.1	96.2
LIGHT OIL RECOVERY (% OOIP)	26.3	30.0	35.8
AVERAGE PEAK TEMPERATURE (K)	622.0	604.4	578.6
MAXIMUM PRESSURE (kPa)	1242.0	833.6	696.5
TOTAL PRODUCTION TIME (day)	145.0	309.6	886.7
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	25.0	27.8	30.2

Table 4.5 The effect of the oxygen concentration

FLOW RATE = 2.04 10^5 gMOL/day			
VARIABLES	OXYGEN CONCENTRATION		
	100%	50%	20%
WATER RECOVERY (% OF OWIP)	117.2	117.5	122.1
TOTAL OIL RECOVERY (% OOIP)	93.6	93.0	93.8
LIGHT OIL RECOVERY (% OOIP)	22.5	25.1	28.6
AVERAGE PEAK TEMPERATURE (K)	620.0	594.6	568.0
MAXIMUM PRESSURE (kPa)	1369.3	1029.6	857.4
TOTAL PRODUCTION TIME (day)	95.1	198.2	553.0
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	22.3	24.8	27.1

Table 4.6 The effect of the oxygen concentration

FLOW RATE = 2.72 10^5 gMOL/day			
VARIABLES	OXYGEN CONCENTRATION		
	100%	50%	20%
WATER RECOVERY (% OF OWIP)	117.4	118.0	122.4
TOTAL OIL RECOVERY (% OOIP)	92.8	92.0	92.6
LIGHT OIL RECOVERY (% OOIP)	20.6	22.6	23.8
AVERAGE PEAK TEMPERATURE (K)	621.8	597.6	554.0
MAXIMUM PRESSURE (kPa)	1558.3	1177.9	1013.8
TOTAL PRODUCTION TIME (day)	72.0	149.0	414.5
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	20.6	23.0	25.0

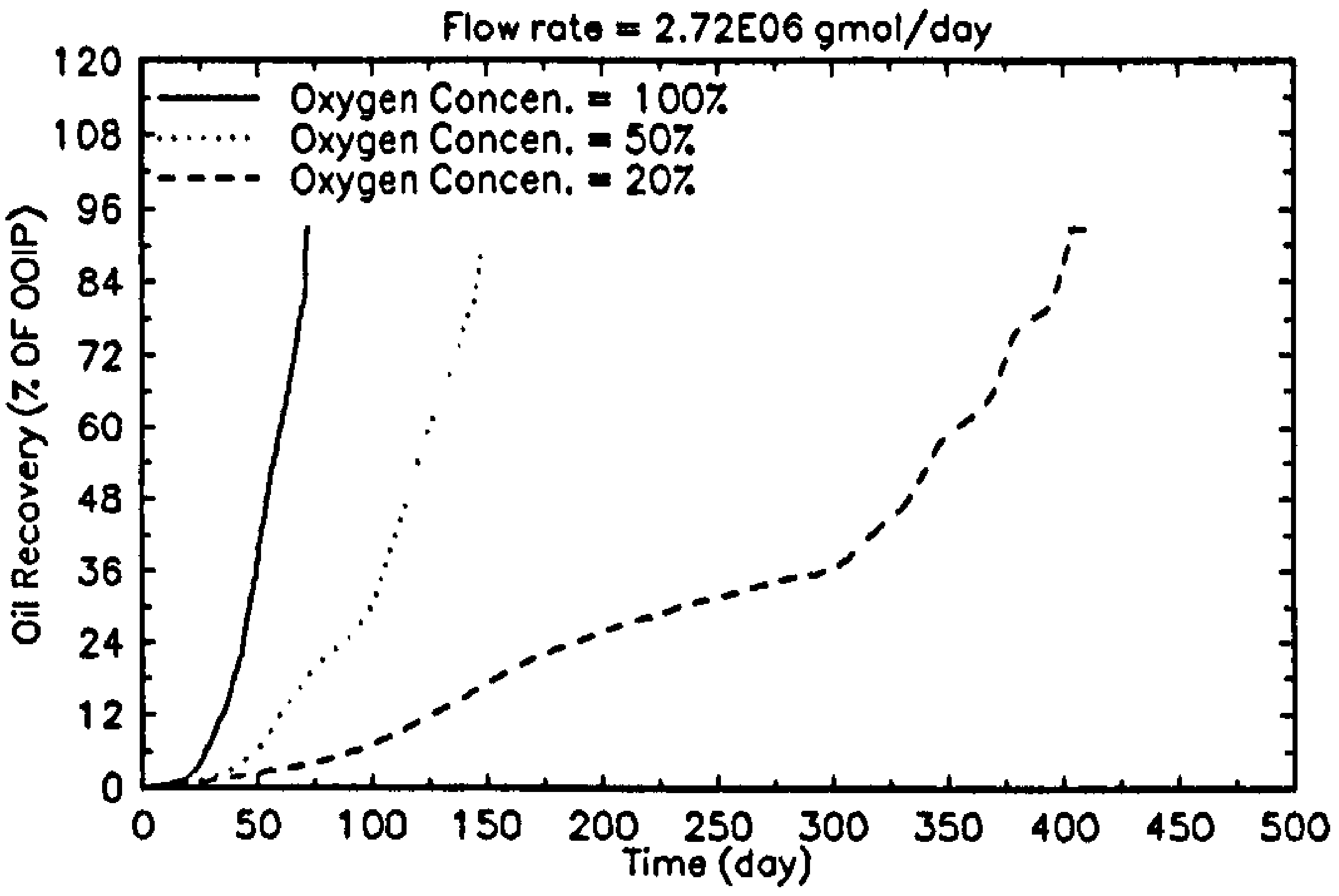


Figure 4.13 Oil Recovery Profile

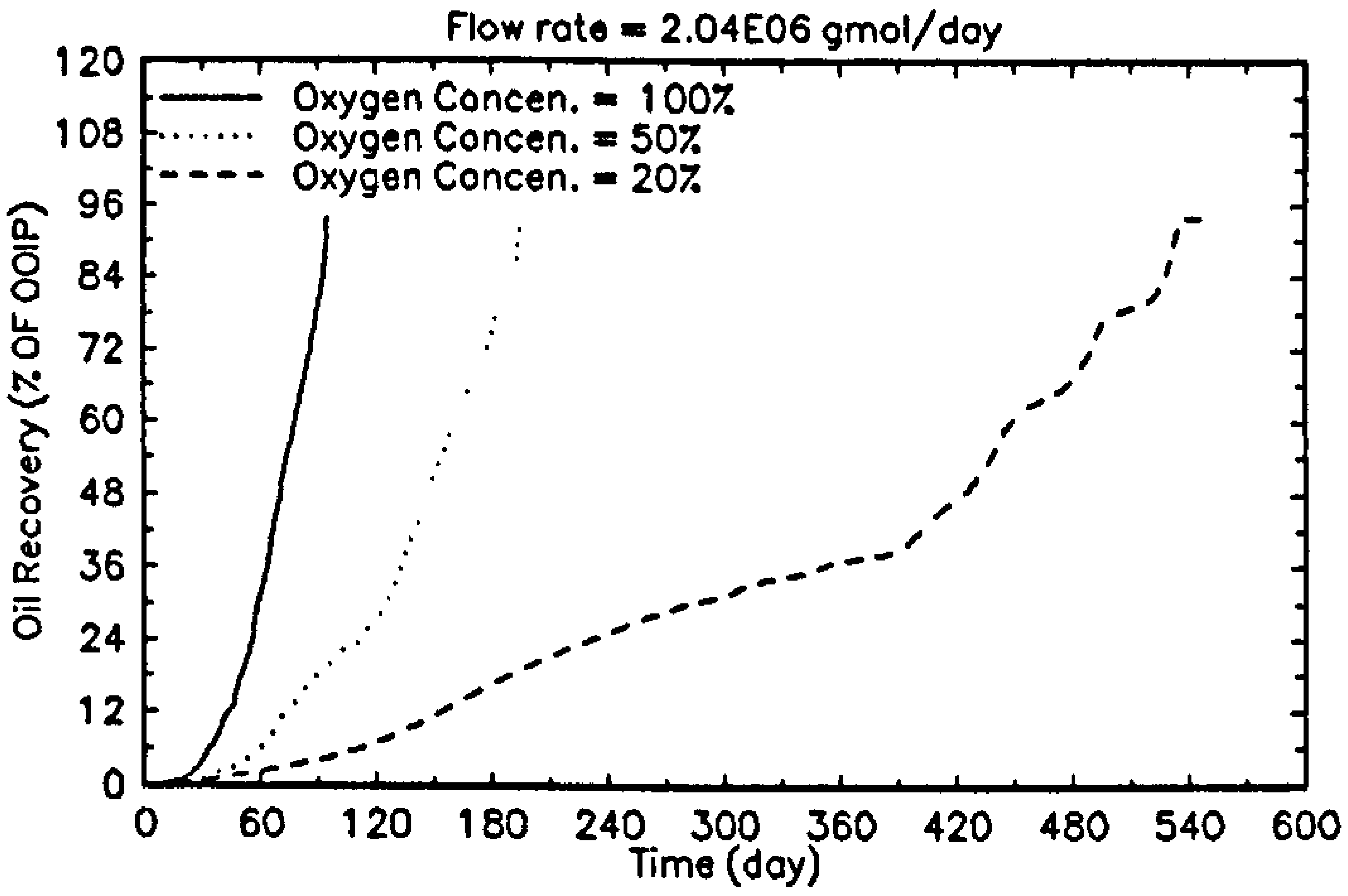


Figure 4.14 Oil Recovery Profile

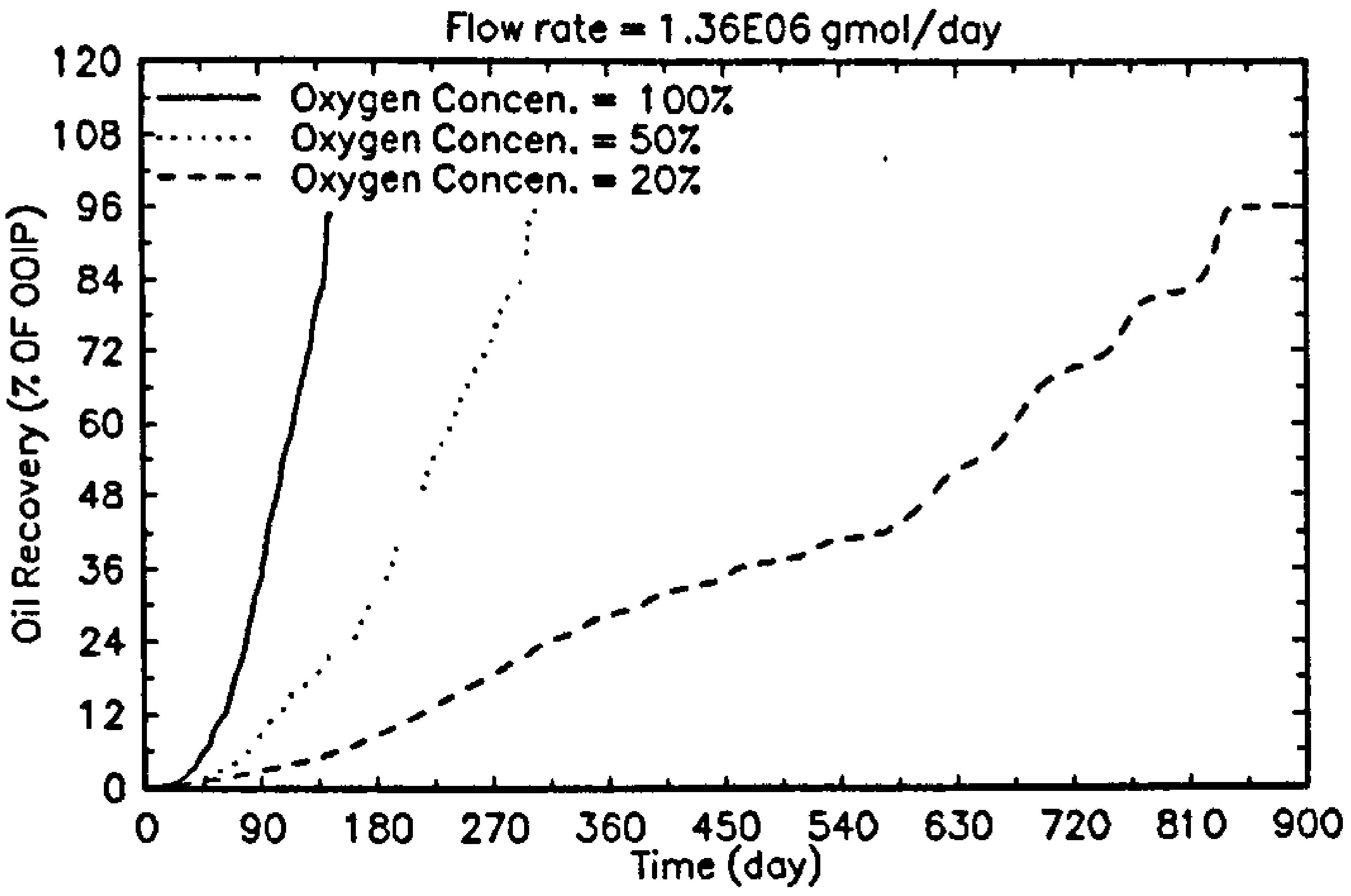


Figure 4.15 Oil Recovery Profile

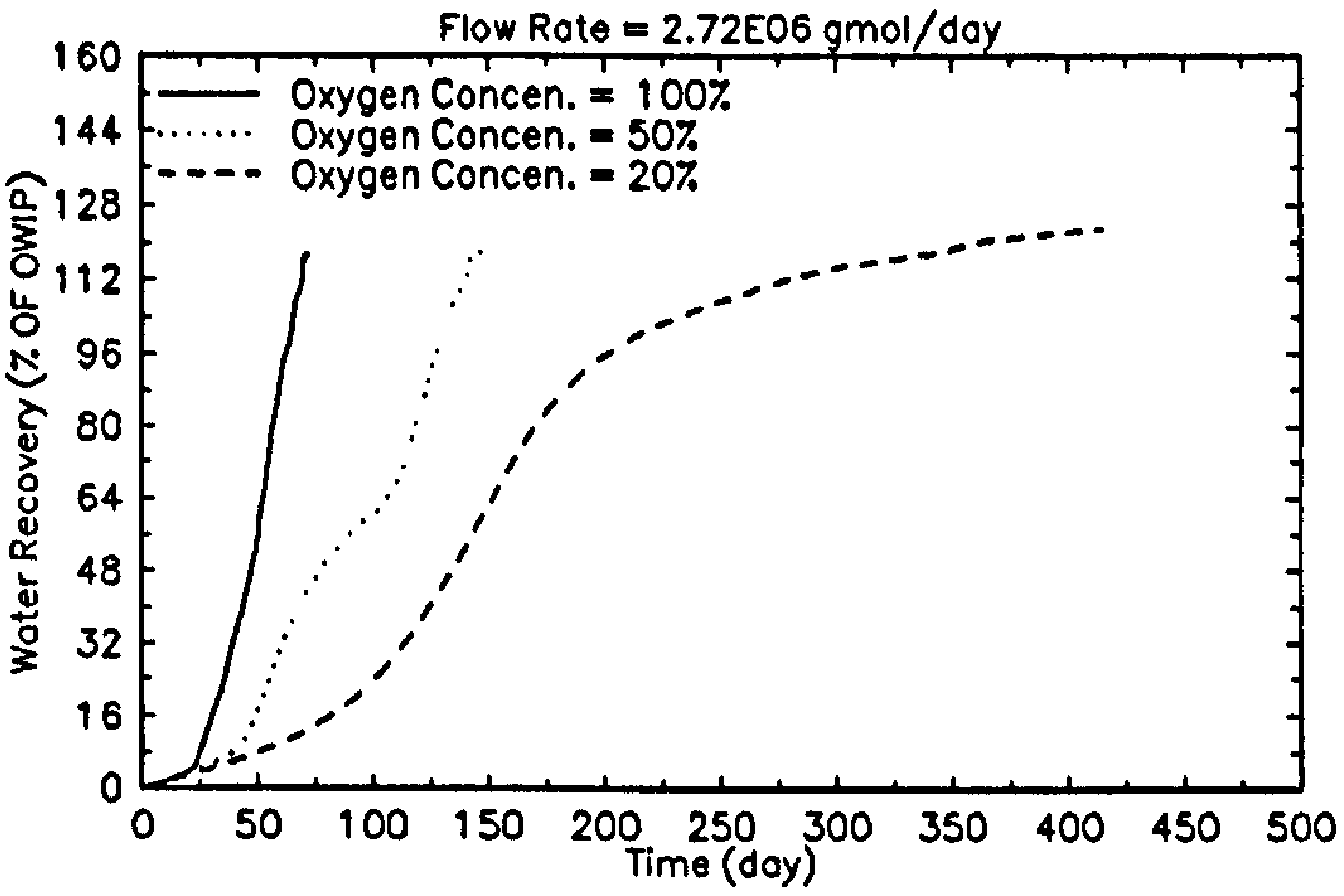


Figure 4.16 Water Recovery Profile

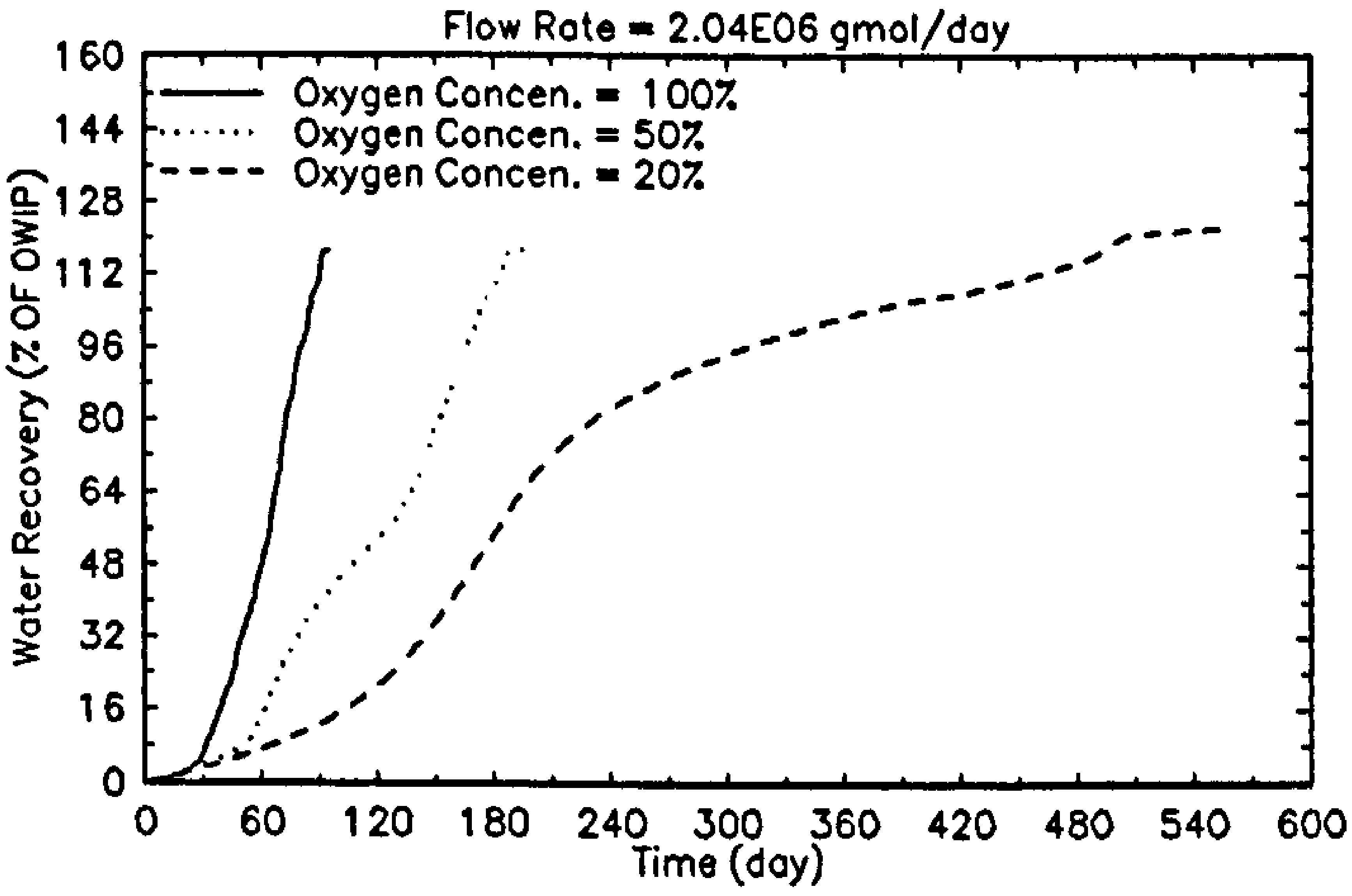


Figure 4.17 Water Recovery Profile

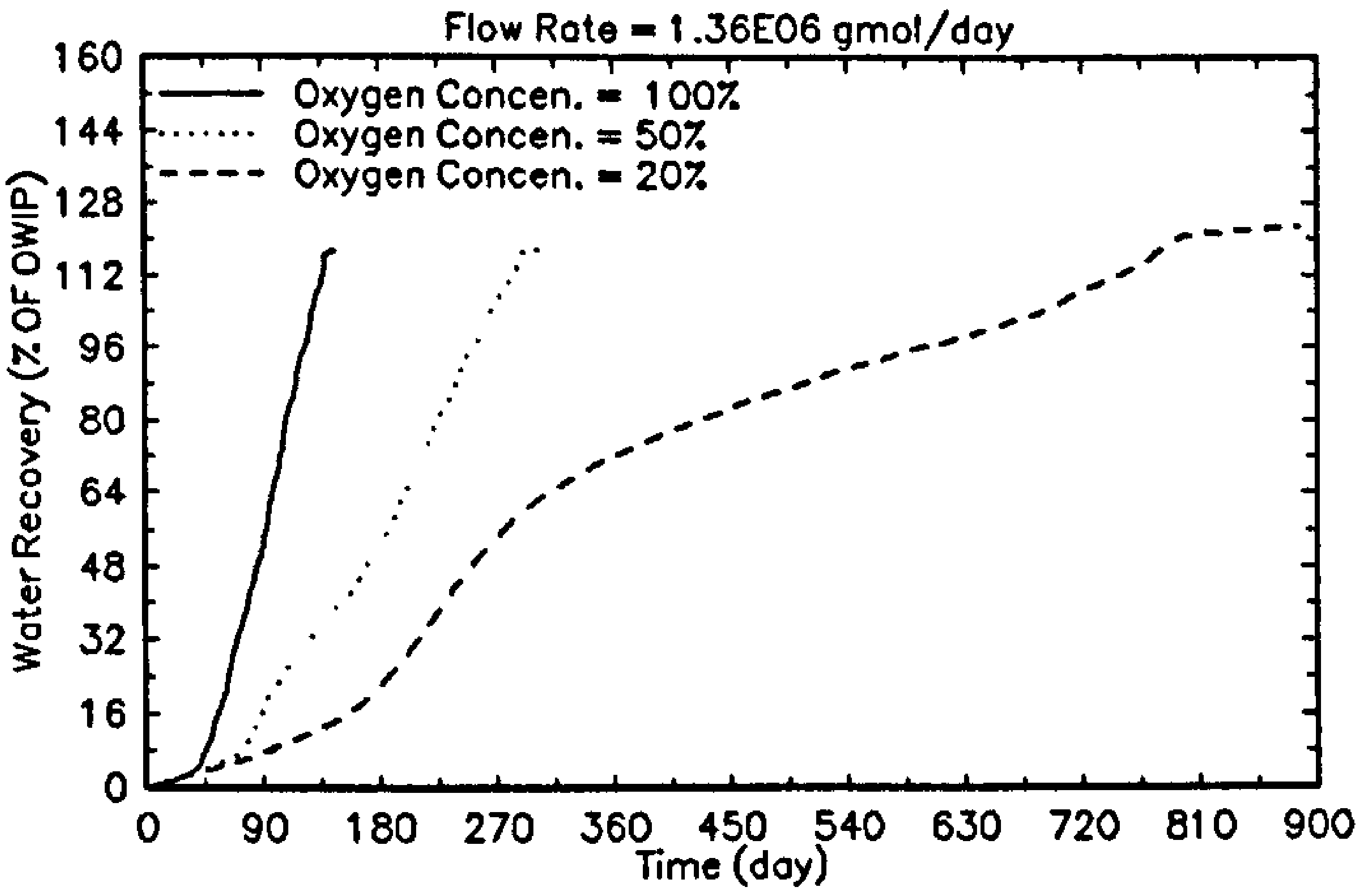


Figure 4.18 Water Recovery Profile

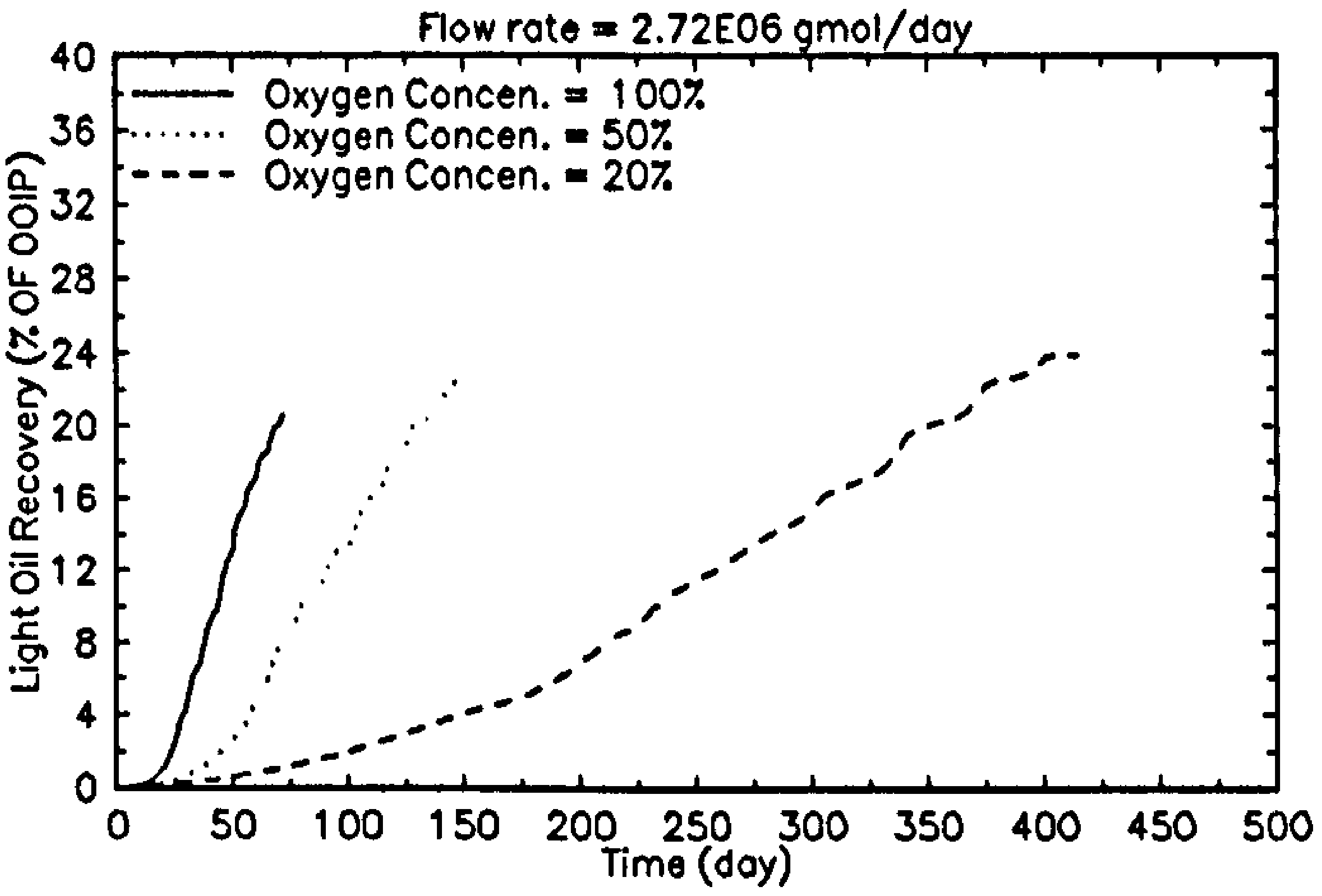


Figure 4.19 Light Oil Recovery Profile

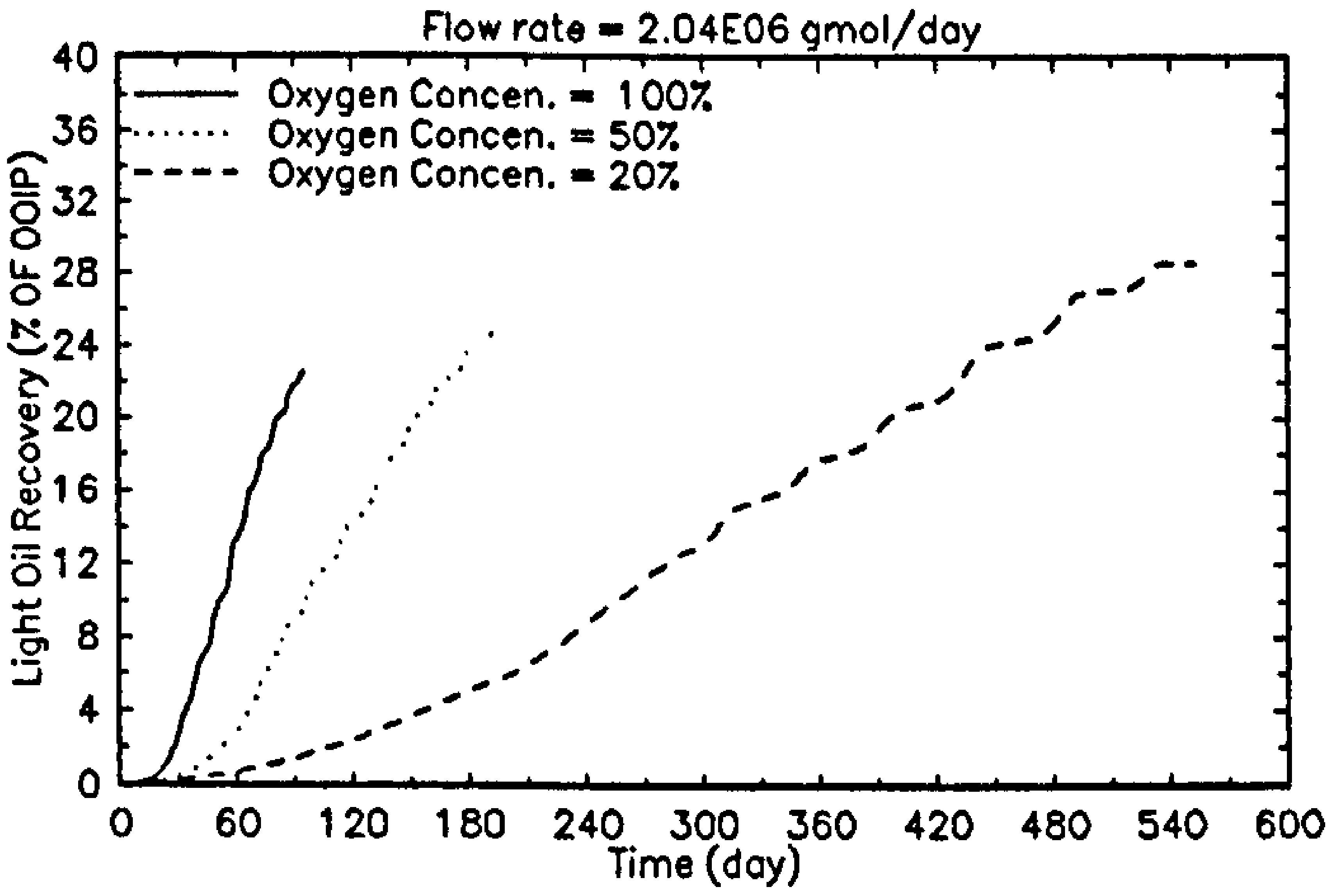


Figure 4.20 Light Oil Recovery Profile

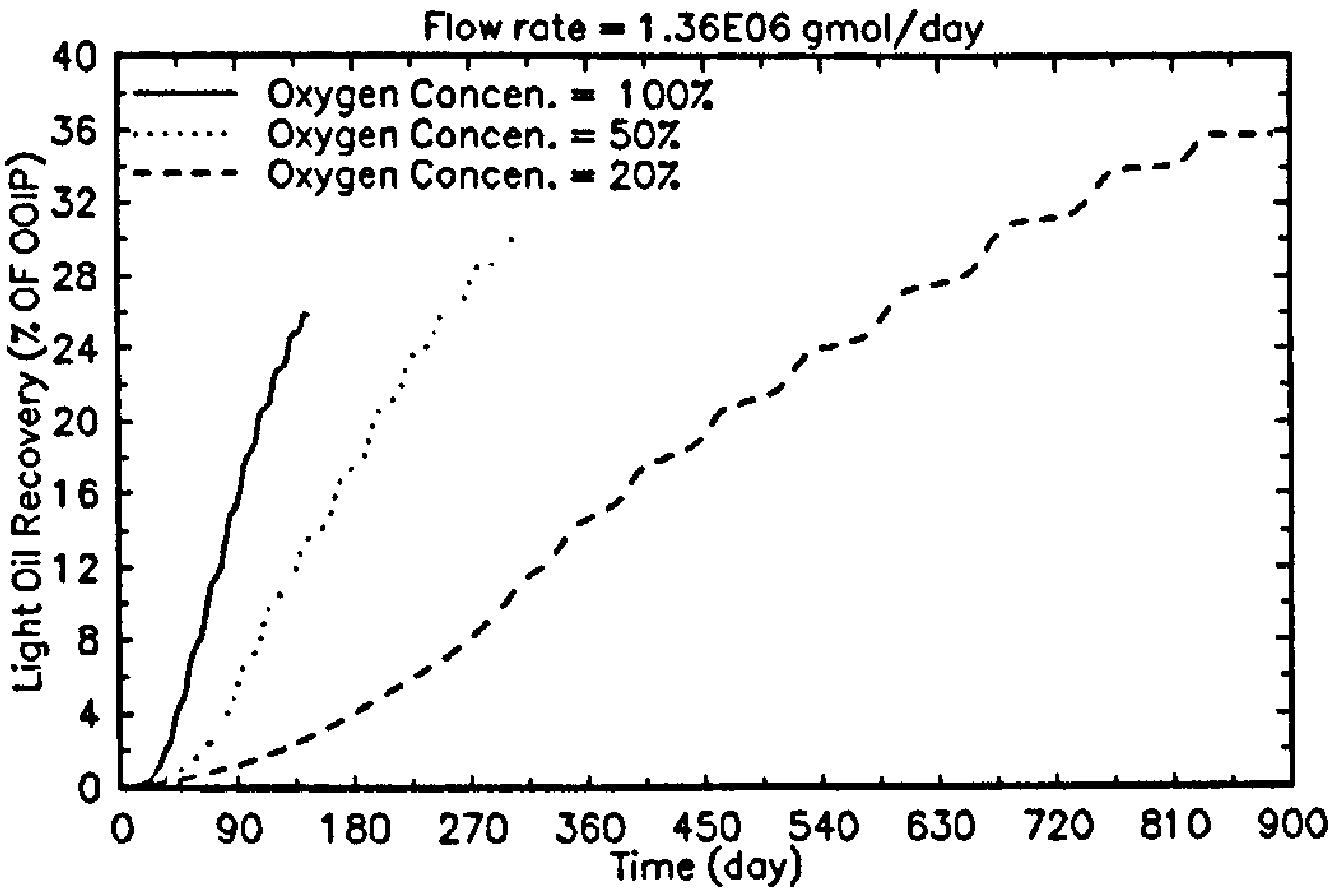


Figure 4.21 Light Oil Recovery Profile

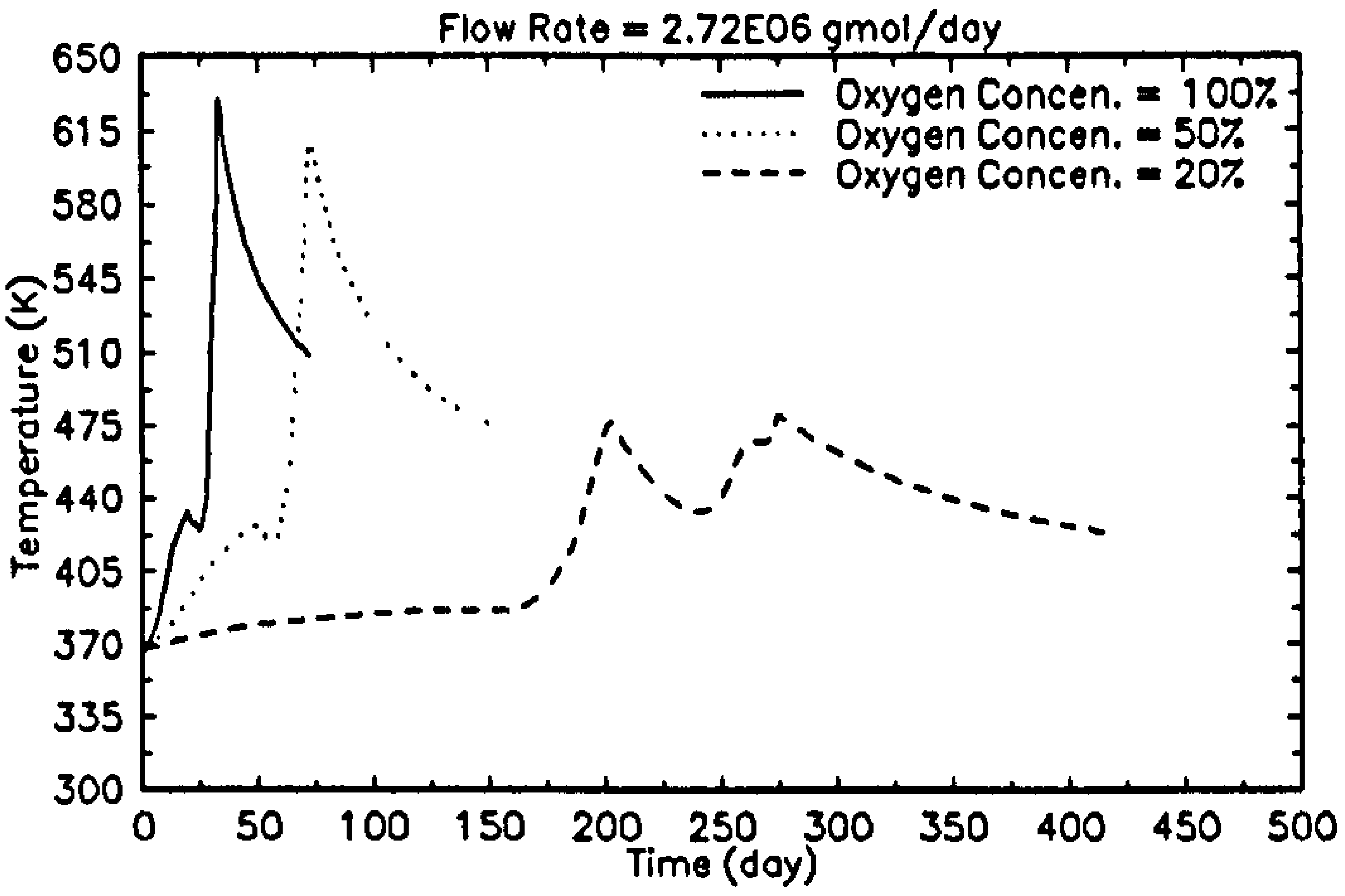


Figure 4.22 Temperature Profile Of Grid Block 3

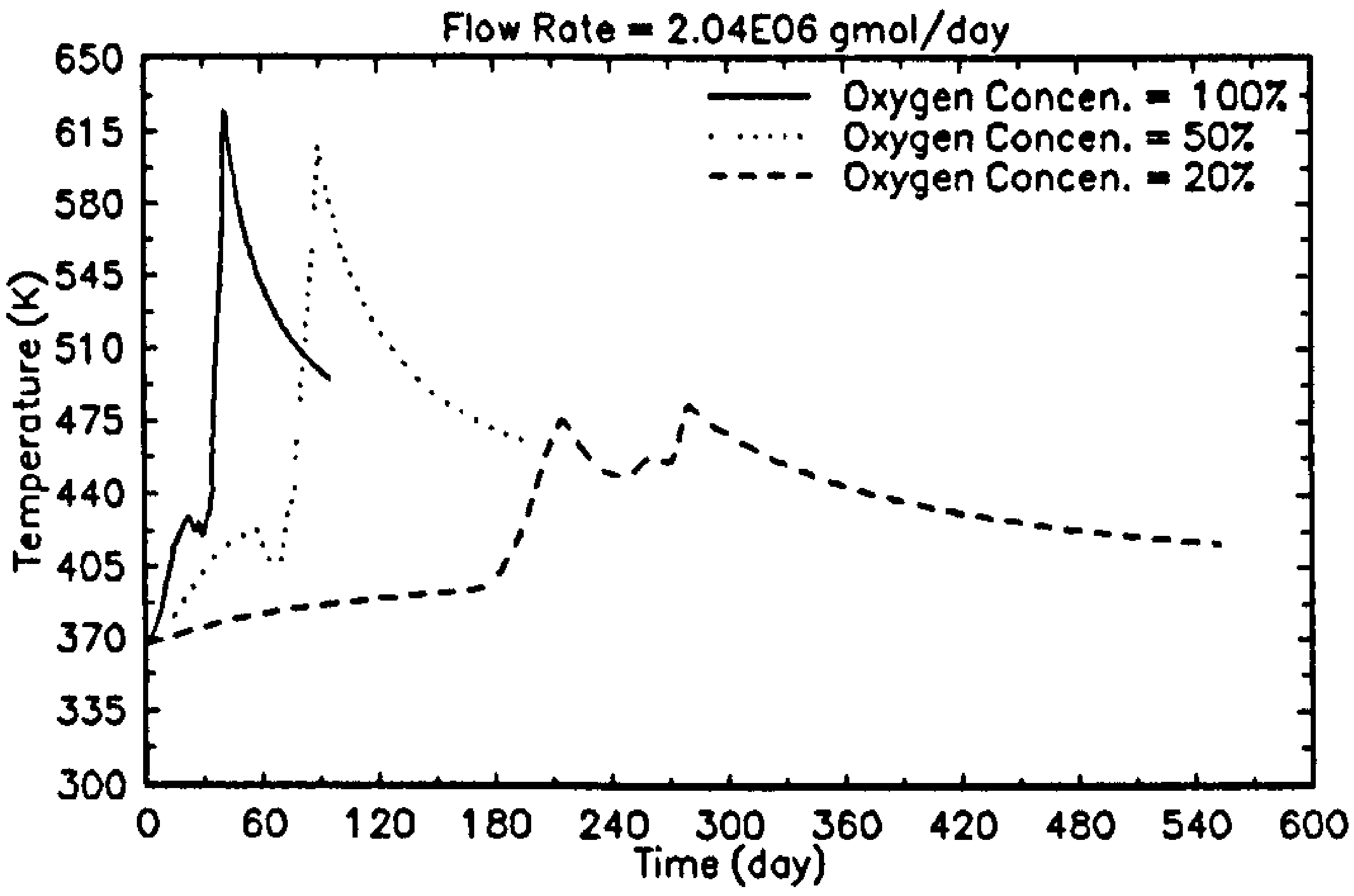


Figure 4.23 Temperature Profile Of Grid Block 3

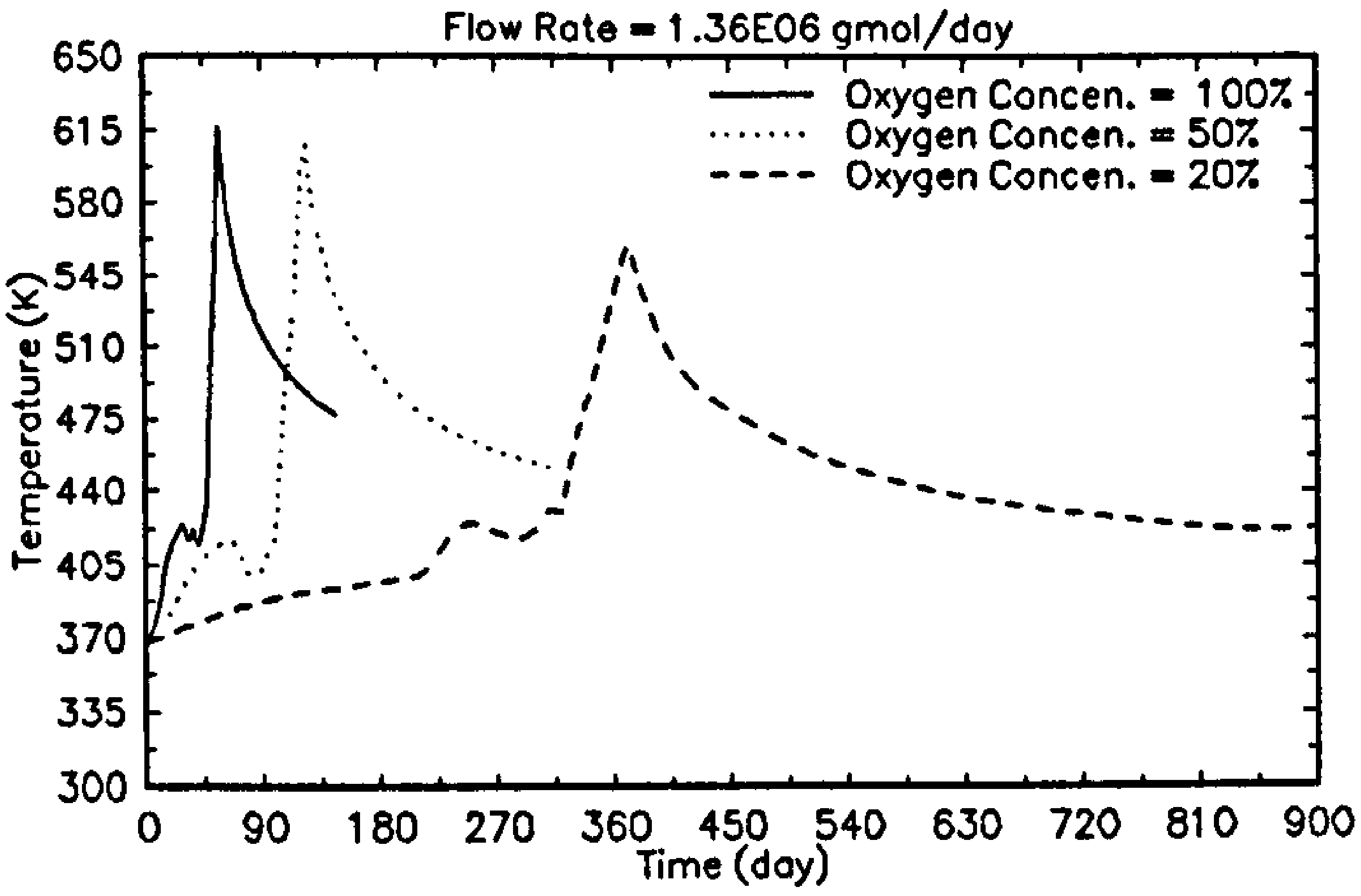


Figure 4.24 Temperature Profile Of Grid Block 3

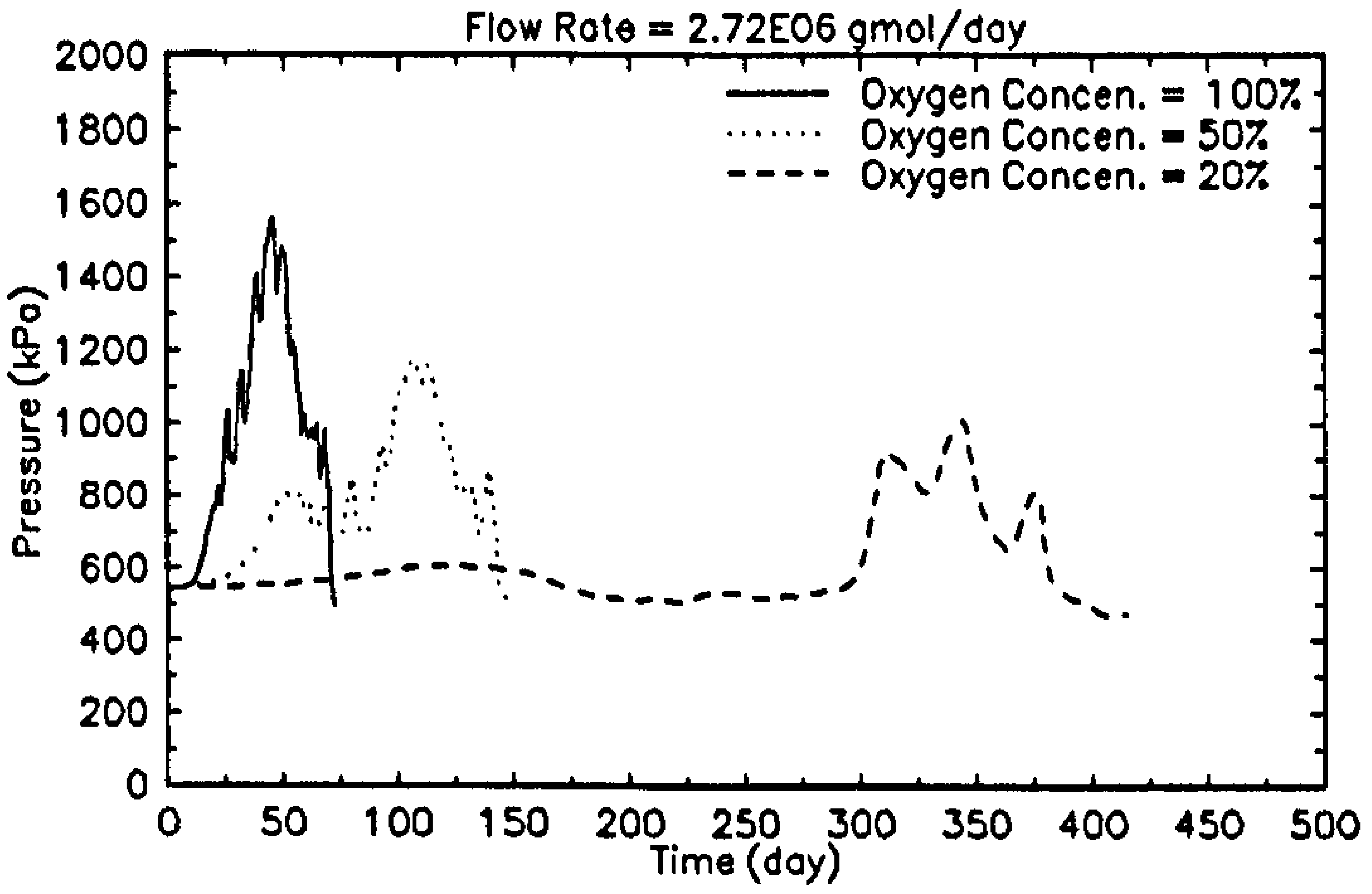


Figure 4.25 Pressure Profile Of Grid Block 3

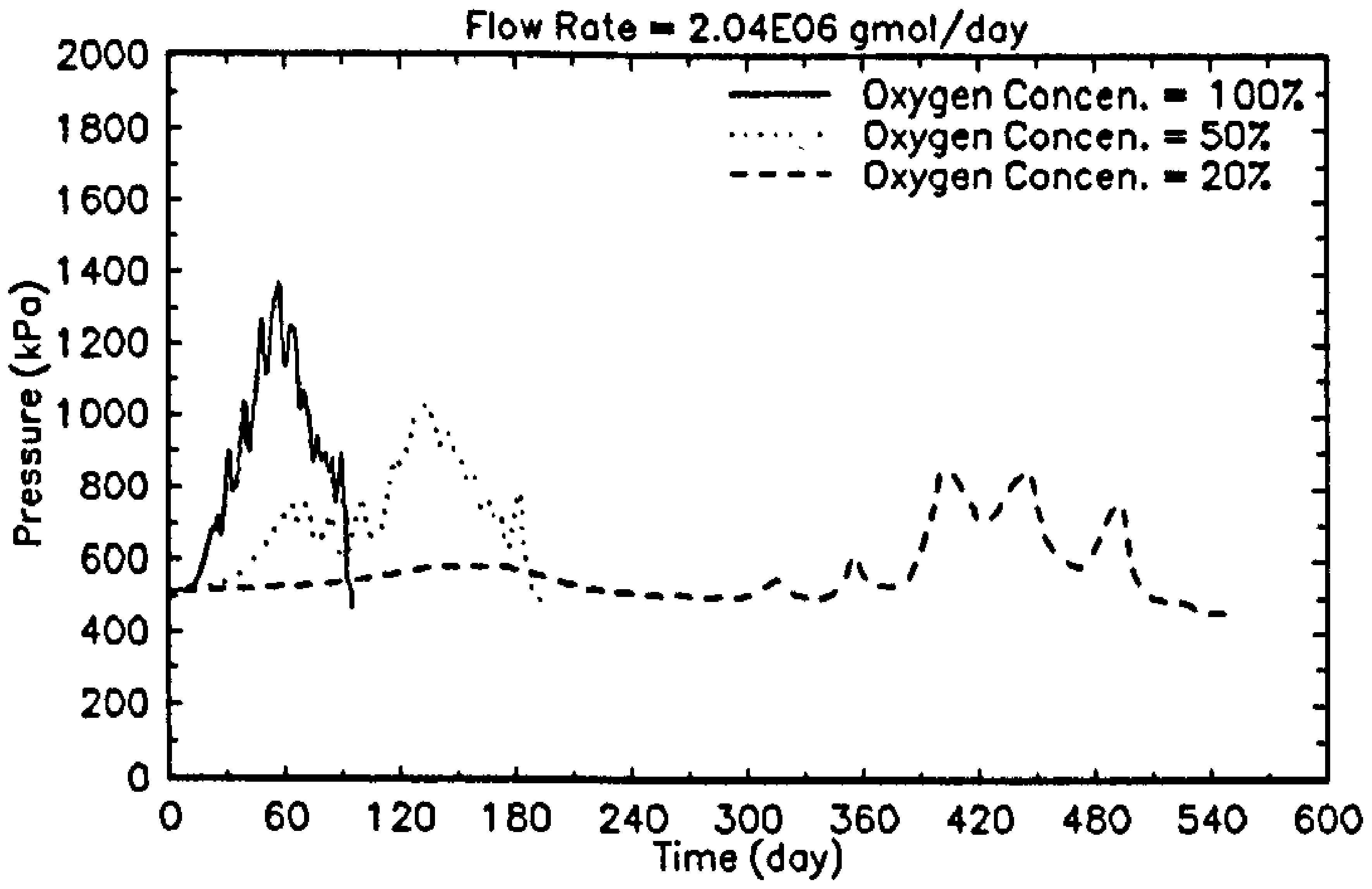


Figure 4.26 Pressure Profile Of Grid Block 3

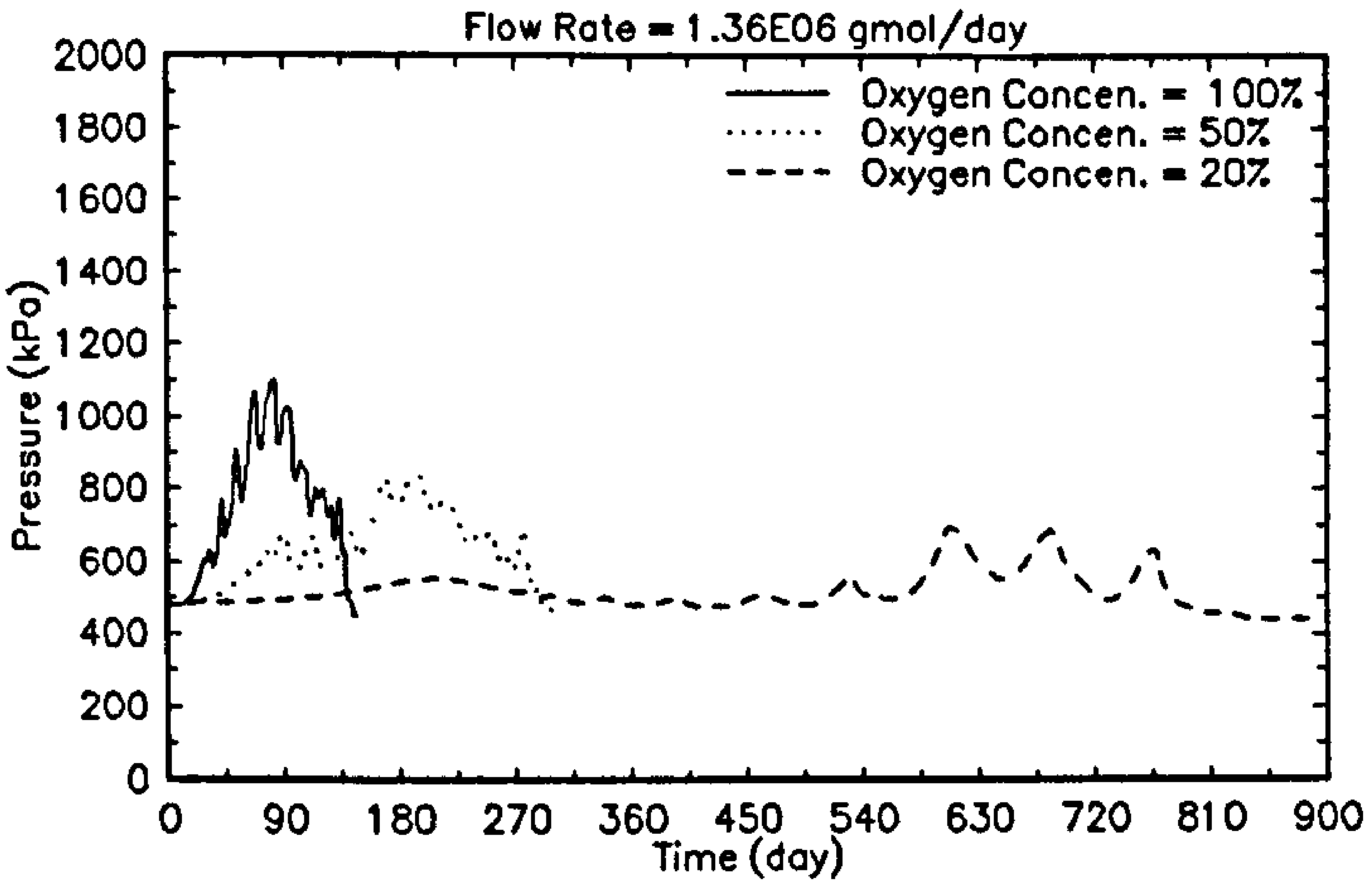


Figure 4.27 Pressure Profile Of Grid Block 3

4.3 Hydrodynamic Effects

4.3.1 The Effects Of Relative Permeability

4.3.1.1 Effect Of Oil Relative Permeability

Oil relative permeability is directly proportional to the oil phase mobility through Darcy's law. Therefore, it is expected to have a significant effect on the performance characteristics of in-situ combustion.

Oil relative permeability is calculated using the following equations;

$$K_{rog} = K_{rocw}[1 - (S_g - S_{gc})/(1 - S_{wc} - S_{org} - S_{gc})]^{Z_{og}} \tag{4.1}$$

$$K_{row} = K_{rocw}[1 - (S_w - S_{wc})/(1 - S_{orw} - S_{wc})]^{Z_{ow}} \tag{4.2}$$

$$K_{ro} = K_{rocw}[(K_{row}/K_{rocw} + K_{rw})(K_{rog}/K_{rocw} + K_{rg}) - K_{rg} - K_{rw}] \tag{4.3}$$

where K_{rg} is the gas relative permeability

K_{rw} is the water relative permeability

K_{ro} is the oil relative permeability

K_{rog} is oil-gas two phase relative permeability to oil

K_{row} is oil-water two phase relative permeability to oil

K_{rocw} is oil-water two-phase relative permeability to oil at connate water saturation

S_{wc} is connate water saturation

S_{gc} is critical gas saturation

S_{org} is residual oil saturation for oil-gas system

S_{orw} is residual oil saturation for oil-water system

Z_{og} and Z_{ow} are empirical constants

Z_{ow} and Z_{og} are changed to study the effect of oil relative permeability. These constants are chosen due to their direct relationship with the oil relative permeability. The bracketed terms in equation 4.1 and 4.2 are always fractional which means that the oil relative permeability decreases when Z_{ow} and Z_{og} are increased. Although the changes in Z_{ow} and Z_{og} lead to a slight change in the relative permeability, they can still provide a general insight into the effect of the oil relative permeability on the performance characteristics of in-situ combustion. Tables 4.7 and 4.8 show the effect of changing Z_{ow} and Z_{og} respectively.

It can be seen from table 4.7 that the time needed to achieve maximum recovery is increased when Z_{ow} is decreased. This is due to the decrease in the oil mobility.

Table 4.7 also shows that the changes in Z_{ow} show insignificant changes on the total amount of oil and water produced; the slight increase in the light oil produced when Z_{ow} was increased is a result of the lower oil mobility. Also, with the slower movement of the oil the fuel availability increases which leads to an increase in the average peak temperature.

Figure 4.28 shows the pressure profile of a typical grid block in the reservoir (grid block three). The increase shown in the pressure as Z_{ow} increases is due to the increase in the oil vapor pressure with the slower oil mobility.

Table 4.7 shows that the changes in Z_{ow} did not lead to any significant changes in the amount of heat losses to the surrounding strata.

Table 4.8 shows that similar effects to that of Z_{ow} have been encountered with the changes in Z_{og}.

Table 4.7 The effect of oil relative permeability

VARIABLES	Z_{ow}		
	1.00	3.00*	5.00
WATER RECOVERY (% OF OWIP)	117.1	117.4	118.0
TOTAL OIL RECOVERY (% OOIP)	94.0	95.3	96.4
LIGHT OIL RECOVERY (% OOIP)	24.7	26.3	29.1
AVERAGE PEAK TEMPERATURE (K)	626.3	622.0	650.1
MAXIMUM PRESSURE (kPa)	985.5	1242.0	1409.3
TOTAL PRODUCTION TIME (day)	142.2	145.1	154.4
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	26.7	25.0	25.8

Table 4.8 The effect of oil relative permeability

VARIABLES	Z_{og}		
	1.00	3.00*	5.00
WATER RECOVERY (% OF OWIP)	116.5	117.4	117.3
TOTAL OIL RECOVERY (% OOIP)	92.6	95.3	96.2
LIGHT OIL RECOVERY (% OOIP)	22.4	26.3	28.3
AVERAGE PEAK TEMPERATURE (K)	594.3	622.0	640.5
MAXIMUM PRESSURE (kPa)	1180.2	1242.0	1232.9
TOTAL PRODUCTION TIME (day)	134.5	145.1	151.3
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	24.4	25.0	25.4

* Original case study

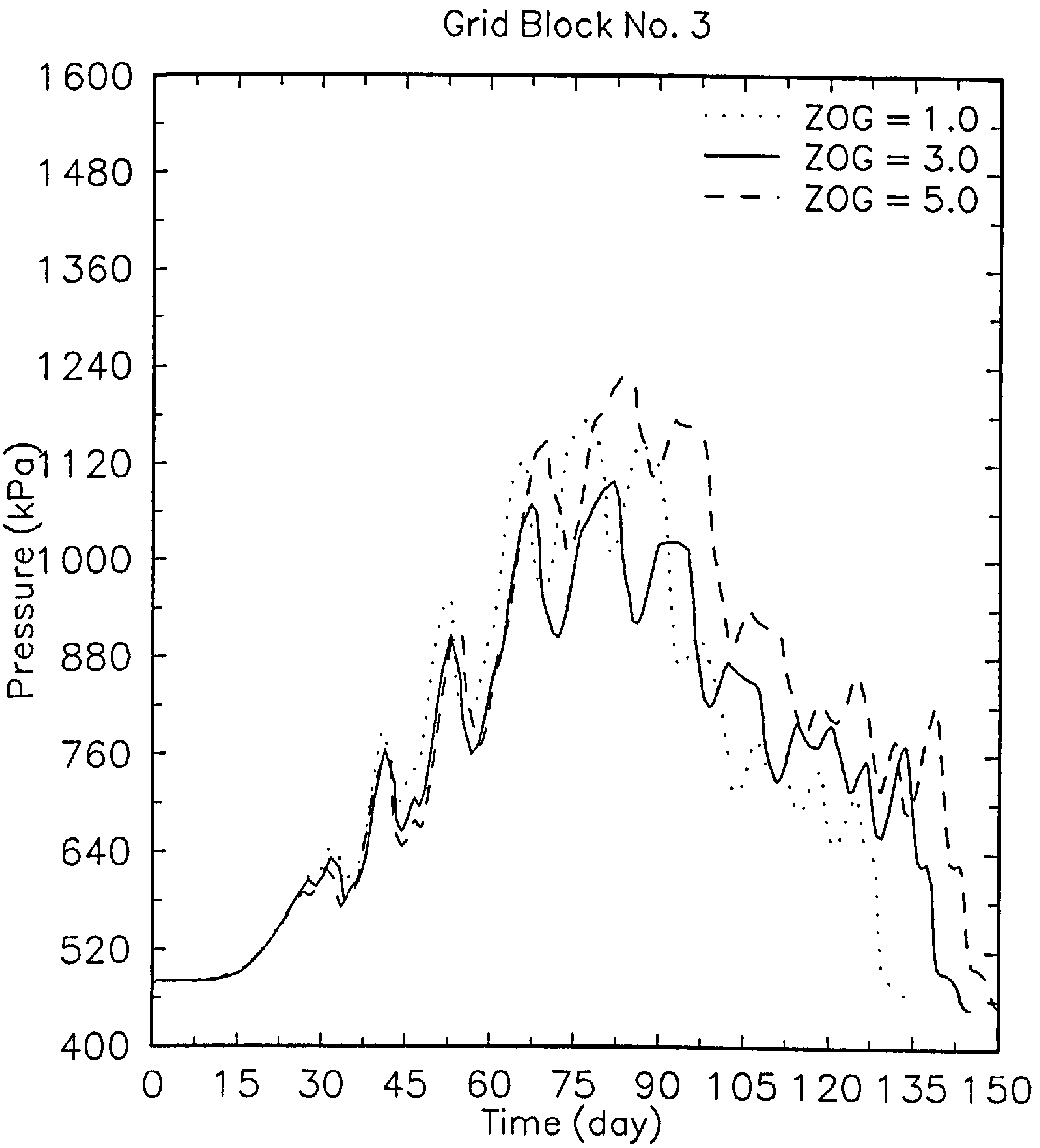


Figure 4.28 Pressure Profile Of Grid Block 3

4.3.12 The Effect Of Gas Relative Permeability

The gas phase mobility is directly related to the gas relative permeability through Darcy's law. Also, gas relative permeability is related to oil relative permeability as shown in equation 4.3.

Gas relative permeability is calculated using the following equation;

$$K_{rg} = K_{rgro}[(S_g - S_{gc})/(1 - S_{wc} - S_{org} - S_{gc})]^{Z_g} \qquad (4.4)$$

where K_{rg} is the gas relative permeability

K_{rgro} is oil-gas two phase relative permeability to gas at residual oil saturation

S_{wc} is connate water saturation

S_{gc} is critical gas saturation

S_{org} is residual oil saturation for oil-gas system

Z_g is empirical constant

Z_g is changed to study the effect of the gas phase relative permeability. Equation 4.4 shows that the gas phase relative permeability decreases when Z_g is increased. Table 4.9 shows the effect of the changes in Z_g.

It can be seen from table 4.9 that when Z_g is increased, the pressure increases due to the lower gas mobility (more gas buildup). This increase in the pressure is also shown in figure 4.29 which presents the pressure profile of a typical grid block from the reservoir (grid block three).

The increase in the pressure leads to higher oxidation rates and consequently lower production times although the decrease in the gas relative permeability does not lead to a significant change in the total amount of oil and water produced and a slight decrease in the amount of light oil produced due to the higher light oil oxidation rate.

Table 4.9 also shows that the increase in Z_g did not have any significant effect on the heat losses to the surrounding strata. The oscillating average peak temperature is due to the fact that the temperature is effected by many conflicting variables in this case such as higher oil mobility and higher oxidation reaction rates.

Table 4.9 The effects of the gas relative permeability

VARIABLES	Z_g		
	1.00*	2.00	3.00
WATER RECOVERY (% OF OWIP)	117.4	112.4	111.2
TOTAL OIL RECOVERY (% OOIP)	95.3	95.0	94.4
LIGHT OIL RECOVERY (% OOIP)	26.3	26.7	24.0
AVERAGE PEAK TEMPERATURE (K)	622.0	639.2	624.3
MAXIMUM PRESSURE (kPa)	1242.0	1767.6	2146.8
TOTAL PRODUCTION TIME (day)	145.1	141.8	127.5
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	25.0	26.6	26.4

* Original case study

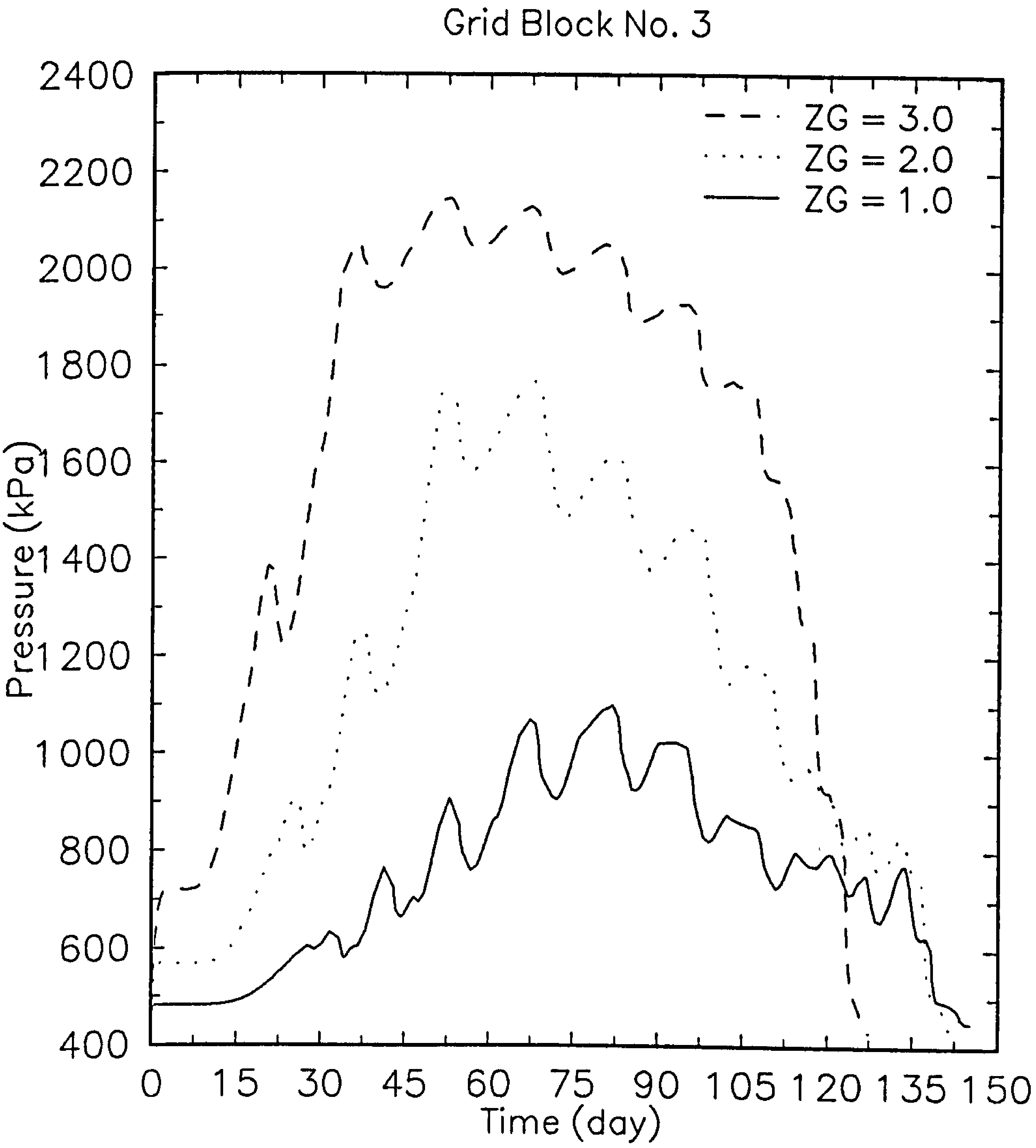


Figure 4.29 Pressure Profile Of Grid Block 3

4.3.13 The Effect Of Water Relative Permeability

Water relative permeability is directly proportional to the water phase mobility according to Darcy's law. Also, like the gas phase relative permeability, it is directly related to the oil relative permeability as shown in equation 4.3.

Water relative permeability is calculated using the following equation;

$$K_{rw} = K_{rwro}[(S_w - S_{wc})/(1 - S_{wc} - S_{orw})]^{Z_w} \qquad (4.5)$$

where K_{rw} is the water relative permeability

K_{rwro} is oil-water two-phase relative permeability to oil at residual oil saturation

S_{wc} is connate water saturation

S_{orw} is residual oil saturation for oil-water system

Z_w is empirical constant

Z_w is changed in order to study the effect of water relative permeability. Equation 4.5 shows that when Z_w increases, the water relative permeability decreases. Table 4.10 shows the effects of the changes in Z_w.

Inspection of the results shows that the changes in Z_w did not have any significant effect on the total amount of oil, water, light oil produced, average peak temperature, and the heat losses to the surrounding strata.

The increase in the pressure with the increase in Z_w (decrease in the water relative permeability) is due to the lower water mobility and higher vapor pressure. This increase in the pressure is also shown in figure 4.30 which presents the pressure profile of a typical grid block a typical grid block from the reservoir (grid block three).

Equation 2.50 shows that the decrease in the water relative permeability leads to a decrease in the oil relative permeability and consequently lower oil mobility which led to the longer production time.

Table 4.10 The effects of the water relative permeability

VARIABLES	Z_w		
	1.00	3.00^*	4.00
WATER RECOVERY (% OF OWIP)	116.4	117.4	116.5
TOTAL OIL RECOVERY (% OOIP)	96.4	95.3	93.4
LIGHT OIL RECOVERY (% OOIP)	25.5	26.3	26.9
AVERAGE PEAK TEMPERATURE (K)	624.7	622.0	624.1
MAXIMUM PRESSURE (kPa)	925.7	1242.0	1475.0
TOTAL PRODUCTION TIME (day)	142.7	145.1	149.3
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	25.1	25.0	25.0

* Original case study

4.3.2 The Effect Of Absolute Permeability

The mobility of gas, water, and oil is directly proportional to absolute permeability according to Darcy's law.

Table 4.11 shows the effect of the changes in the absolute permeability. Inspection of the results shows that the changes in the absolute permeability do not lead to any significant changes on the amount of oil and water produced. However, the total production time has decreased with the increase in the absolute permeability due to the increase in the oil mobility. This increase in the oil mobility means a reduction in the amount of the heavy oil available for the cracking reaction which means less fuel produced and consequently, the average peak temperature decreased. This decrease in the temperature is also shown in figure 4.31 which presents the temperature profile of a typical grid block from the reservoir (grid block three). Also the decrease in the heavy oil cracking reaction rate meant a decrease in the amount of light oil produced, which is one of products from this reaction, as shown in figure 4.32.

The slight decrease in the heat loss with increase in absolute permeability is a result of the low overall temperature of the formation. Similarly, the lower temperatures lead to lower oil and water vapor pressures which lead to a lower overall pressure. Figure 4.33 shows the pressure profile of a typical grid block from the reservoir (grid block three).

Table 4.11 The effects of the absolute permeability

VARIABLES	ABSOLUTE PERMEABILITY (m^2)		
	2.10^{-12}	4.10^{-12} *	8.10^{-12}
WATER RECOVERY (% OF OWIP)	117.0	117.4	117.5
TOTAL OIL RECOVERY (% OOIP)	96.6	95.3	93.7
LIGHT OIL RECOVERY (% OOIP)	31.8	26.3	22.2
AVERAGE PEAK TEMPERATURE (K)	674.5	622.0	591.3
MAXIMUM PRESSURE (kPa)	1916.9	1242.0	822.7
TOTAL PRODUCTION TIME (day)	158.7	145.1	137.9
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	27.0	25.0	23.4

* Original case study

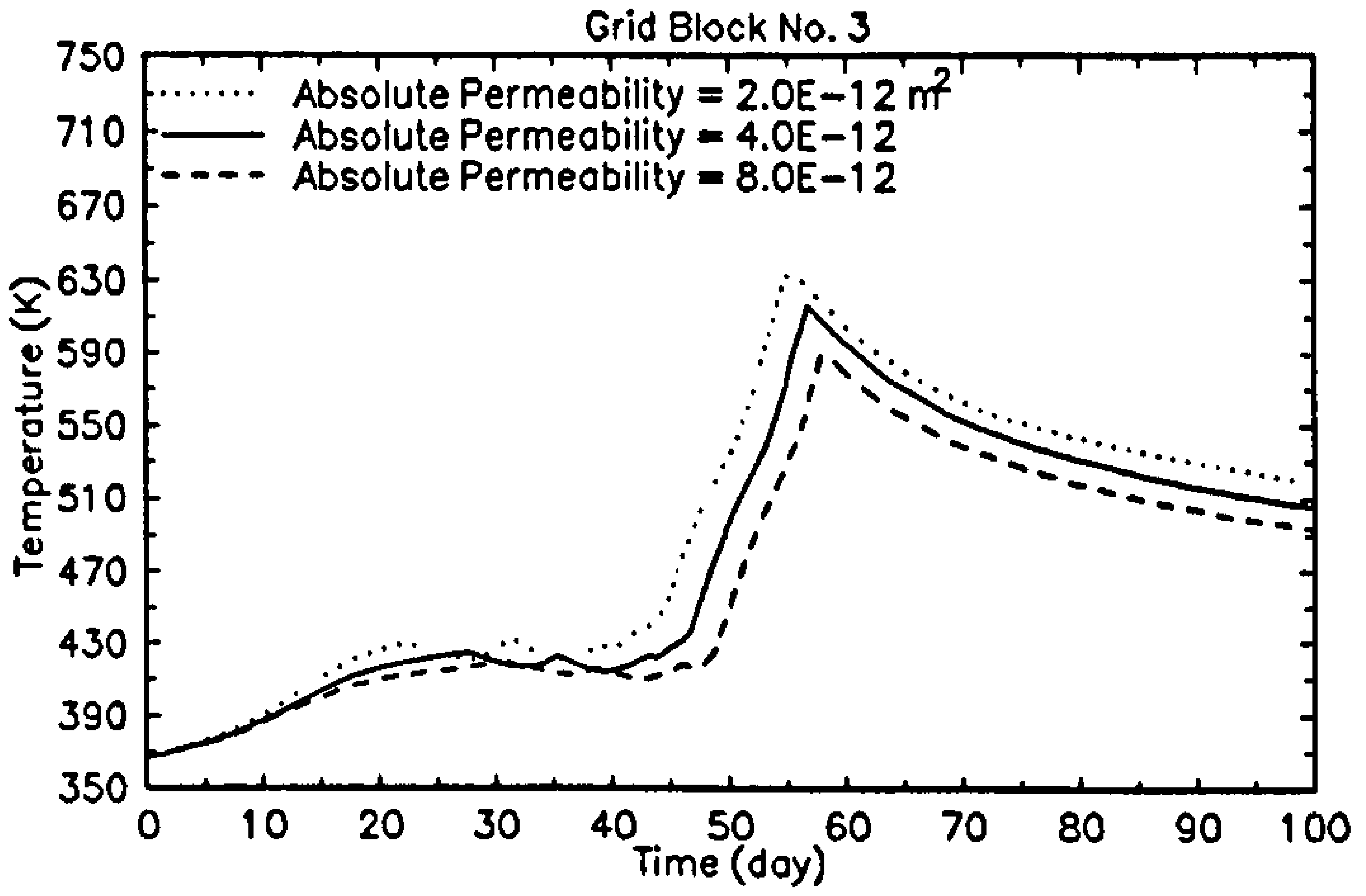


Figure 4.31 Temperature Profile Of Grid Block 3

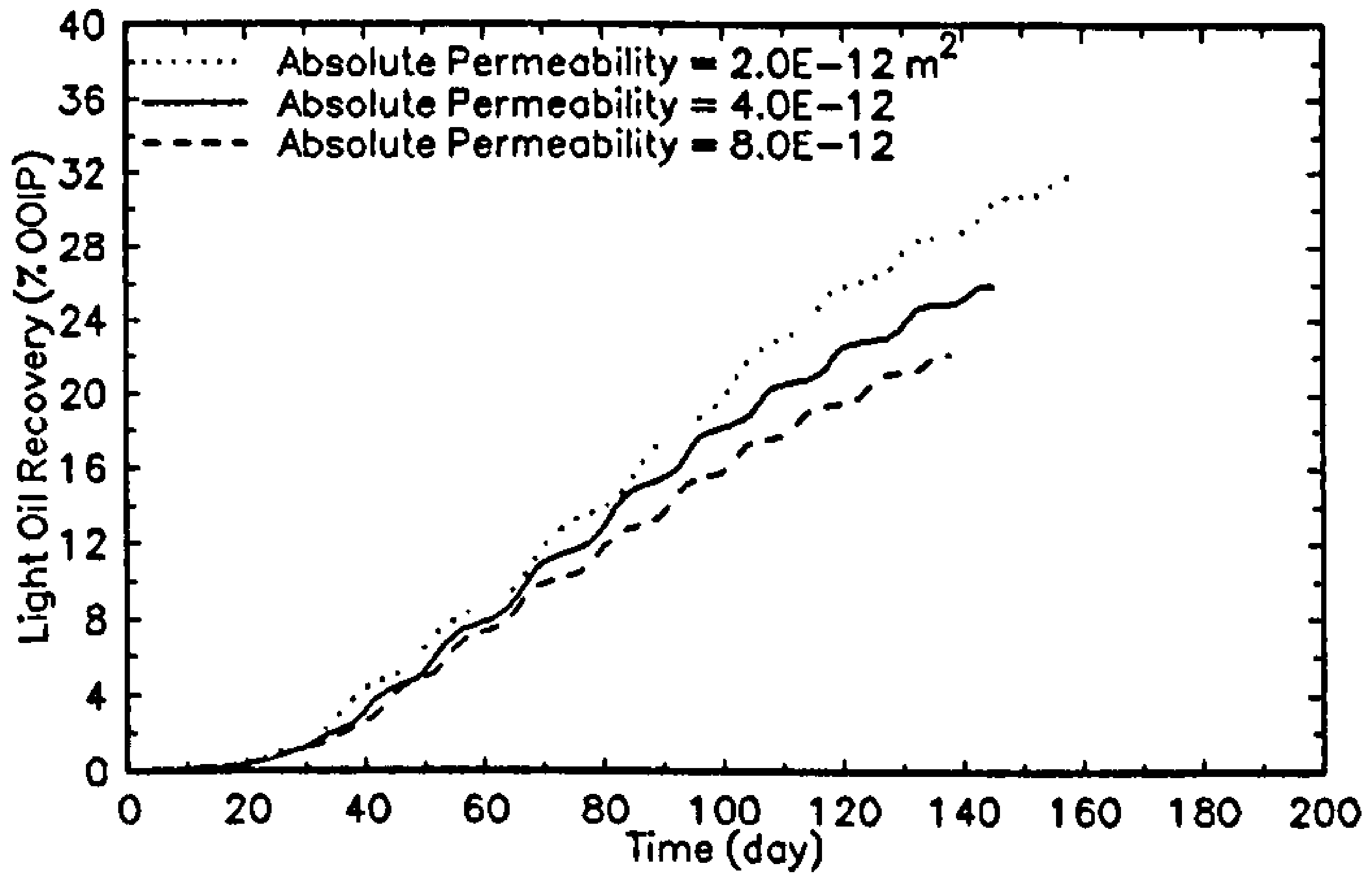


Figure 4.32 Light Oil Recovery Profile

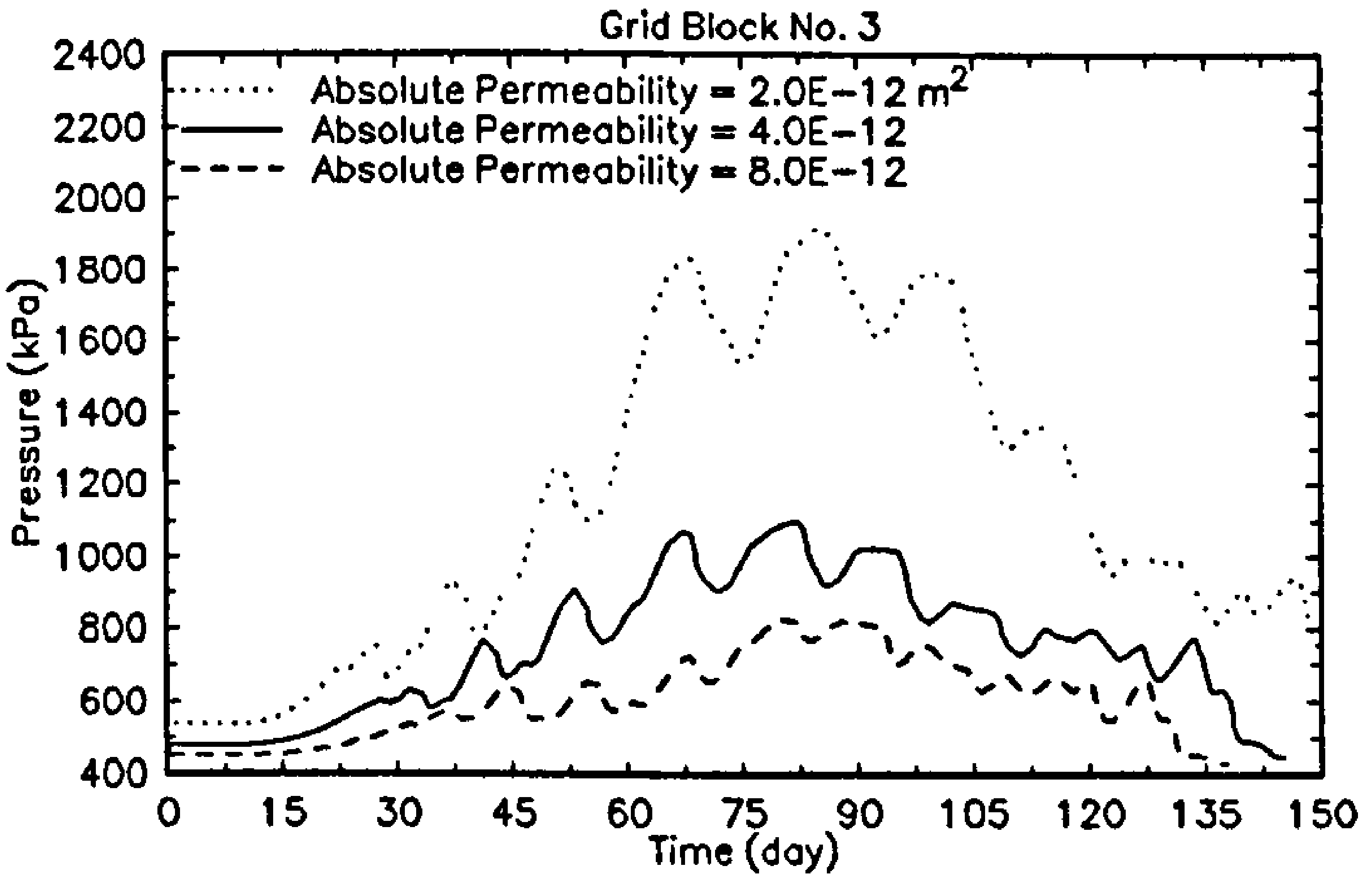


Figure 4.33 Pressure Profile Of Grid Block 3

4.3.3 Effects Of Oil Viscosity

The oil phase viscosity is calculated using the following dependence on composition;

$$\mu_o = (\mu_{o3})^{X_3} (\mu_{o4})^{X_4} \qquad (4.6)$$

where μ_{o3} and μ_{o4} are the light and heavy oil viscosities which are calculated using de Guzmar-Andrade's equation for temperature dependence (Watson [1943]);

$$\mu_{o3} = AV_3 \, e^{(BV_3/T)} \qquad (4.7)$$

$$\mu_{o4} = AV_4 \, e^{(BV_4/T)} \qquad (4.8)$$

where AV_3, AV_4, BV_3, and BV_4 are constants

T is the temperature

X_3 and X_4 are the light and heavy oil compositions in the oil phase.

In order to study the effect of oil viscosity on the characteristic performance of the in situ combustion process, the values of the constants AV_3 and AV_4 are changed.

Equation 4.7 shows the relationship between the light oil viscosity and AV_3. It can be seen from this equation that the increase in AV_3 leads to an increase in the light oil viscosity and consequently lower oil phase mobility. Table 4.12 shows the effects of the changes in AV_3.

Since the initial oil in the reservoir is usually a heavy oil only, in the present context and the only light oil produced arises from the cracking reaction which happens at a temperature where all the light oil turns to vapour, It can be seen that the increase in AV_3 (increase in the light oil viscosity) does not have any significant effect on the overall performance of the process

Equation 4.8 shows the relationship between the heavy oil viscosity and AV_4. It can be seen that the increase in AV_4 leads to an increase in the heavy oil viscosity and consequently lower oil phase mobility. Table 4.13 shows the effect of the changes in AV_4.

The increase in AV_4 (increase in the heavy oil viscosity) did not show any significant effect on the maximum amount of oil and water which can be recovered regardless of time. However, a slight increase can be seen in the amount of light oil produced which is due to the low mobility of the heavy oil (an increase in the cracking reaction rate). Also, the total production time has increased with the increase in the heavy oil viscosity due to the lower oil phase mobility.

Since the average peak temperature is a function of the amount of the fuel available which increases with the increase in the viscosity, it can be seen that the average peak temperature increases with the increase in the heavy oil viscosity. This increase in the peak temperature is also shown in figure 4.34 which presents the temperature profile of a typical grid block from the reservoir (grid block three).

The slight increase in the heat losses to the surrounding strata with the increase in the viscosity is due to the increase in the overall temperature of the formation.

Table 4.12 Effect of light oil viscosity

LIGHT OIL VISCOSITY (kPa.day)			
VARIABLES	$AV_3 * 10^{13}$		
	4.820	2.410*	1.205
WATER RECOVERY (% OF OWIP)	116.7	117.4	117.0
TOTAL OIL RECOVERY (% OOIP)	94.0	95.3	94.2
LIGHT OIL RECOVERY (% OOIP)	25.9	26.3	25.8
AVERAGE PEAK TEMPERATURE (K)	657.7	622.0	621.7
MAXIMUM PRESSURE (kPa)	1191.4	1242.0	1113.6
TOTAL PRODUCTION TIME (day)	145.0	145.1	145.5
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	26.5	25.0	24.9

* Original case study

Table 4.13 Effect of heavy oil viscosity

HEAVY OIL VISCOSITY (kPa.day)			
VARIABLES	$AV_4 * 10^{15}$		
	8.380	4.190*	2.095
WATER RECOVERY (% OF OWIP)	119.0	117.4	117.2
TOTAL OIL RECOVERY (% OOIP)	96.1	95.3	94.1
LIGHT OIL RECOVERY (% OOIP)	30.9	26.3	21.8
AVERAGE PEAK TEMPERATURE (K)	667.0	622.0	588.8
MAXIMUM PRESSURE (kPa)	1388.8	1242.0	937.3
TOTAL PRODUCTION TIME (day)	161.5	145.1	135.0
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	26.1	25.0	24.0

* Original case study

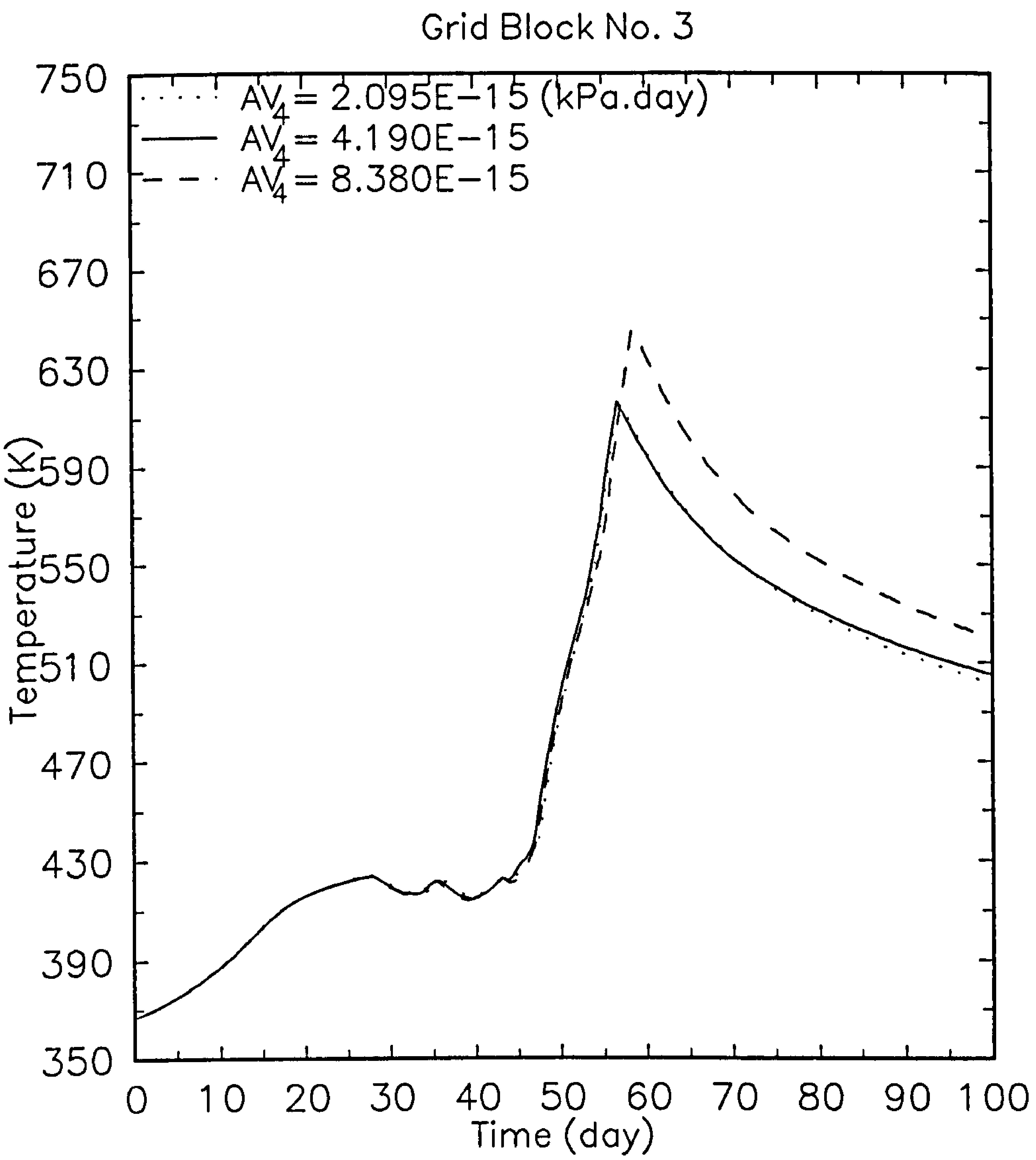


Figure 4.34 Temperature Profile Of Grid Block 3

4.4 Reservoir Parameters Effects

4.4.1 The Effects Of Porosity

The amount of oil (and other phases) in a system is directly proportional to its porosity. The porosity is expected to have an effect on the peak temperature, the velocity of the combustion front, and oil and water recovery. Table 4.14 shows the effect of the porosity. Values of the porosity from 0.19 to 0.76 were considered. Although the latter value is unrealistic, it can provide a general insight into the performance characteristics of in-situ combustion.

The rate of the light oil oxidation, heavy oil oxidation, and heavy oil cracking reactions are directly proprotional to the amount of light and heavy oil which is calculated as the product of porosity, oil saturation, oil density, and the light or heavy oil mole fraction. Therefore, it can be concluded that these reactions rates are directly proportional to the porosity.

It can be seen from table 4.14 that the average peak temperature is dramatically increased with increased porosity. This increase in the temperature is also shown in figure 4.35 which presents the temperature profile of a typical grid block in the reservoir (grid block three).

Table 4.14 also shows that the total a mount of water recovered increases with the decrease. in the porosity. This increase results from the consequent decrease in the rates of reactions (decrease in the overall heat of reactions) with the decrease in the porosity since they are directly proportional to the porosity. With the decrease in the reaction rates, more oil is needed to provide the fuel necessary to sustain the combustion. Since water is a product of the oxidation reactions, it increases too. Figure 4.36 shows the water recovery profile.

The amount of time needed to achieve maximum recovery increases with the increase in the porosity as shown in table 4.14. This increase is due to the availability of more oil to be produced. Figure 4.37 shows the oil recovery profile.

Table 4.14 also shows that heat losses to the surrounding strata increase with the increase in porosity because of the higher overall temperatures attained.

Table 4.14 The effect of porosity

VARIABLES	Porosity		
	0.760	0.380*	0.190
WATER RECOVERY (% OF OWIP)	113.2	117.4	135.5
TOTAL OIL RECOVERY (% OOIP)	99.8	95.3	83.4
LIGHT OIL RECOVERY (% OOIP)	26.2	26.3	17.8
AVERAGE PEAK TEMPERATURE (K)	972.7	622.0	543.4
MAXIMUM PRESSURE (kPa)	1237.7	1242.0	887.2
TOTAL PRODUCTION TIME (day)	229.8	145.1	127.3
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	29.5	25.0	23.0

* Original case study

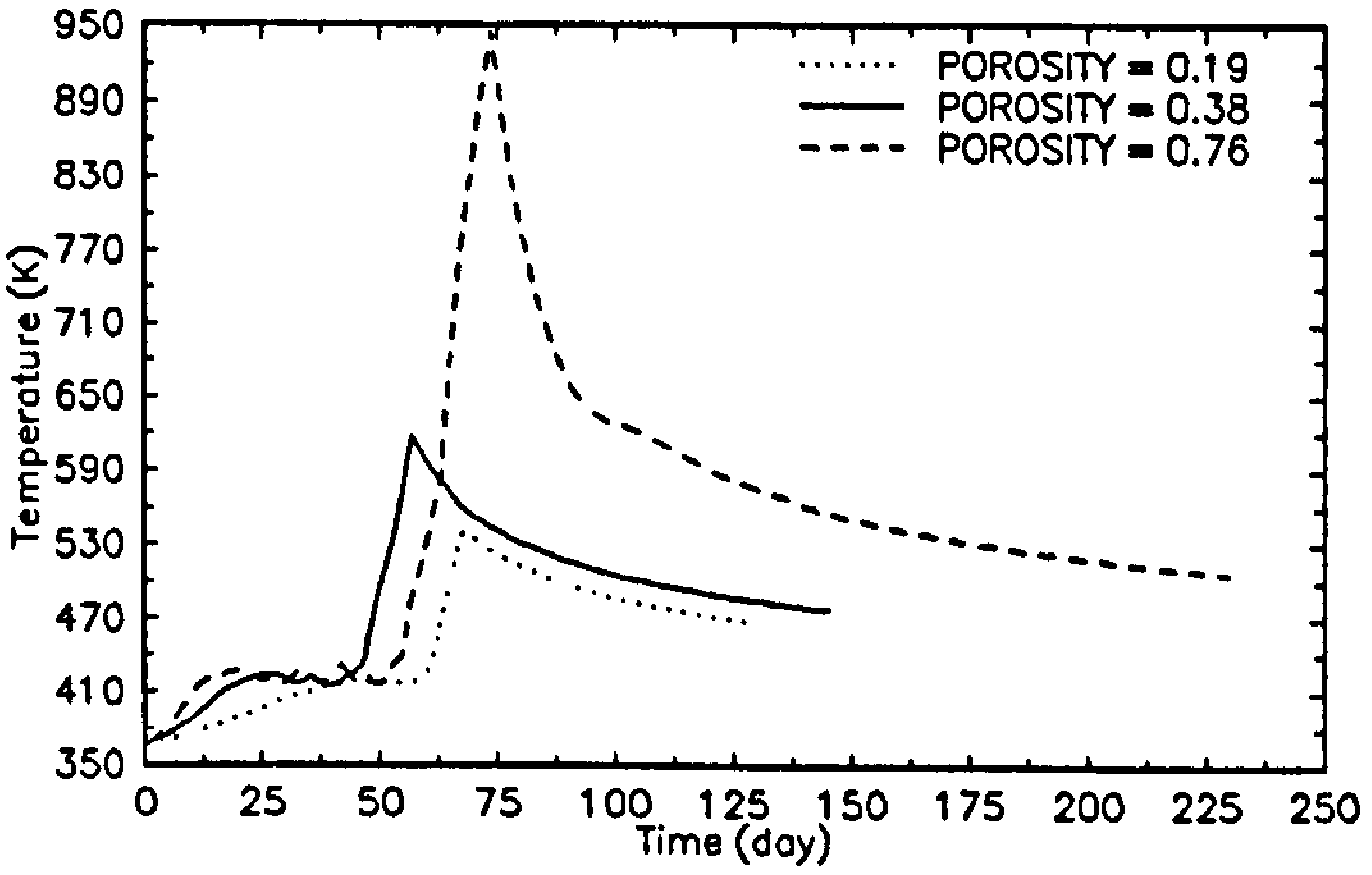


Figure 4.35 Temperature Profile Of Grid Block 3

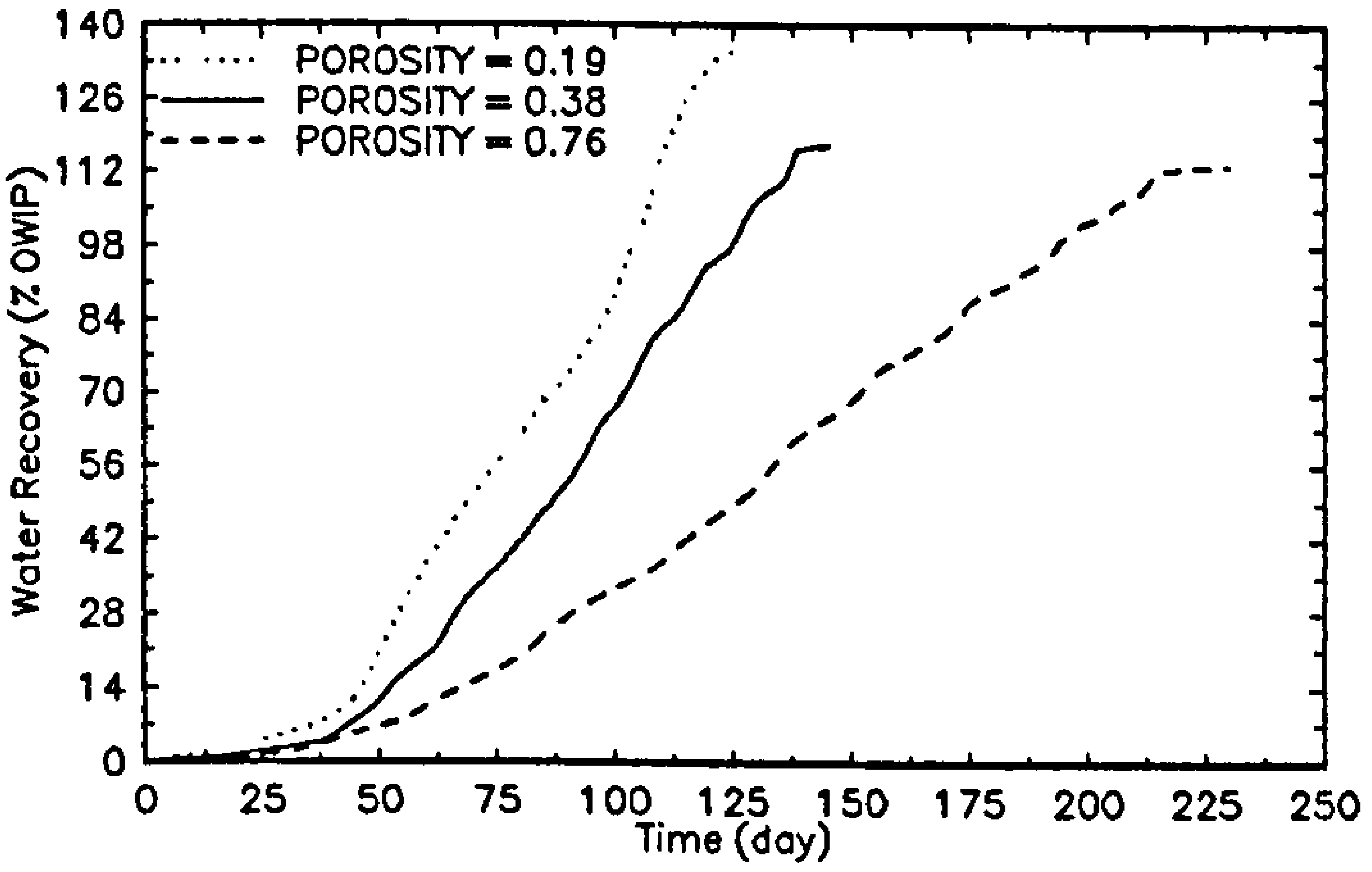


Figure 4.36 Water Recovery Profile

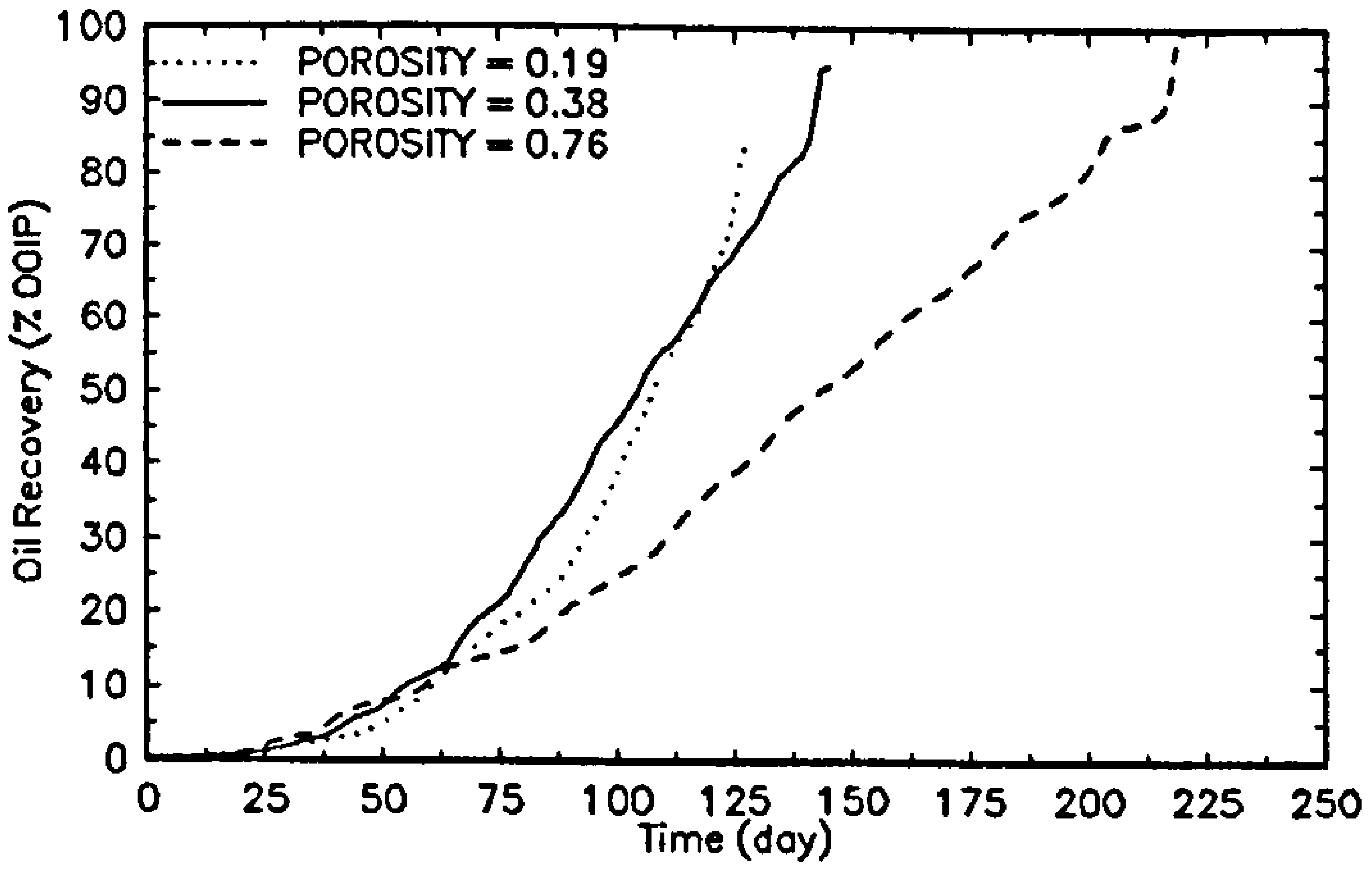


Figure 4.37 Oil Recovery Profile

4.5 Thermal Parameters Effects

4.5.1 The Effect of Thermal Conductivity of the Formation

Table 4.15 shows the effect of changes in the thermal conductivity of the formation. It can be seen that variations in the thermal conductivity show only a negligible effect on the maximum amount of oil, water, and light oil which can be recovered regardless of time.

However, as table 4.15 shows, the average peak temperature decreases considerably with increase in the thermal conductivity. This decrease in the peak temperature results from the fact that a higher proportion of heat is lost to the surrounding strata rather than being carried downstream by conduction and convection. This decrease in the peak temperature is also shown in figure 4.38 which presents the temperature profile of a typical grid block in the reservoir (grid block three). This case signifies the fact that radial conduction is more influential than axial conduction.

Figure 4.38 also shows that a longer time is needed for the temperature to reach the peak as the thermal conductivity is increased. Hence, the combustion front will have a slower velocity because the lower temperatures result in lower reaction rates. It can also be seen that after the combustion is completed, the temperature drops at a faster rate with higher thermal conductivity as a result of the higher heat loss rate to the surrounding strata

When the thermal conductivity of the surrounding strata was changed from 240 to 0.00 (kJ/M.day.K) (the adiabatic case), figure 4.38 shows that a higher peak temperature was obtained since no heat is lost to the surrounding strata and all the heat generated is carried downstream. Also, it shows that the temperature drops at a very slow rate after the combustion is completed since the only cooling effect is caused by the injected gas.

Table 4.15 The effects of the thermal conductivity

VARIABLES	Thermal conductivity (kJ/K.day.M)			
	480.0	240.0[x]	120.0	0.000[+]
WATER RECOVERY (% OF OWIP)	117.2	117.4	117.2	116.2
TOTAL OIL RECOVERY (% OOIP)	93.5	95.3	95.0	97.9
LIGHT OIL RECOVERY (% OOIP)	25.8	26.3	25.8	25.7
AVERAGE PEAK TEMPERATURE (K)	596.7	622.0	639.6	707.4
MAXIMUM PRESSURE (kPa)	1196.1	1242.0	1087.3	1052.9
TOTAL PRODUCTION TIME (day)	150.7	145.1	141.9	130.3
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	29.5	25.0	20.5	00.0

* Original case study

\+ Adiabatice case

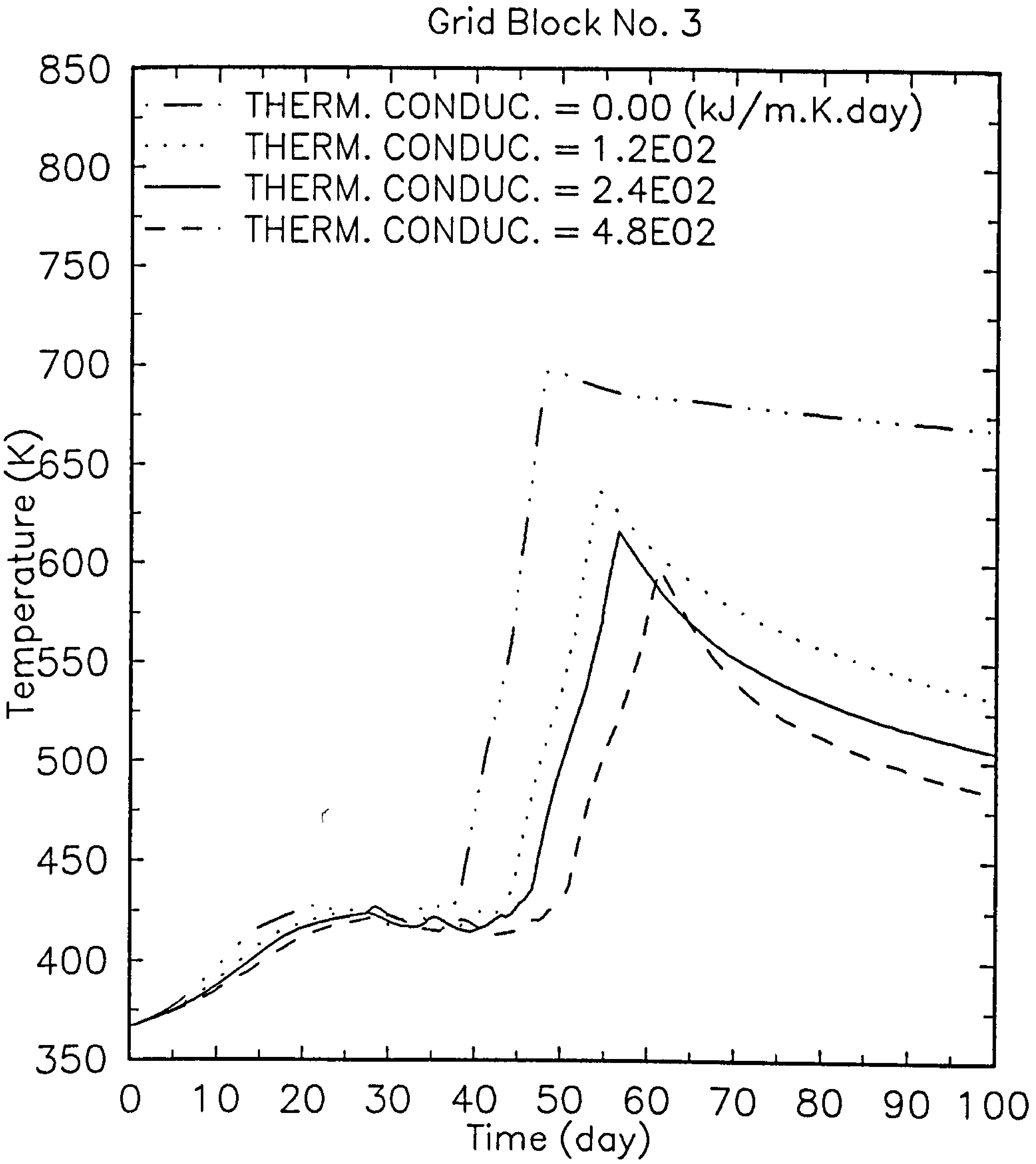


Figure 4.38 Temperature Profile Of Grid Block 3

4.5.2 The Effect Of Variation Of Reaction Enthalpies

Table 4.16 shows the effects of the changes in the heat of the light oil oxidation reaction. It can be seen that changes in the heat of this reaction do not have any significant effect on the overall performance characteristics of the process. This is because there is no initial light oil in the reservoir.

Table 4.17 shows the effect of the changes in the heat of heavy oil oxidation reaction. Increase in the heat of reaction increases the average peak temperature due to the increase in the heat source available and consequently, an increase in the oil mobility which is shown by the decrease in the total production time. The increase in temperature is also shown in figure 4.39 which presents the temperature profile of a typical grid block from the reservoir (grid block three).

Table 4.17 also shows a decrease in the water produced and a slight increase in the oil produced with the increase in the heat of reaction. This is also due to increase in the oil mobility. Figure 4.40 shows the water recovery profile.

Table 4.18 shows the effect of the changes in the heat of the heavy oil cracking reaction. It should be pointed out that although the heat of this reaction is taken as exothermic following the data constructed from Crookston et al [1979], this reaction is normally endothermic. Due to the low amount of heat generated by this reaction compared with the other reactions, it can be seen that the changes in the heat of reaction do not show any effect on the overall performance characteristics of the process.

The heat of the coke oxidation reaction is well established and the changes made here on its value are only hypothetical and do not have any physical meaning. However, this case can help to test the simulator robustness.

Table 4.19 shows the effect of the changes in the heat of coke oxidation reaction. It can be seen that the increase in the heat of this reaction leads to a significant increase in the average peak temperature. This is due to the large contribution of this reaction in providing the heat source for this process. The increase in the temperature is also shown in figure 4.41 which presents the temperature profile of a typical grid block from the reservoir (grid block three).

Table 4.19 also shows an increase in the maximum amount of oil that can be recovered with the increase in the heat of reaction. This increase is caused by the higher oil mobility resulted from the increase in the heat of reaction (an increase in the heat source). Consequently, the heavy oil cracking reaction does not have sufficient time to crack more heavy oil. The increase in the oil mobility also explains the reduction in the time needed to achieve maximum recovery. Also, the reduction in the amount of heavy oil to be cracked explains the decrease in the light oil recovery since light oil is one of the products of the heavy oil cracking reaction. Figures 4.42 and 4.43 show the oil and light oil recovery profiles for this case.

An increase in the pressure is also shown in table 4.19 with the increase in the heat of reaction which is due to the increase in the oil and water vapor pressures resulting from the higher overall temperatures. This increase in the pressure is also shown in figure 4.44 which presents the pressure profile in a typical grid block of the reservoir (grid block three).

Table 4.16 The effect of heat of reaction

Light Oil Oxidation			
VARIABLES	HEAT OF REACTION (kJ/mol)		
	4420.0	2210.0*	1105.0
WATER RECOVERY (% OF OWIP)	116.9	117.4	116.9
TOTAL OIL RECOVERY (% OOIP)	94.7	95.3	94.5
LIGHT OIL RECOVERY (% OOIP)	25.8	26.3	25.5
AVERAGE PEAK TEMPERATURE (K)	618.7	622.0	618.4
MAXIMUM PRESSURE (kPa)	1111.3	1242.0	1114.5
TOTAL PRODUCTION TIME (day)	145.2	145.1	143.7
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	24.9	25.0	24.8

* Original case study

Table 4.17 The effect of heat of reaction

Heavy Oil Oxidation			
VARIABLES	HEAT OF REACTION (kJ/mol)		
	16240.0	8120.0*	4060.0
WATER RECOVERY (% OF OWIP)	113.3	117.4	124.2
TOTAL OIL RECOVERY (% OOIP)	96.5	95.3	91.1
LIGHT OIL RECOVERY (% OOIP)	25.6	26.3	26.3
AVERAGE PEAK TEMPERATURE (K)	643.5	622.0	598.6
MAXIMUM PRESSURE (kPa)	1251.0	1242.0	941.3
TOTAL PRODUCTION TIME (day)	125.8	145.1	184.8
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	25.0	25.0	24.7

* Original case study

Table 4.18 The effect of heat of reaction

Heavy Oil Cracking			
VARIABLES	HEAT OF REACTION (kJ/mol)		
	93.00	46.50*	23.325
WATER RECOVERY (% OF OWIP)	117.4	117.4	116.8
TOTAL OIL RECOVERY (% OOIP)	95.0	95.3	94.4
LIGHT OIL RECOVERY (% OOIP)	26.1	26.3	25.3
AVERAGE PEAK TEMPERATURE (K)	626.3	622.0	620.1
MAXIMUM PRESSURE (kPa)	1153.7	1242.0	1100.1
TOTAL PRODUCTION TIME (day)	145.6	145.1	145.8
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	25.0	25.0	24.9

* Original case study

Table 4.19 The effect of heat of reaction

Coke Oxidation			
VARIABLES	**HEAT OF REACTION (kJ/mol)**		
	1046.0	523.0*	261.5
WATER RECOVERY (% OF OWIP)	114.6	117.4	122.1
TOTAL OIL RECOVERY (% OOIP)	99.2	95.3	90.3
LIGHT OIL RECOVERY (% OOIP)	24.7	26.3	27.6
AVERAGE PEAK TEMPERATURE (K)	862.2	622.0	546.5
MAXIMUM PRESSURE (kPa)	1464.5	1242.0	921.4
TOTAL PRODUCTION TIME (day)	126.1	145.1	179.0
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	25.8	25.0	25.0

* Original case study

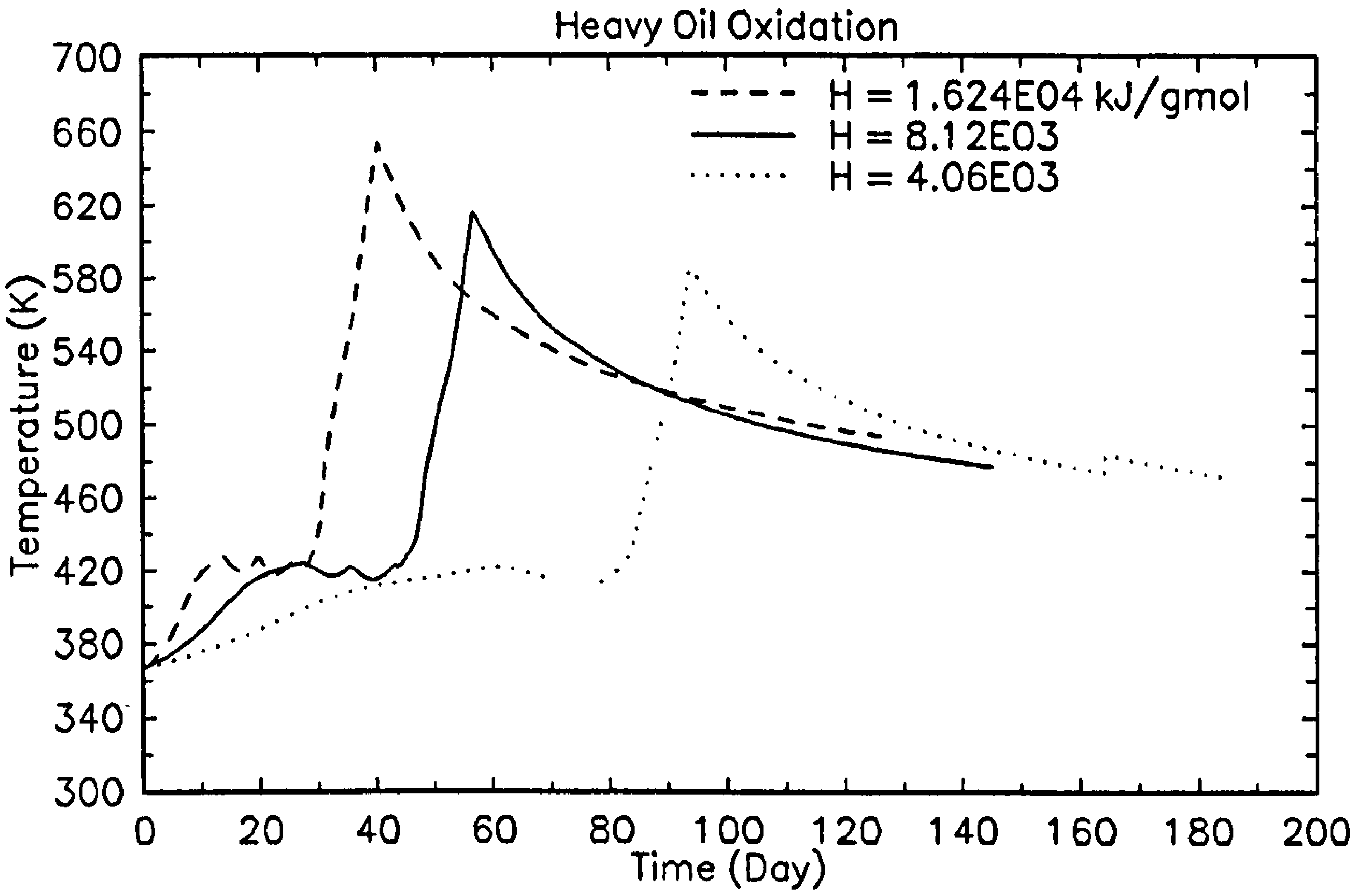


Figure 4.39 Temperature Profile Of Grid Block 3

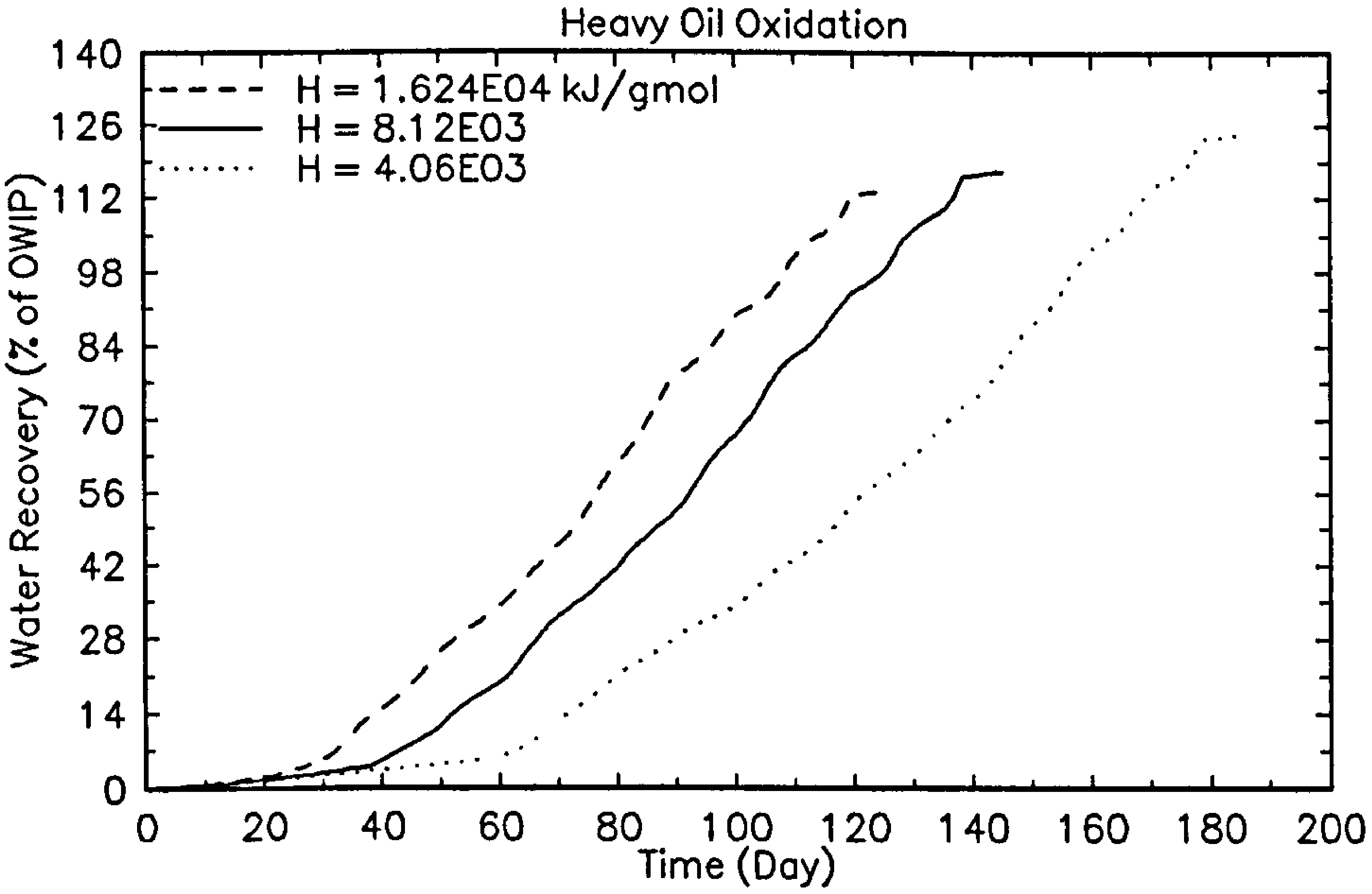


Figure 4.40 Water Recovery Profile

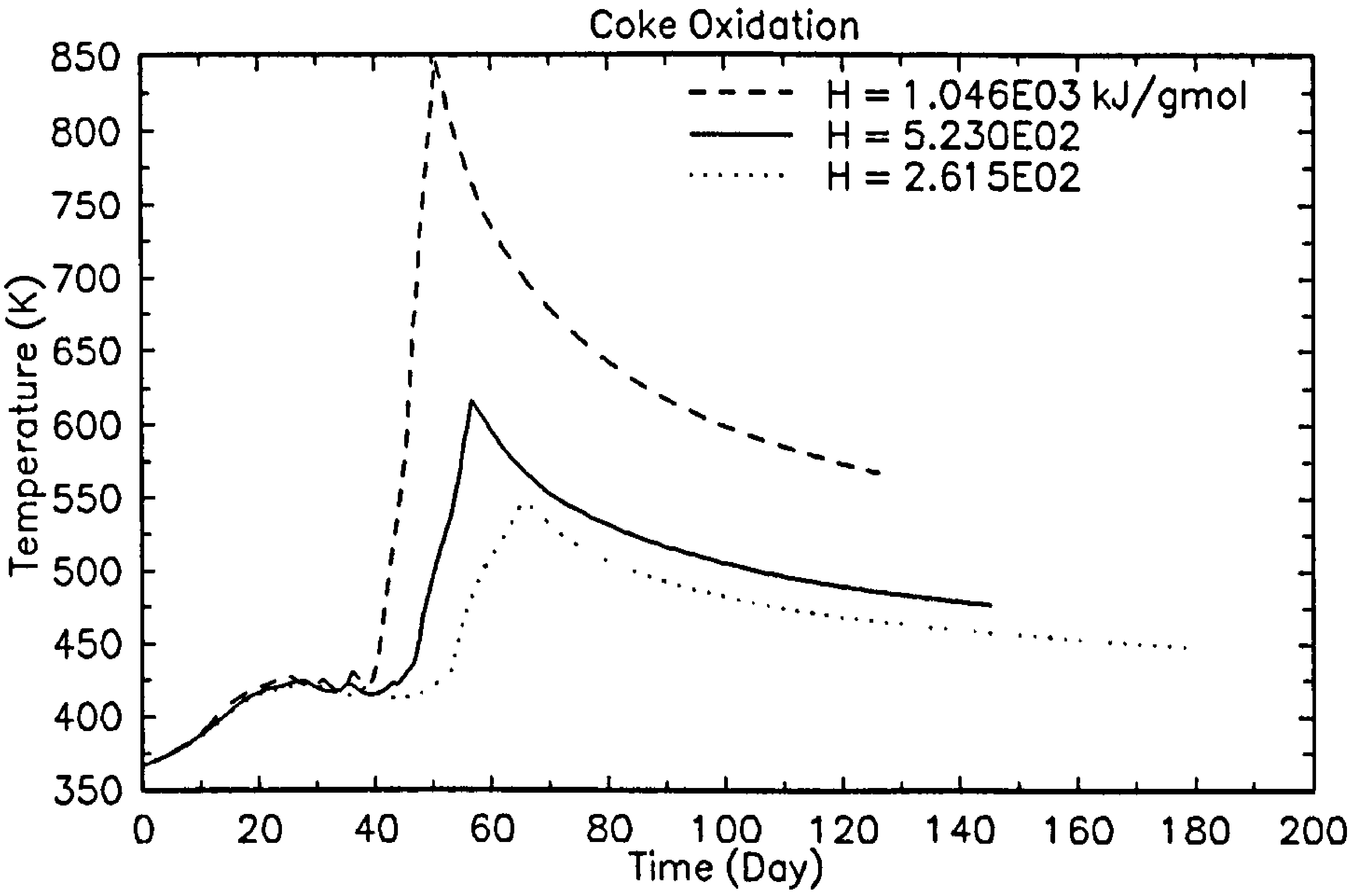


Figure 4.41 Temperature Profile Of Grid Block 3

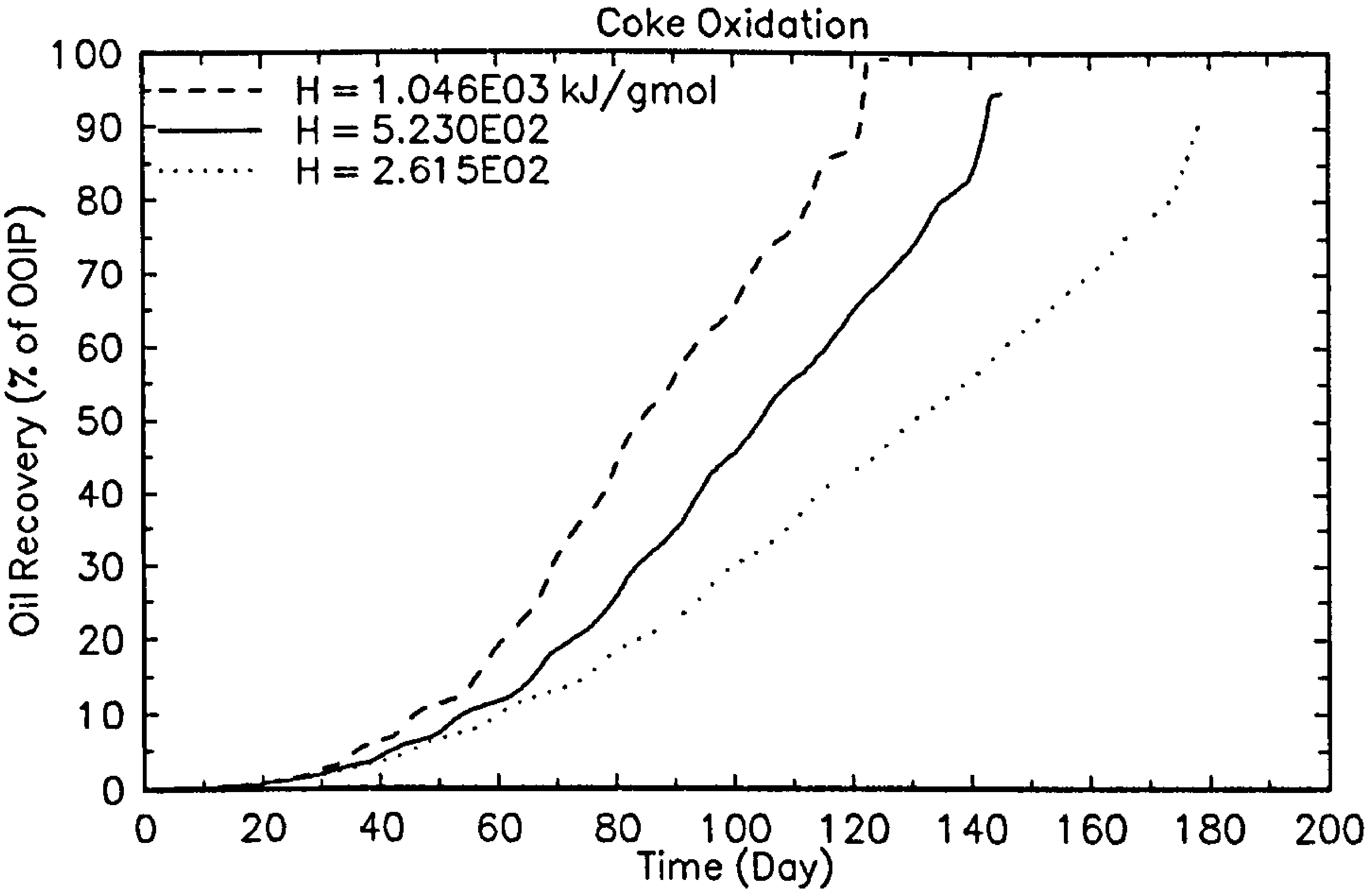


Figure 4.42 Oil Recovery Profile

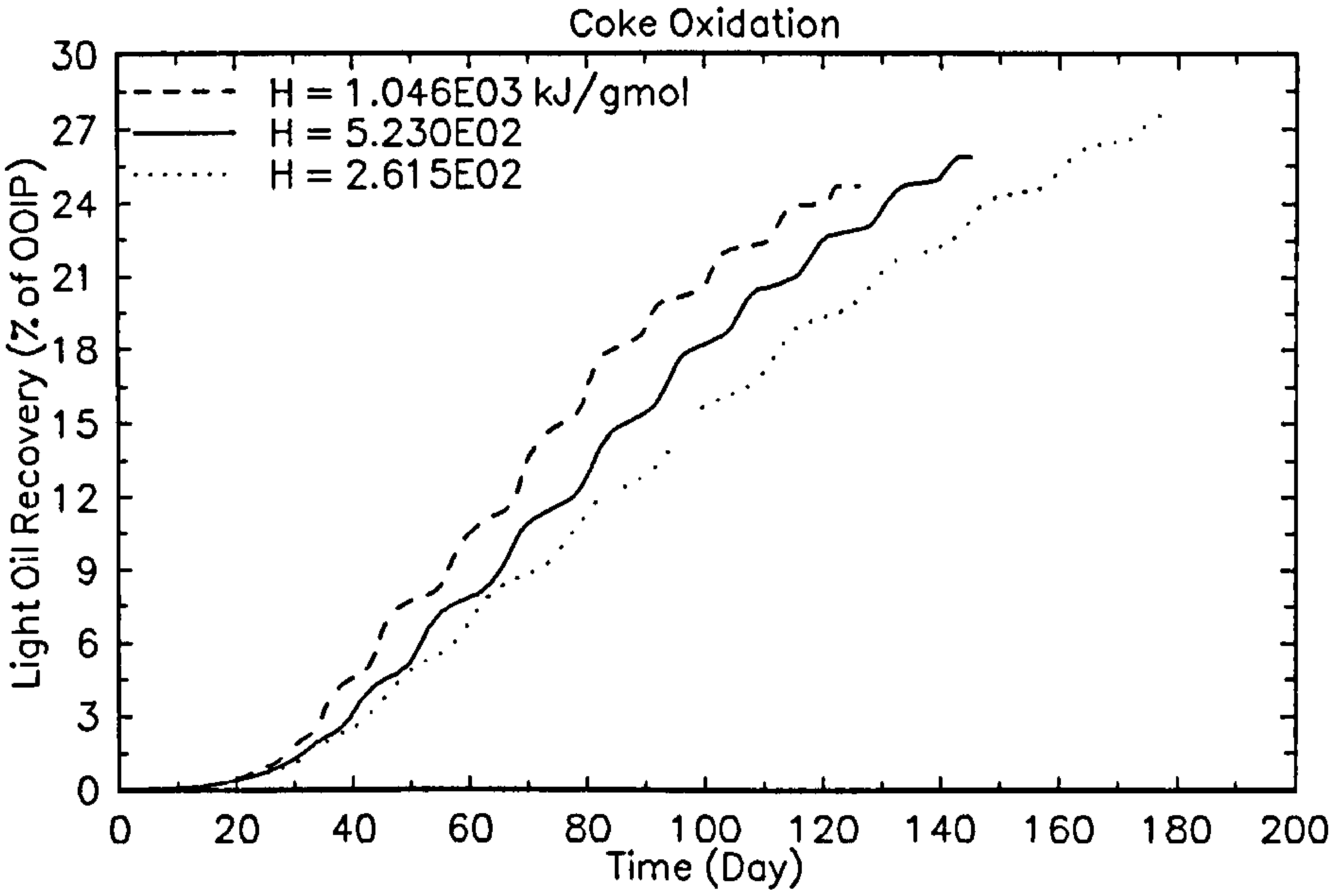


Figure 4.43 Light Oil Recovery Profile

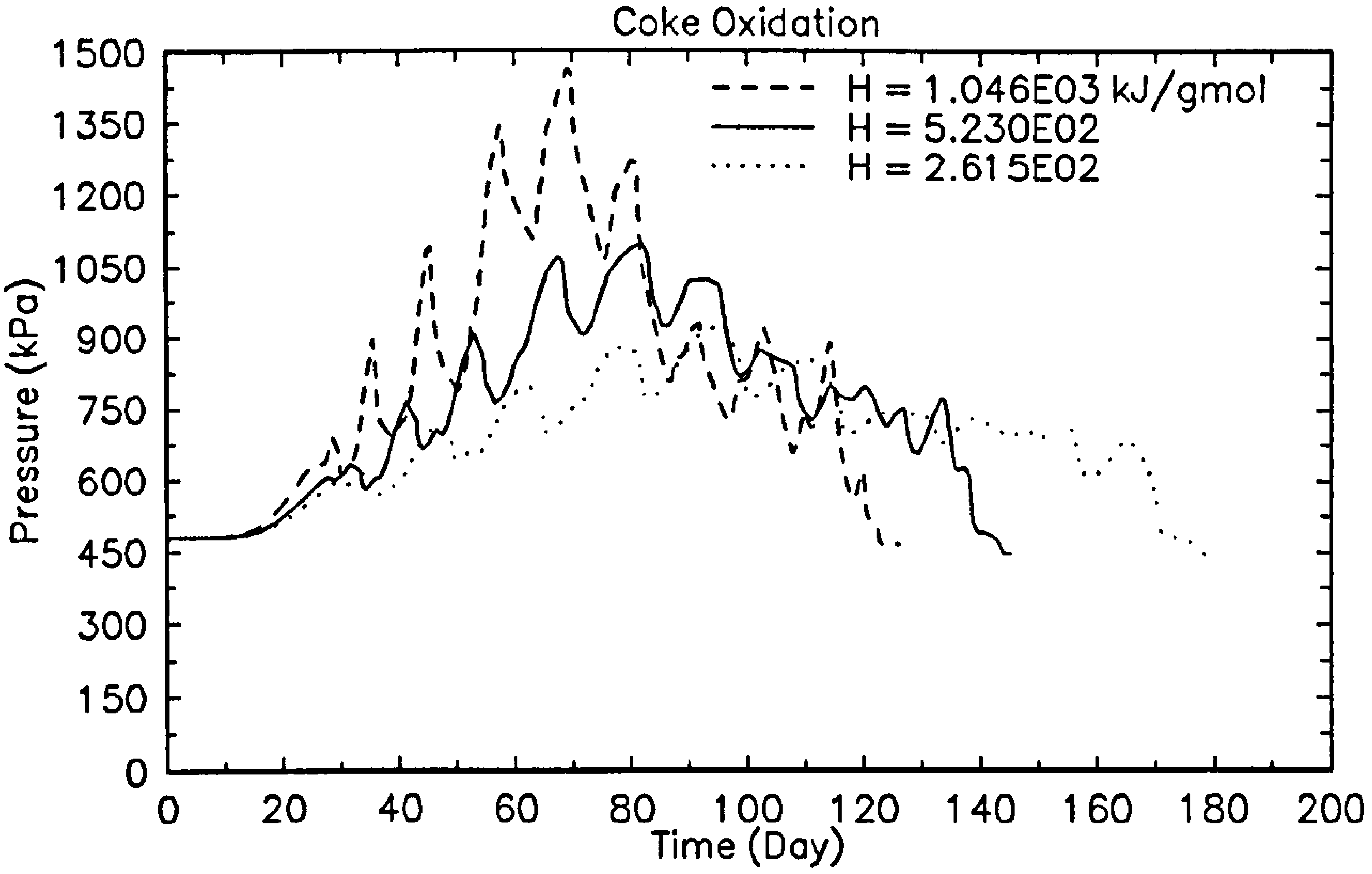


Figure 4.44 Pressure Profile Of Grid Block 3

4.6 Kinetic Parameter Effects

4.6.1 The Effect of Arrhenius Constant (The pre-exponential constant in the reaction rate equation)

Tables 4.20, 4.21, 4.22, and 4.23 show the effect of the changes in the Arrhenius constant of the four reactions.

Since no initial light oil exists in the reservoir, table 4.20 shows that the changes in Arrhenius constant of the light oil oxidation reaction do not have any significant effect on the overall performance of the process.

In order to study the effect of removing this reaction, table 4.20 also shows a case in which the Arrhenius constant is set to zero. It can be seen that removing this reaction does not have any effect on the process. However, this conclusion cannot be generalized unless a more comprehensive study is done on different data sets in some of which the initial oil in the reservoir contains some light oil.

Table 4.21 shows the effect of the changes in Arrhenius constant of the heavy oil oxidation reaction. It can be seen from that table that the average peak temperature increases with the increase in the Arrhenius constant. This increase in the temperature which is also shown in figure 4.45, is caused by the increase in the rate of this exothermic reaction.

Also, as can be seen from table 4.21, the time needed to achieve maximum recovery decreases slightly with the increase of the Arrhenius constant. This decrease is due to the higher oil mobility which results from the higher overall temperatures. However, no significant change is encountered in the maximum amount of oil which can be recovered regardless of time and the slight decrease in the water recovery with the increase in Arrhenius constant which is also shown in figure 4.46, is due to the decrease in the amount of oil needed to provide the necessary heat to sustain the combustion.

Table 4.21 also reveals that the heat losses to the surrounding strata increase slightly with the increase in Arrhenius constant which is due to the higher overall temperatures.

Another run was done in order to study the significance of the heavy oil oxidation reaction. In this run, the Arrhenius constant was set to zero i.e. the reaction is removed. This case did not lead to ignition and the temperature did not rise enough to start the coke oxidation reaction. This leads to the conclusion that the heavy oil oxidation reaction plays an important role in the process.

Table 4.22 shows that when the Arrhenius constant of the heavy oil cracking reaction increases, the average peak temperature increases dramatically. This increase is due to the increase in the rate of this reaction and consequently the increase in the amount of coke produced. The additional coke provides extra heat which leads to the overall increase in the temperature. This increase in temperature is also shown in figure 4.47 which presents the temperature profile of a typical grid block of the reservoir (grid block three).

Table 4.22 also shows a decrease in the maximum amount of heavy oil which can be recovered when the Arrhenius constant is increased from 1.5 to 6.0 1/day (decreased from 74.4% to 59.9% of the OOIP). This decrease is caused by the increase in the rate of this reaction which means more heavy oil is converted to coke and light oil. This also explains the increase in the light oil recovery with the increase in Arrhenius constant. The changes on the oil and light oil recovery profiles can be seen in figures 4.48 and 4.49 respectively.

Inspection of the results in table 4.23 reveals that with the increase in the rate of the coke oxidation reaction, the average peak temperature and the total production time decrease slightly. The reason for the decrease in the temperature is that coke consumption is occurring at a lower temperature that it would normally do due to the fact that at any given temperature, the rate of a reaction increases with the increase in the Arrhenius constant of that reaction. The decrease in the production time is attributed to the faster combustion front which results from the increase in the rate of the coke oxidation reaction. Apart from that, table 4.23 shows that the changes in Arrhenius constant of reaction four did not have a significant effect on the overall performance of the process.

Column four in tables 4.22 and 4.23 represents the result of a case in which the Arrhenius Constants of the heavy oil cracking and the coke oxidation reactions are set to zero at the same time (since ignoring one reaction at a time is meaningless). This case is done in order to study the importance of inclusion of these reactions. It can be seen that ignition still occurred but more oil had to be oxidized to provide the heat necessary to sustain the combustion. A shorter production time is needed because of the reduction in the a mount of oil to be recovered and the decrease in the average peak temperature is due to the lack of coke in this case.

Table 4.20 The effect of Arrhenius constant

Light Oil Oxidation Reaction				
VARIABLES	Arrhenius Constant (10^5 kPa/day)			
	2.90	1.45*	0.725	0.000
WATER RECOVERY (% OF OWIP)	116.9	117.4	116.7	116.7
TOTAL OIL RECOVERY (% OOIP)	94.5	95.3	94.3	94.5
LIGHT OIL RECOVERY (% OOIP)	25.6	26.3	25.5	25.5
AVERAGE PEAK TEMPERATURE (K)	617.9	622.0	617.2	619.0
MAXIMUM PRESSURE (kPa)	1115.5	1242.0	1100.3	1114.5
TOTAL PRODUCTION TIME (day)	144.5	145.1	144.3	144.2
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	24.8	25.0	24.8	24.8

* Original case study

Table 4.21 The effect of Arrhenius constant

Heavy Oil Oxidation Reaction			
VARIABLES	Arrhenius Constant (10^5 kPa/day)		
	2.90	1.45*	0.725
WATER RECOVERY (% OF OWIP)	115.5	117.4	119.0
TOTAL OIL RECOVERY (% OOIP)	94.2	95.3	93.6
LIGHT OIL RECOVERY (% OOIP)	25.3	26.3	26.8
AVERAGE PEAK TEMPERATURE (K)	635.0	622.0	609.0
MAXIMUM PRESSURE (kPa)	1271.7	1242.0	1038.5
TOTAL PRODUCTION TIME (day)	141.1	145.1	161.8
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	25.8	25.0	23.7

* Original case study

Table 4.22 The effect of Arrhenius constant

Heavy Oil Cracking Reaction				
VARIABLES	**Arrhenius Constant** (10^5 1/day)			
	6.00	3.00 *	1.50	0.00
WATER RECOVERY (% OF OWIP)	119.7	117.4	116.5	121.2
TOTAL OIL RECOVERY (% OOIP)	97.7	95.3	90.3	84.2
LIGHT OIL RECOVERY (% OOIP)	37.8	26.3	15.9	00.0
AVERAGE PEAK TEMPERATURE (K)	691.3	622.0	585.0	597.8
MAXIMUM PRESSURE (kPa)	1075.2	1242.0	1168.9	1456.5
TOTAL PRODUCTION TIME (day)	175.7	145.1	133.0	133.0
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	26.3	25.0	24.5	26.1

* Original case study

Table 4.23 The effect of Arrhenius constant

Coke Oxidation Reaction				
VARIABLES	Arrhenius Constant (10^5 kPa/day)			
	2.90	1.45*	0.725	0.00
WATER RECOVERY (% OF OWIP)	116.3	117.4	118.3	121.2
TOTAL OIL RECOVERY (% OOIP)	94.5	95.3	94.9	84.2
LIGHT OIL RECOVERY (% OOIP)	25.8	26.3	25.5	00.0
AVERAGE PEAK TEMPERATURE (K)	612.7	622.0	626.2	597.8
MAXIMUM PRESSURE (kPa)	1140.7	1242.0	1143.5	1456.5
TOTAL PRODUCTION TIME (day)	141.1	145.1	148.9	133.0
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	24.6	25.0	25.1	26.1

* Original case study

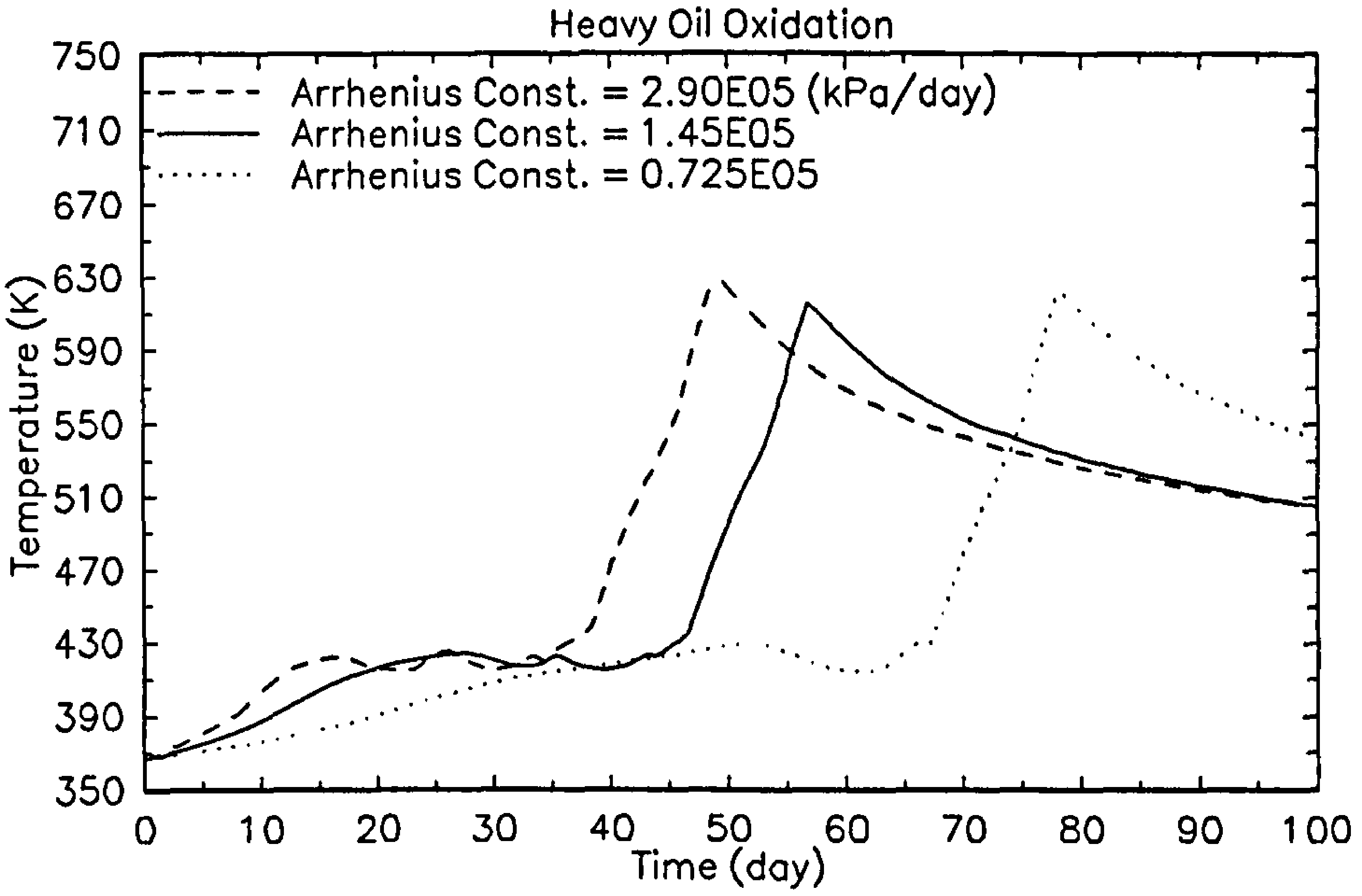


Figure 4.45 Temperature Profile Of Grid Block 3

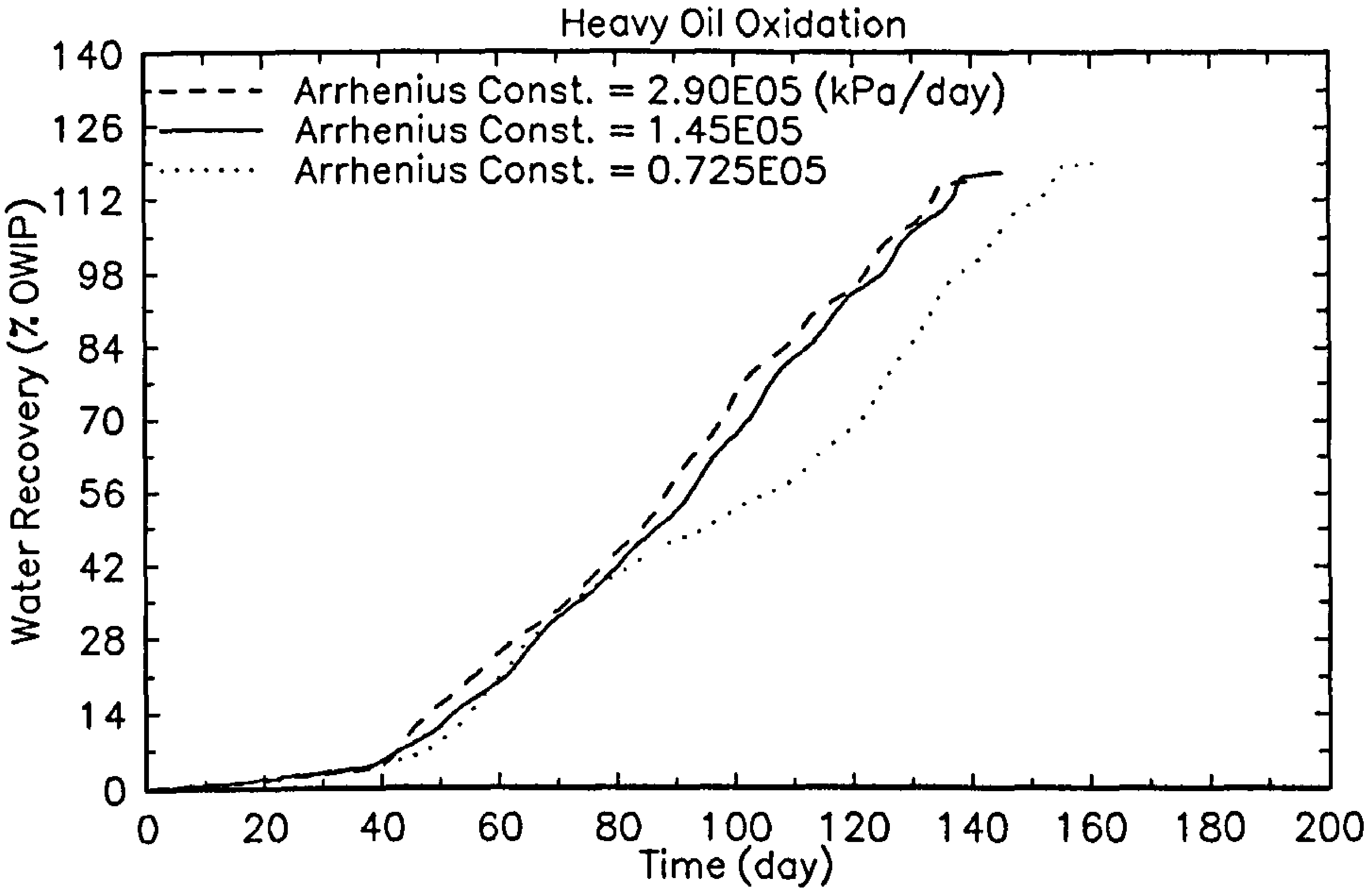


Figure 4.46 Water Recovery Profile

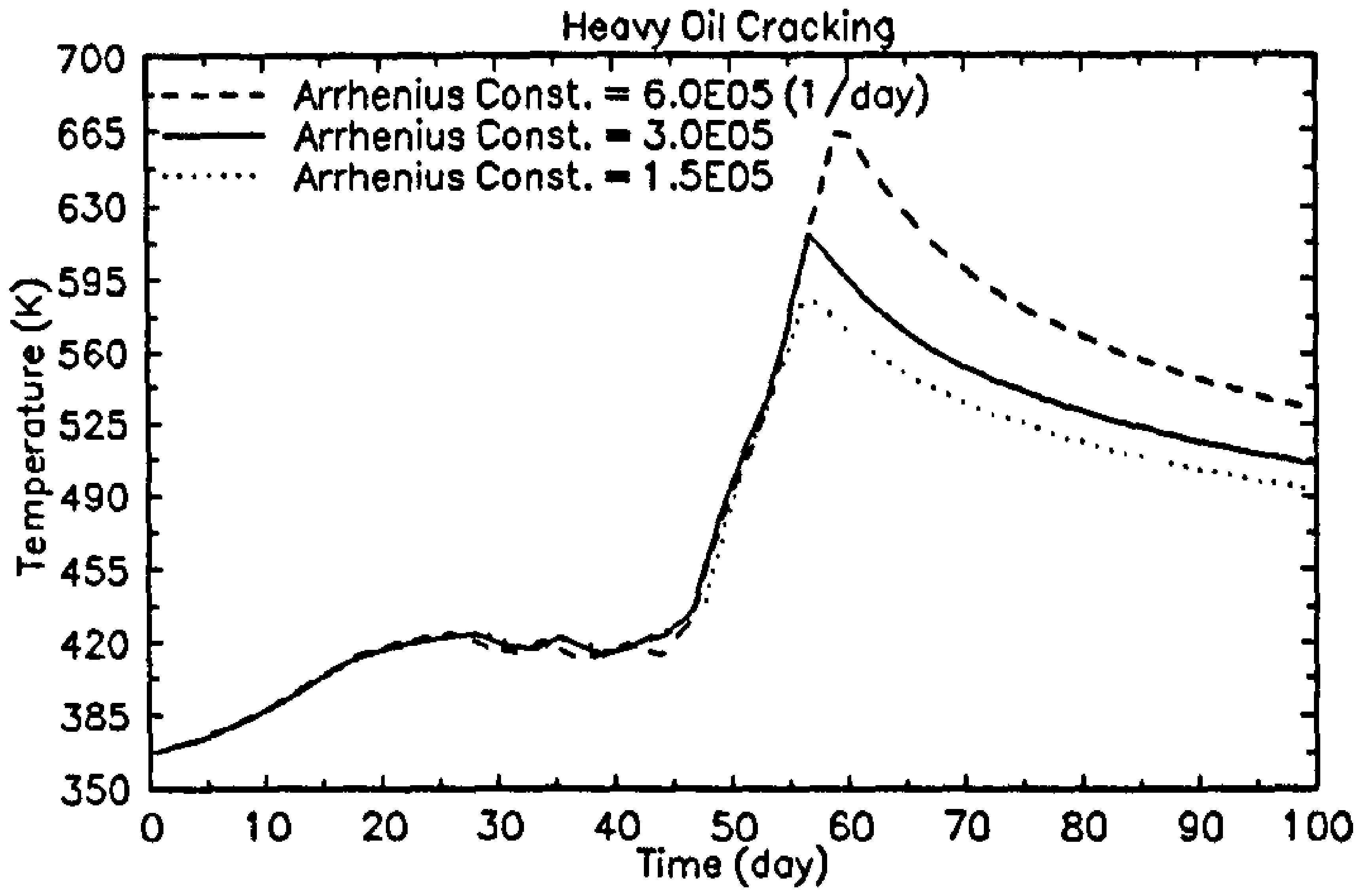


Figure 4.47 Temperature Profile Of Grid Block 3

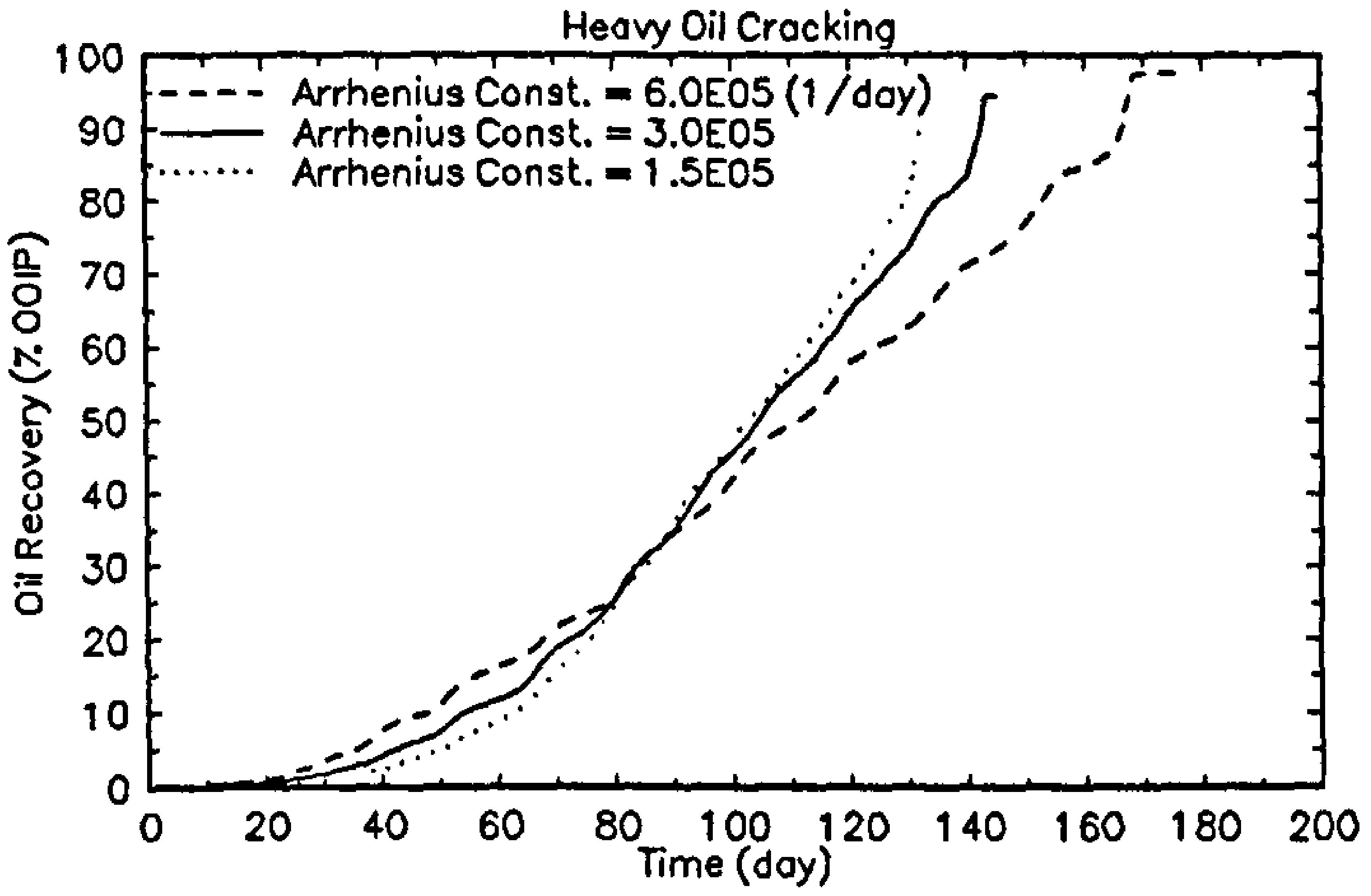


Figure 4.48 Oil Recovery Profile

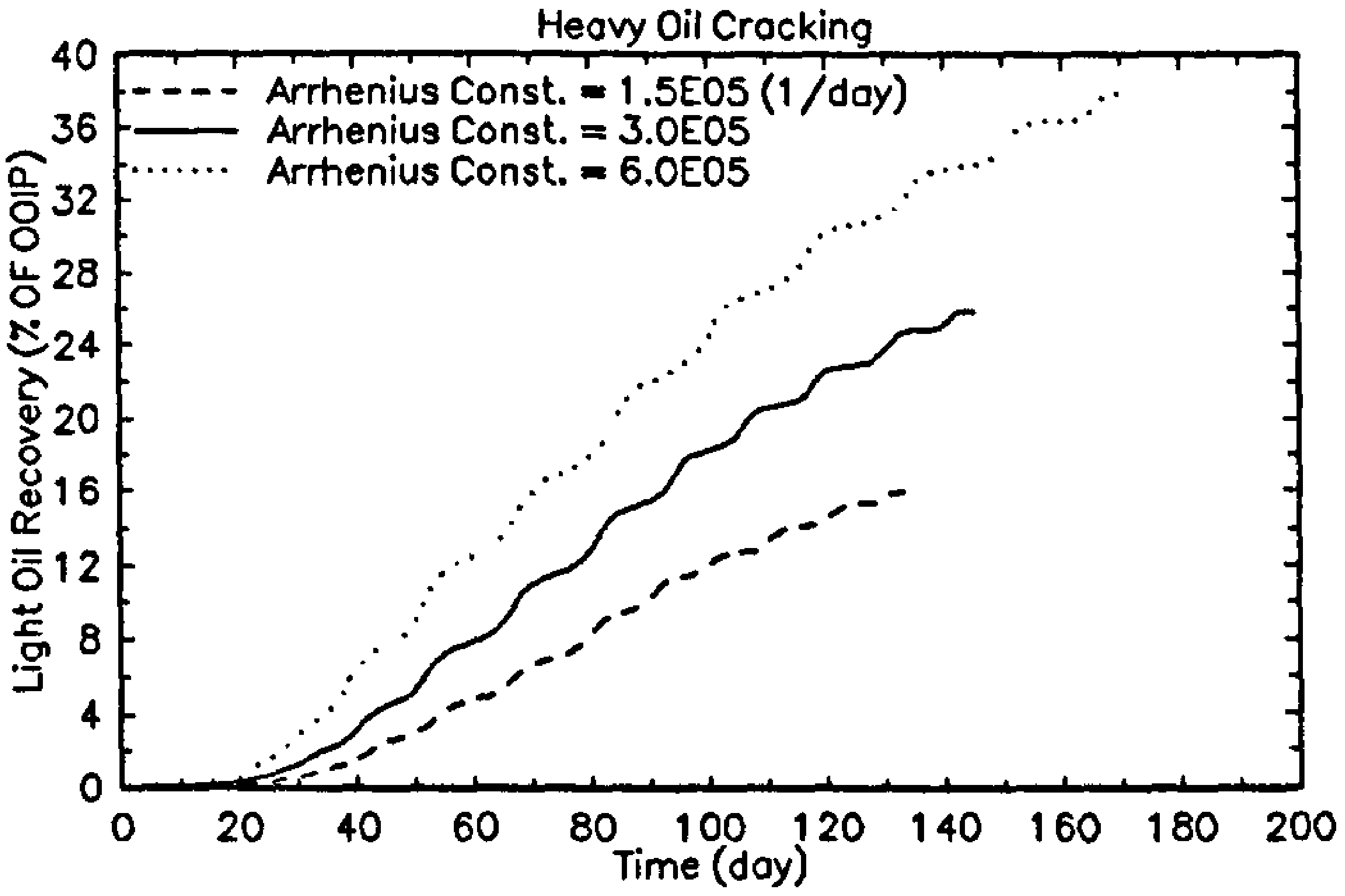


Figure 4.49 Light Oil Recovery

4.6.2 The Effect Of Activation Energy

The effect of the changes in the activation energy is expected to be more noticeable than those of the Arrhenius constants since the activation energy is an exponential parameter in the reaction rate equation.

Table 4.25 shows the effect of the changes in the activation energy of the light oil oxidation reaction. An increase in the average peak temperature can be seen with decreasing activation energy which is due to the increase in the rate of this reaction. The increase in the temperature is also shown in figure 4.50 which present the temperature profile of a typical block from the reservoir (grid block three). Also, a decrease in the light oil recovery can be observed with the decrease in the activation energy; this is also due to the increase in the reaction rate.

The time needed to achieve maximum recovery can be seen from table 4.25 to increase with decreasing activation energy. This increase is a result of the availability of more heavy oil to be recovered (the maximum amount of heavy oil that can be recovered when the activation energy is set to 38.73 kJ/gmol is (94.5-21.2=73.3 % of the OOIP) whereas it is 69% of the OOIP when the activation energy is set to 77.46).

Table 4.25 also shows an increase in the heat losses to the surrounding strata with decrease in the activation energy (increase in the reaction rate) which is due to the higher overall temperature of the formation.

Table 4.25 shows that when the activation energy is set to (154.92 kJ/gmol), this change does not have any significant effect on the overall performance of the process. This is because of the very low reaction rate which results from this change in the activation energy. This result substantiates the earlier conclusion of the insignificance of this reaction on the particular data set which was made when Arrhenius constant of this reaction is set to zero i.e. the reaction is removed.

Table 4.26 shows the effect of changes in the activation energy of the heavy oil oxidation reaction. It has been found that when the activation energy of the original case study was doubled, ignition did not occur. This result is in line with the previous results obtained when the Arrhenius constant of this reaction was set to zero.

The increase in the average peak temperature with the decrease in the activation energy is due to the increase in the rate of this reaction. The increase in the temperature is also shown in figure 4.51 which presents the temperature profile of a typical block from the reservoir (grid block three). Also, the increase in the reaction rate leads to a reduction in the amount of the heavy oil needed to be cracked, so more oil was available which needed a longer production time for full recovery.

Since the heat loss term shown in table 4.26 is represented as a ratio between the heat loss and the total heat generated by the reactions, the decrease shown in this term with the decrease in the activation energy is due to the increase in the reaction rate, hence an increase in the total heat generated by the reactions.

When the activation energy of the heavy oil cracking reaction is changed from 66.99 kJ/gmol to 100 and 133.98 kJ/gmol, the results obtained are shown in table 4.27. These changes have the same result because the rate of this reaction has become slower than the rate of the heavy oil oxidation reaction of which the activation energy is 77.46 kJ/gmol. As a result of this, the heavy oil oxidation reaction becomes the dominant reaction which provides the heat for the combustion.

Table 4.28 shows the effect of the changes in the activation energy of the coke oxidation reaction. This table shows that when the activation energy of this reaction is changed from 54.43 to 108.86 kJ/gmol, the maximum amount of oil which can be recovered is decreased. This is due to the dominance of the heavy oil oxidation reaction of which the activation energy is 77.46 kJ/gmol. This dominance means that more oil needs to be oxidized in order to provide the heat needed to sustain the combustion. This also explains the increase in the water recovery since water is one of products of the heavy oil oxidation reaction. The decrease in the heat loss to the surrounding strata is due to the lower overall temperatures resulting in a smaller driving force.

Table 4.28 also shows the results of the change in the activation energy from 54.43 to 27.265 kJ/gmol. This reduction means that the coke will be oxidized at lower temperatures and more oil is needed to be cracked to provide sufficient heat to sustain the combustion which explains the drop in the amount of oil that can be recovered. This decrease leads to a lower time needed to achieve maximum recovery. The decrease in the heat loss is also due to the lower overall temperatures.

Table 4.25 The Effect Of The Activation Energy

Light Oil Oxidation Reaction			
VARIABLES	Activation Energy (kJ/mol)		
	154.92	77.46*	38.73
WATER RECOVERY (% OF OWIP)	116.9	117.4	120.6
TOTAL OIL RECOVERY (% OOIP)	94.1	95.3	94.5
LIGHT OIL RECOVERY (% OOIP)	26.0	26.3	21.2
AVERAGE PEAK TEMPERATURE (K)	621.3	622.0	690.0
MAXIMUM PRESSURE (kPa)	1118.4	1242.0	1226.3
TOTAL PRODUCTION TIME (day)	146.2	145.1	158.2
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	24.9	25.0	28.7

* Original case study

Table 4.26 The Effect Of The Activation Energy

Heavy Oil Oxidation Reaction			
VARIABLES	Activation Energy (kJ/mol)		
	154.92	77.46*	38.73
WATER RECOVERY (% OF OWIP)	-----	117.4	126.2
TOTAL OIL RECOVERY (% OOIP)	-----	95.3	95.6
LIGHT OIL RECOVERY (% OOIP)	-----	26.3	21.3
AVERAGE PEAK TEMPERATURE (K)	-----	622.0	773.6
MAXIMUM PRESSURE (kPa)	-----	1242.0	1573.9
TOTAL PRODUCTION TIME (day)	-----	145.1	180.8
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	-----	25.0	23.0

* Original case study

Table 4.27 The Effect Of The Activation Energy

Heavy Oil Cracking Reaction			
VARIABLES	Activation Energy (kJ/mol)		
	133.98	100.00	66.99 *
WATER RECOVERY (% OF OWIP)	122.0	122.0	117.4
TOTAL OIL RECOVERY (% OOIP)	84.9	84.6	95.3
LIGHT OIL RECOVERY (% OOIP)	00.0	0.03	26.3
AVERAGE PEAK TEMPERATURE (K)	598.7	598.2	622.0
MAXIMUM PRESSURE (kPa)	1493.5	1560.6	1242.0
TOTAL PRODUCTION TIME (day)	133.2	133.7	145.1
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	26.2	26.1	25.0

* Original case study

Table 4.28 The Effect Of The Activation Energy

Coke Oxidation Reaction			
VARIABLES	Activation Energy (kJ/mol)		
	108.86	54.43*	27.265
WATER RECOVERY (% OF OWIP)	123.1	117.4	113.4
TOTAL OIL RECOVERY (% OOIP)	87.0	95.3	91.9
LIGHT OIL RECOVERY (% OOIP)	24.5	26.3	25.9
AVERAGE PEAK TEMPERATURE (K)	563.1	622.0	571.3
MAXIMUM PRESSURE (kPa)	959.9	1242.0	1092.6
TOTAL PRODUCTION TIME (day)	140.4	145.1	129.4
HEAT LOSS (% OF TOTAL HEAT OF REACTIONS)	23.3	25.0	21.9

* Original case study

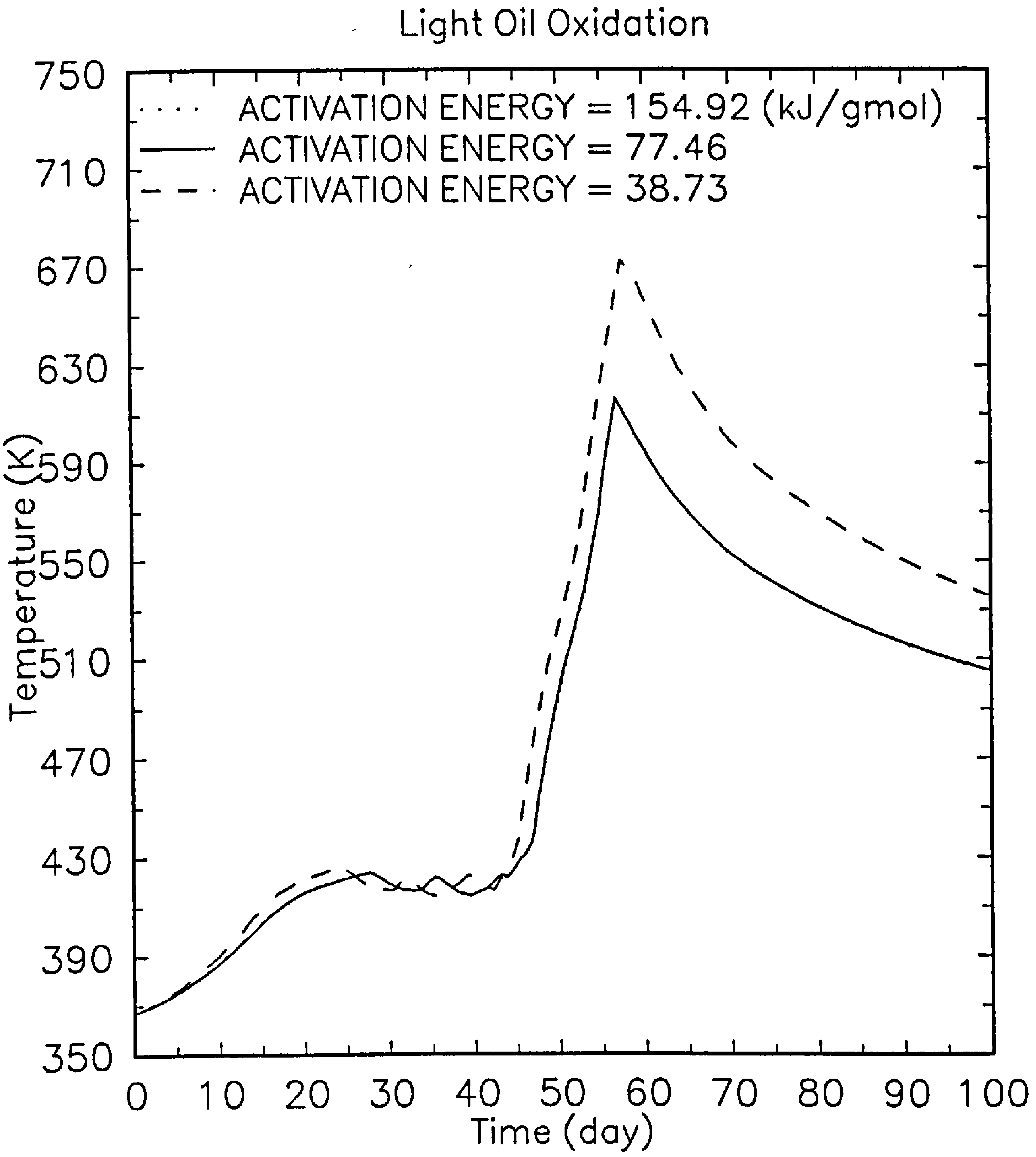


Figure 4.50 Temperature Profile Of Grid Block 3

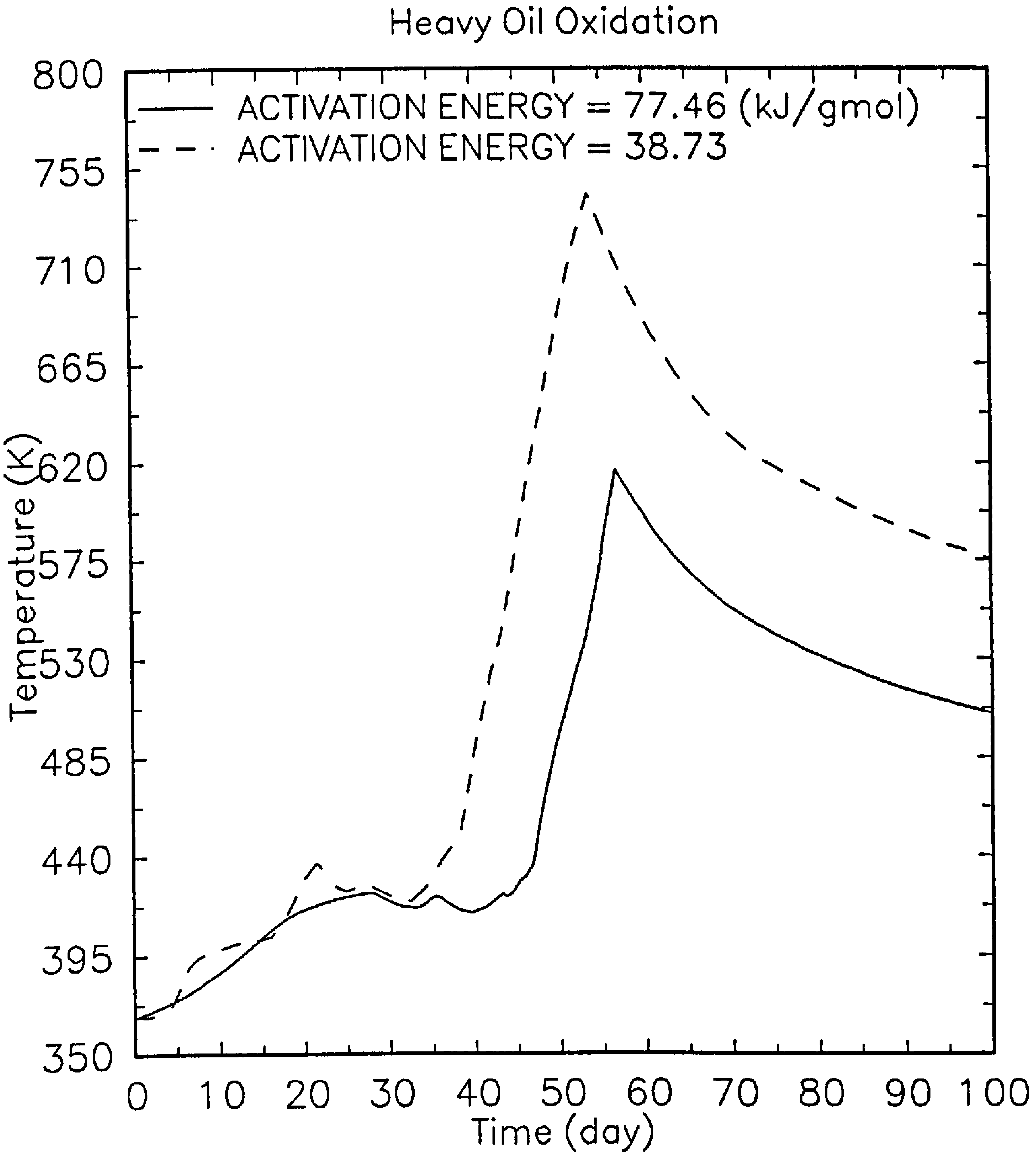


Figure 4.51 Temperature Profile Of Grid Block 3

4.7 Summary

The effects of the parameters studied in the previous sections can be summarized as follows:

1. The flow rate and oxygen concentration of the injected gas are very important factors in controlling the time needed to achieve maximum recovery and the system pressure. Also, oxygen concentration of the injected gas can dramatically effect the peak temperatures.

2. Although the changes made in the oil relative permeability are very small, it can be concluded that oil relative permeability affects the time needed to achieve maximum recovery as well as the peak temperatures and the system pressure.

3. The performance characteristics of in-situ combustion (except for pressure) are found to be insensitive to water and gas relative permeability.

4. Absolute permeability is found to affect the light oil recovery, the time needed to achieve maximum recovery, and the system pressure.

5. Since the aim of in-situ combustion is to reduce the oil viscosity and increase its mobility, oil viscosity is found to be very important in determining the amount of oil that can be recovered as well as the time needed to achieve maximum recovery. It was also found to affect the peak temperature and the system pressure.

6. The amount of fluids in a reservoir is directly proportional to the

porosity of the formation. Therefore, the porosity has been found to effect the amount of oil which can be recovered as well as the time needed to achieve maximum recovery. It is also found that the porosity dramatically affects the peak temperature.

7. It has been found that heat conduction in the radial direction is more significant than that in the axial direction. Therefore, thermal conductivity is found to be important in controlling the amount of heat lost to the surrounding strata and consequently the peak temperatures.

8. Reaction enthalpies as well as the kinetic parameters determine the amount of heat generated by the reactions which in turn affects the amount of oil that can be recovered and the time needed to achieve maximum recovery. Also, the heat generated by the reactions controls the overall temperatures of the formation.

9. Stability problems have not been encountered with the program during the changes made in the input papameters studied in the previous section.

10. The model conclusions are consistent with previously reported literature results.

Chapter Five

MULTI-DIMENSIONAL DEVELOPMENT

5.1 INTRODUCTION

Serious problems in many field combustion projects are caused by viscous fingering or gravity override of the injected gas. These are caused by the large differences in the viscosities and densities of the injected gas and oil and reservoir heterogeneities. The effect of these phenomena can be studied in a multi-dimensional model. Also, a multi-dimensional model can be used to study and optimize well positioning as well as to provide a more detailed picture of what is happening in a complex process like in-situ combustion.

The present 1-D model has been extended to the multi-dimensional case in order to study the above factors. Unfortunately, due to the lack of any detailed experimental or numerical multi-dimensional case study, the model could not be validated. Therefore, such a study is limited unless the model can be fully validated and its result is acceptable. However, three cases are presented in this chapter to provide a general picture of the performance of in situ combustion in a detailed multi-dimensional flow.

It must be emphasized that some problems have arisen in a number of the cases studied resulting from the way the Jacobian matrix is structured as well as the method used to solve it. Unfortunately, due to the relatively large amount of time needed to run the three-dimensional program (compared with the one-dimensional) and the limited amount of CPU time allowed for the project these problems could not be overcome.

No conclusions have been made regarding the production rate of these cases since they were terminated before all the oil in the reservoir is recovered.

5.2 Case One: Two-dimensional run with oxygen injection from the bottom layer.

This case is presented in order to demonstrate the significance of the position from which the gas is injected and the influence of that position on the combustion front behavior.

The reservoir is 50m long, 19.2m high, and 35m wide. It is divided to (4*3*1) grid blocks (4 divisions in the axial direction, 3 in the vertical direction, and undivided in the areal direction). Oxygen is injected from the bottom layer, grid block (1,1,1), at a rate of $2.72*10^5$ gmol/day. Production is from the last grid block of each layer, grid blocks (3,1,1), (3,2,1), and (3,3,1), with the well deliverable against 410 kPa backpressure. All the physical properties and kinetic data are the same as the one dimensional case study discussed in chapter three.

Figures 5.1, and 5.2 show the temperature, and oil saturation profiles for the four grid blocks at each layer. The temperatures and saturations of the middle and top layers were incremented by a fixed value in order to distinguish each layer.

These figures emphasize the importance of the injection position. It can be seen that the burning of grid block one in the middle layer occurred after 140 days. At this time, the combustion was about to develop in grid block four of the bottom layer whereas the top layer is still hardly effected.

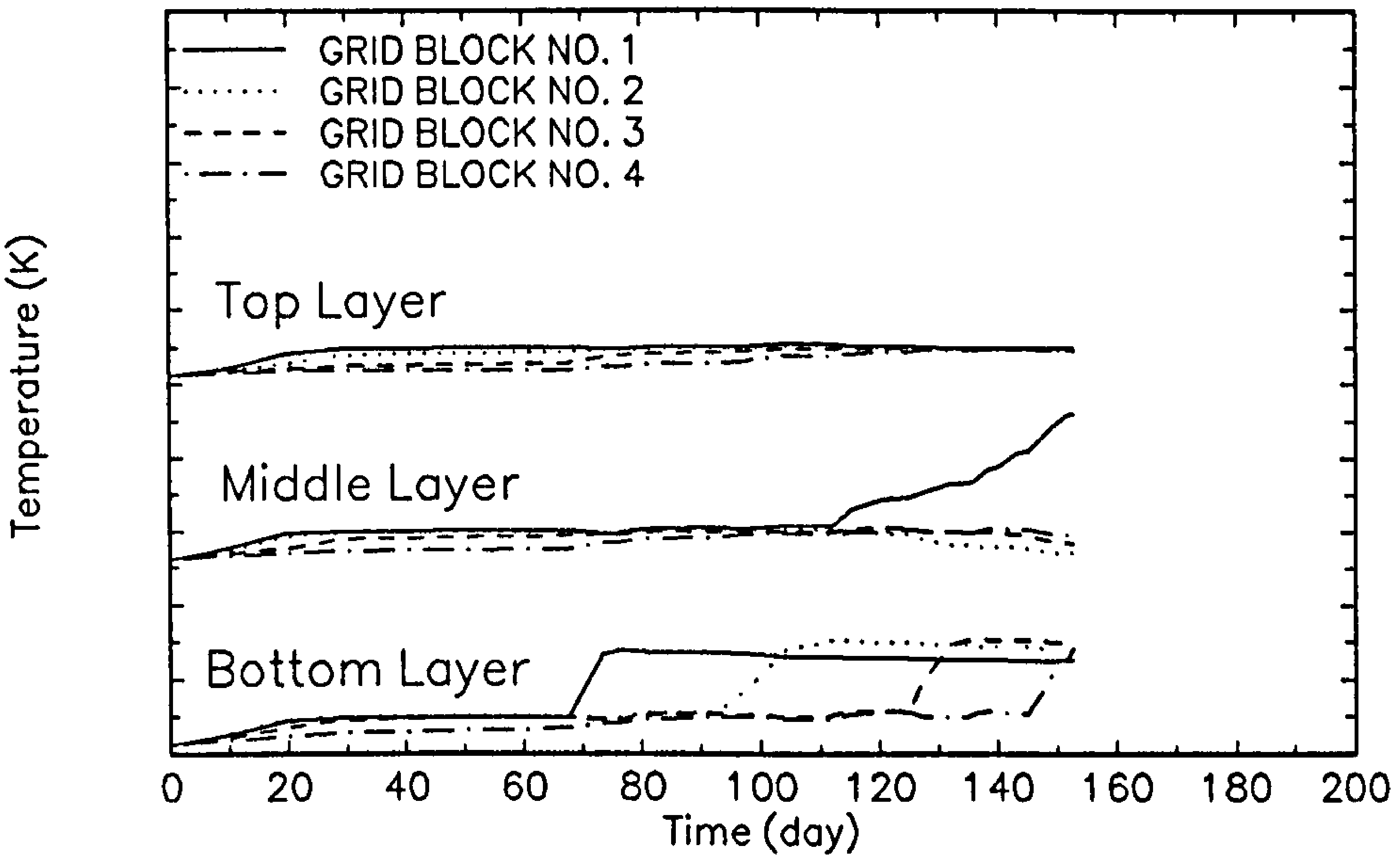


Figure 5.1 Temperature Profile Of The Reservoir (Bottom Layer Injection)

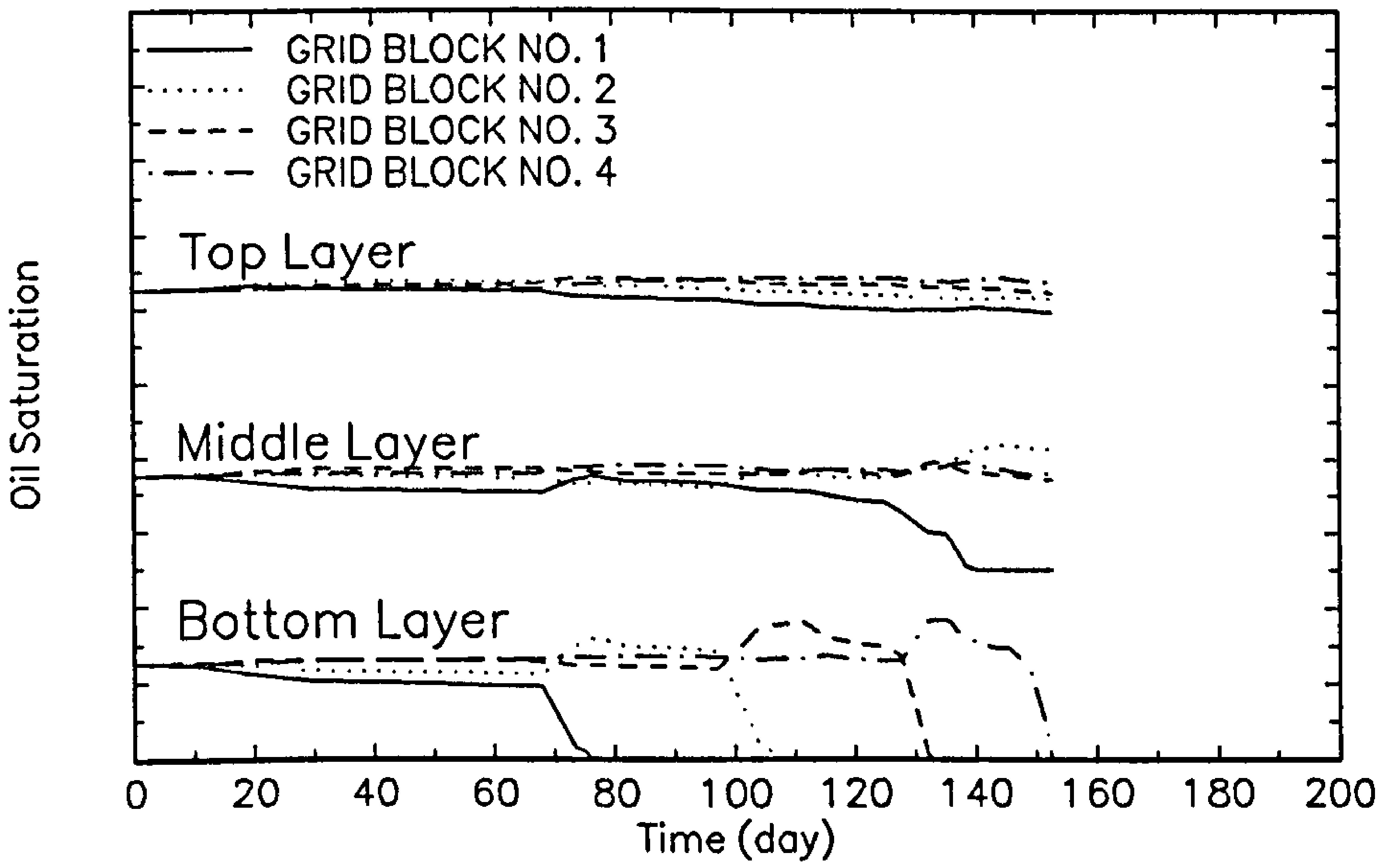


Figure 5.2 Oil Saturation Profile Of The Reservoir (Bottom Layer Injection)

5.3 Case Two: Two-dimensional run with oxygen injection into all three layers

This case demonstrates the effect of the gravity forces on the combustion front.

Oxygen is injected at a rate of $1.36*10^5$ gmol/day in each of the three layers. The remaining conditions are the same as the previous case.

Figures 5.3, and 5.4 show the temperature, and oil saturation profiles of the four grid blocks at each layer.

Due to the constant injection of the oxygen at all the layers, the convection forces between the layers are negligible since there is no pressure driving force. Therefore, only gravity forces are effective between the layers. This effect can clearly be seen in figure 5.~~4~~3 for grid block four where more time was needed to achieve combustion at this block than the above layers blocks.

Another case was attempted with oxygen injection into the top layer. The program failed to converge under this condition. A possible reason for the failure is the way the Jacobian matrix is structured and the solution method used to solve it. This problem can be solved be restructuring the Jacobian or adapting another solution method. The restructuring can be done by reversing the grid block numbering in the vertical direction and rewriting the flow equations for the new conditions.

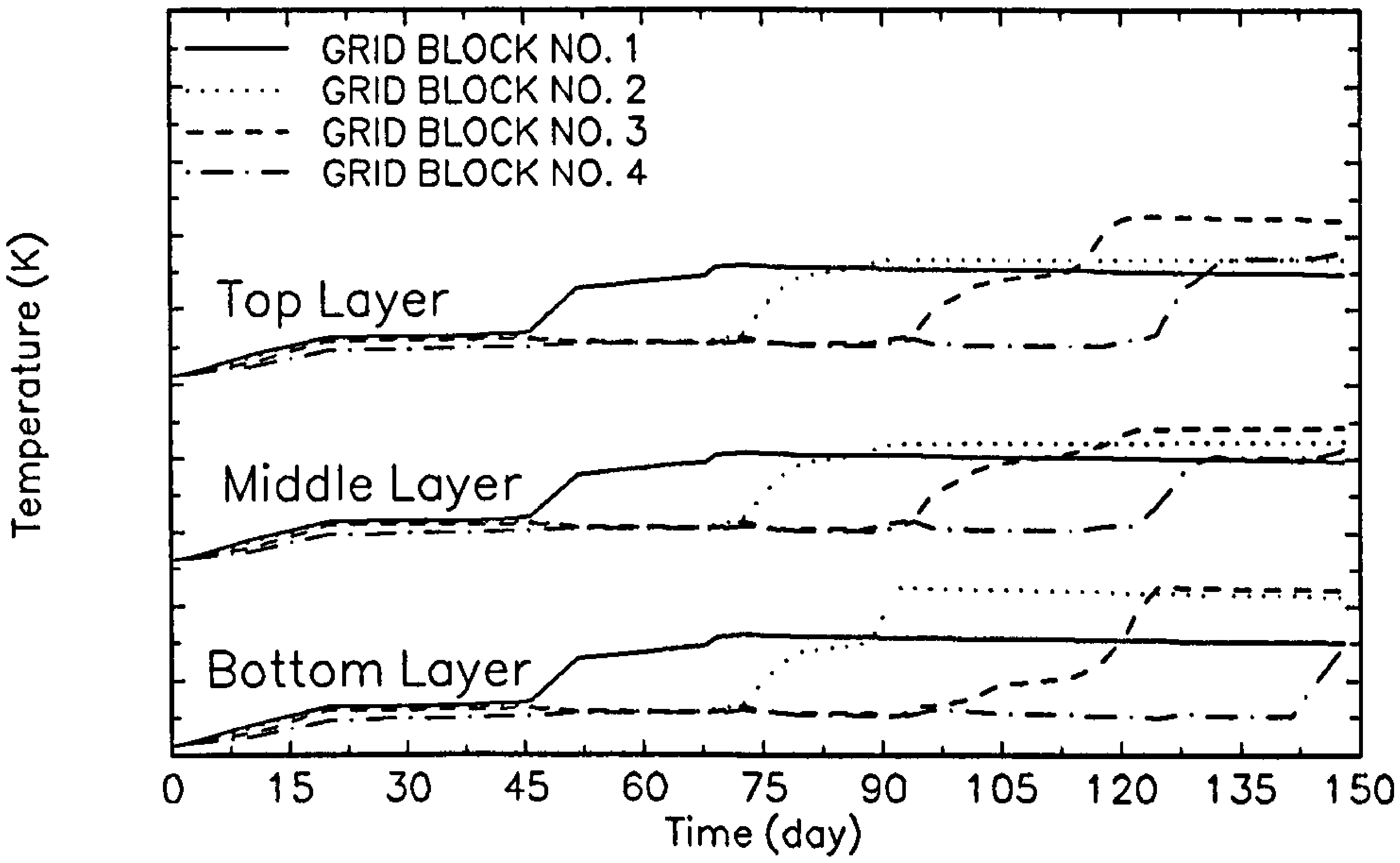


Figure 5.3 Temperature Profile Of The Reservoir (Three Layers Injection)

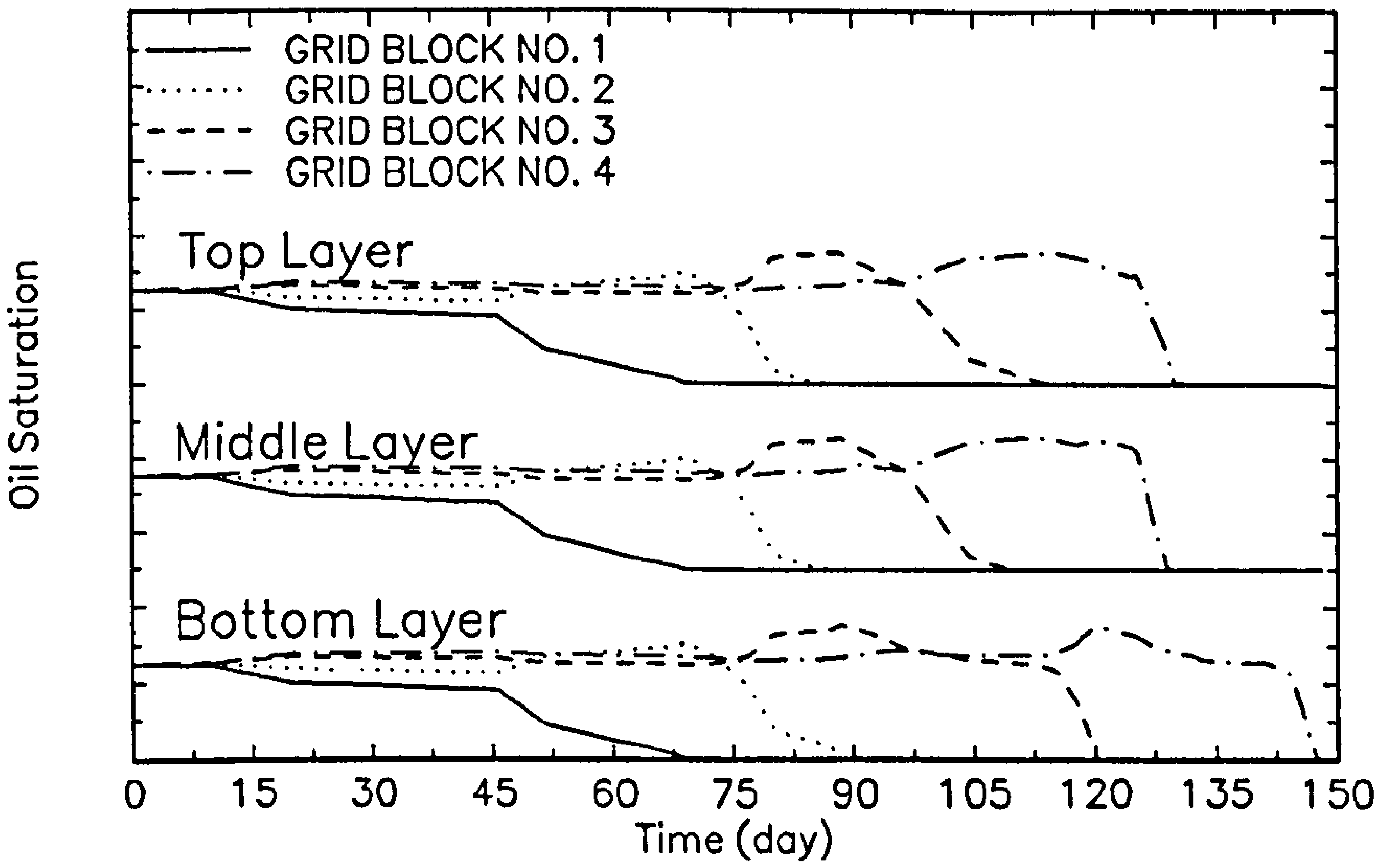


Figure 5.4 Oil Saturation Profile Of The Reservoir (Three Layers Injection)

5.4 Case Three: Three-dimensional Run

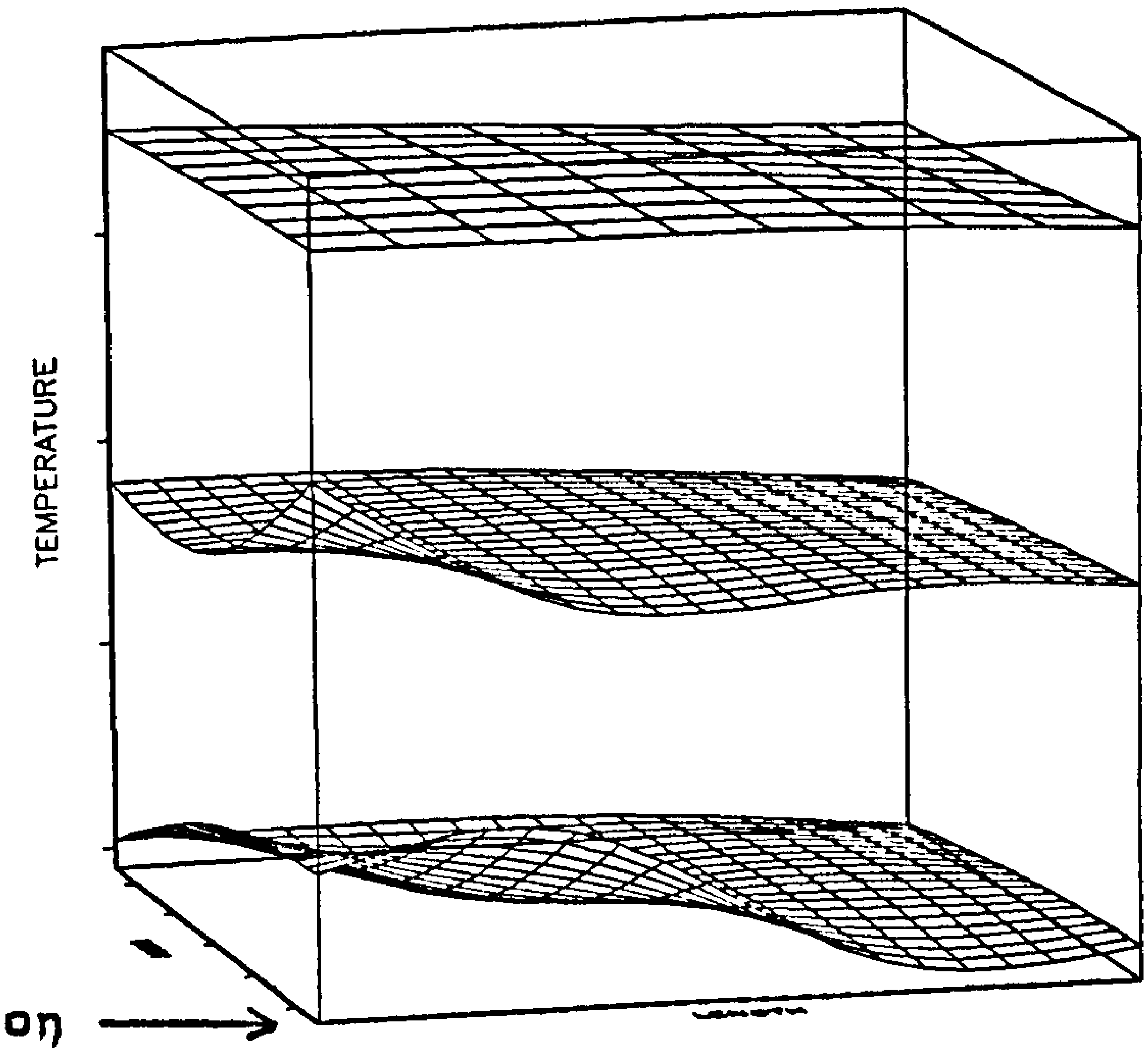
This case is presented in order to show the combustion front movement and the direction it takes.

The reservoir is divided into (4*3*3) grid blocks(4 divisions in the axial direction, 3 in the vertical direction, and 3 in the areal direction). Oxygen is injected at a rate of 1.36 10^5 gmol/day into grid block (1,1,1). Production is from all the last cells with the well deliverable against 410 kPa backpressure. The rest of the conditions are similar to the two-dimensional cases.

Figures 5.5-5.6 show the temperature profiles over the reservoir after 100 and 200 days from the beginning of oxygen injection, while figures 5.7-5.8 and 5.9-5.10 show the corresponding oil and water saturation profiles. In this case, the values of the variables at the middle and top layers have also been shifted by a fixed constant in order to distinguish each layer. The aim of presenting these figures is to show the shape of the profile. Therefore, in order to avoid any confusion, the scales on the axis have been removed.

It can be seen from these figures that for this relatively thick formation (comparing with its length), there is a pronounced time lag in both temperature and saturations profiles from the bottom to the top layer. This agrees with the conclusion made in the first of the two dimensional cases.

These figures also show that the combustion front advances in the primary flow direction (toward the production grid blocks) faster than that in the areal direction.



Injection Location

Figure 5.5 Temperature Profile Over The Reservoir
(After 100 Days)

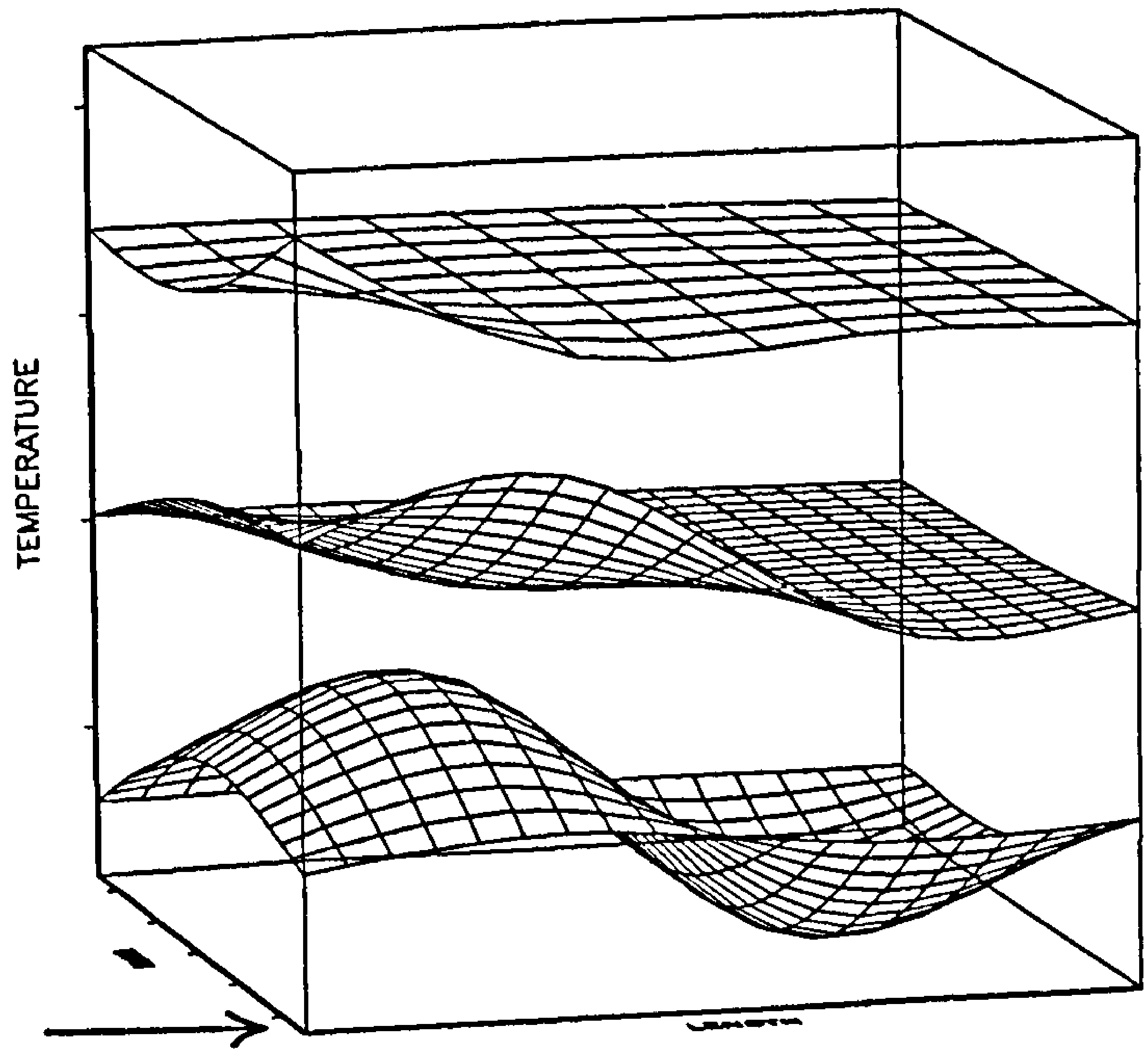


Figure 5.6 Temperature Profile Over The Reservoir
(After 200 Days)

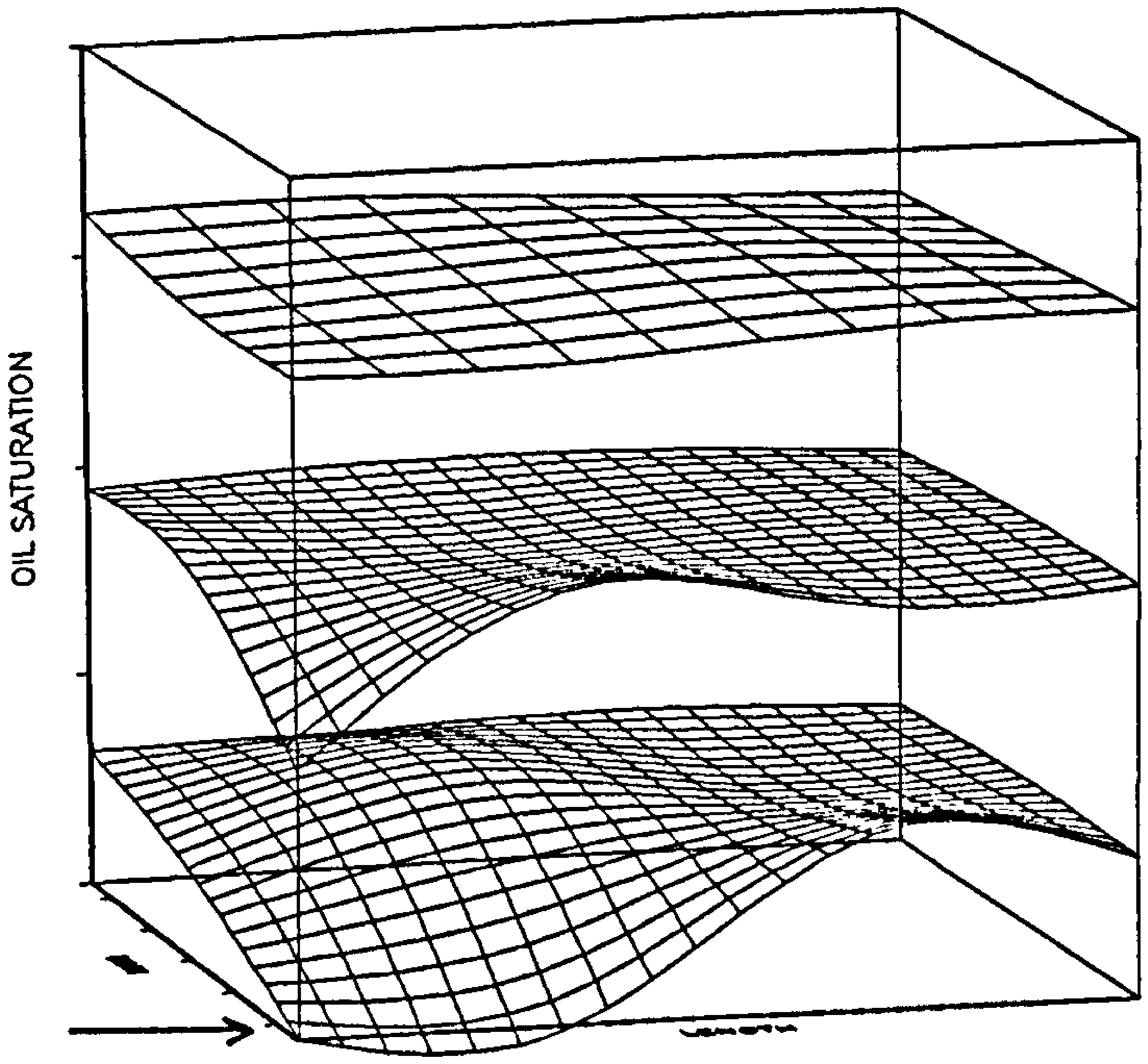


Figure 5.7 Oil Saturation Profile Over The Reservoir
(After 100 Days)

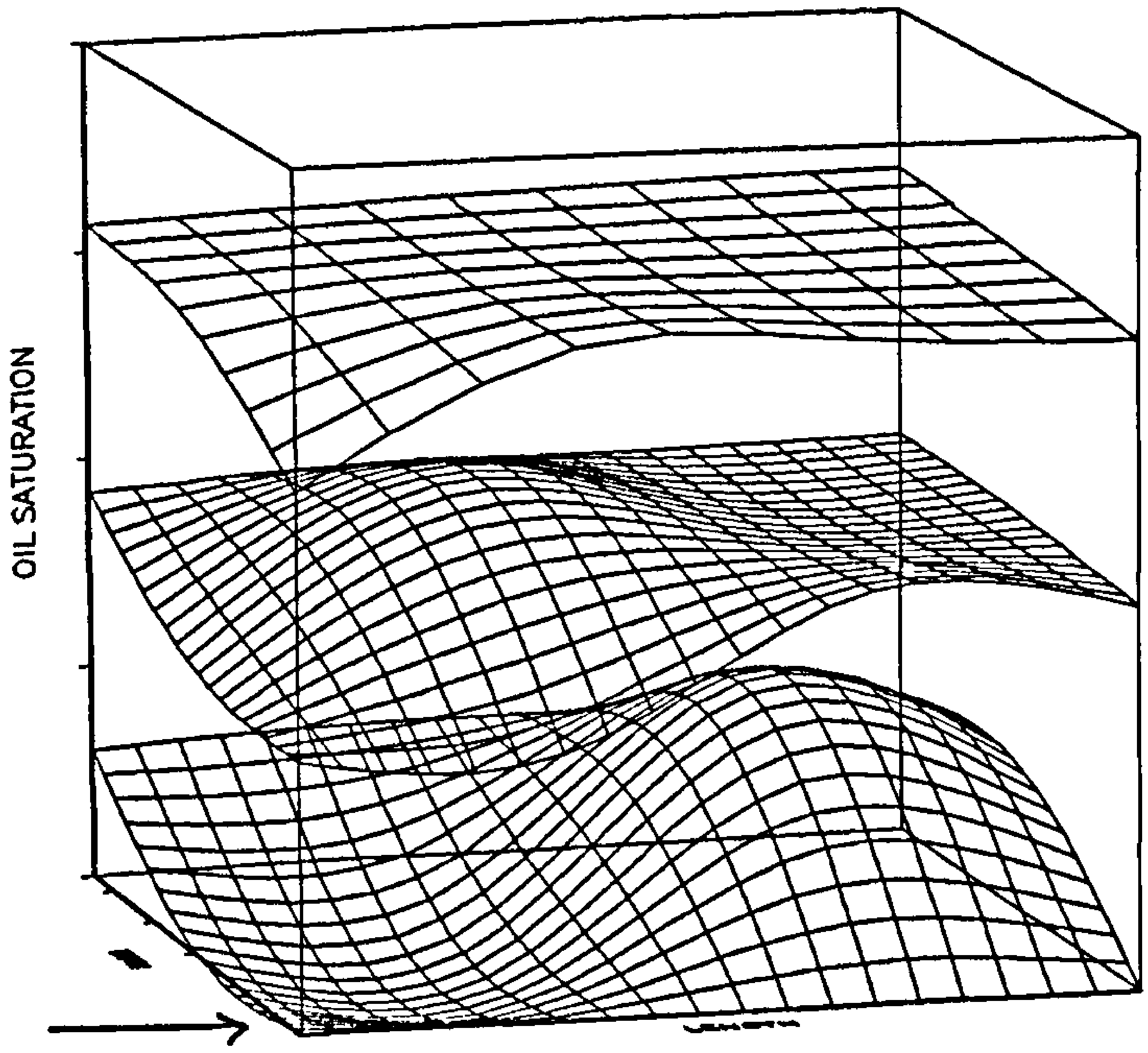


Figure 5.8 Oil Saturation Profile Over The Reservoir
(After 200 Days)

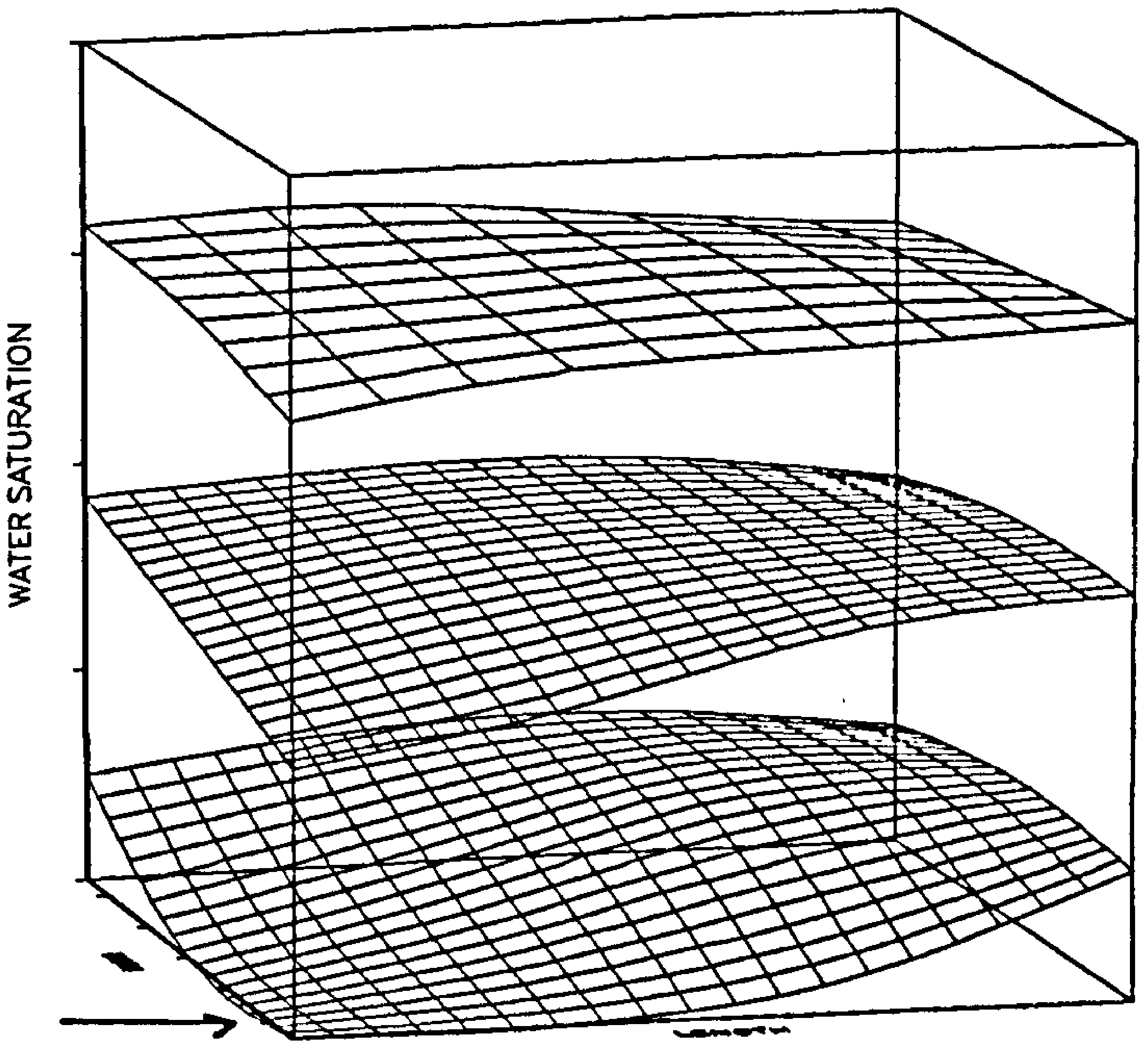


Figure 5.9 Water Saturation Profile Over The Reservoir (After 100 Days)

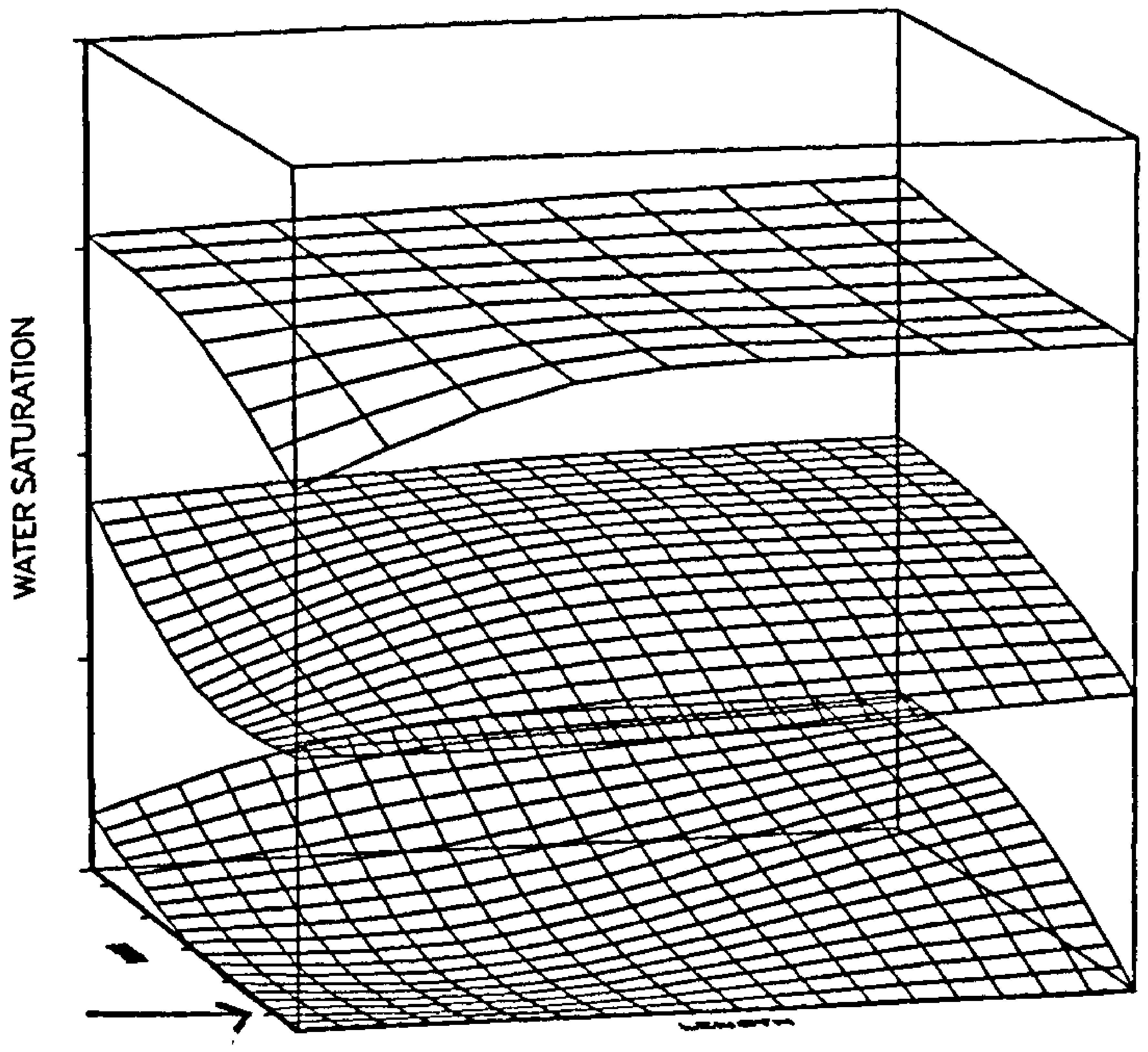


Figure 5.10 Water Saturation Profile Over The Reservoir (After 200 Days)

Chapter Six

SUMMARY, CONCLUSIONS, AND RECOMMENDATION FOR FUTURE WORK

6.1 Summary and Conclusions

Due to the complexity of the process of in-situ combustion, the development of the model has been conducted in the following stages;

Stage One: A one-dimensional simulator has been written. This simulator is a three phase flow simulator. It has six components (oxygen, inert gas, a light oil, a heavy oil, water, and coke). The inert gas consists of all the noncondensible gases other than the oxygen. The vaporization and condensation of both water and oil is governed by vapour-liquid equilibria, using temperature and pressure dependent equilibrium coefficients. Splitting of the oil into fractions is done to include the distillation effect.

The model includes four chemical reactions. These represent in turn the formation of coke from the heavy oil, the oxidation of the coke, and the oxidation of the light and the heavy oil. Heat transport is assumed to take place by convection and conduction. The effects of gravity, capillary pressure between any two fluid phases , and heat losses to the surrounding rock via conduction are also included.

Stage Two: The physical validity of the behavior of the main variables has been discussed. This has been achieved by running the simulator using data reported by Crookston et al [1979] and subsequently used by Grabowski et al [1979] and Vaughn [1986].

Stage three: The results of the previous stage has been compared with results of similar calculations performed using the ISCOM simulator. ISCOM (Grabowski et al [1979]) is a fully implicit multi-dimensional finite difference general thermal simulator developed by the computing modelling group (CMG) at Calgary. This study was needed in order to establish confidence in the predictive capability of the present model for a particular field. An excellent agreement was obtained between the two simulators.

Stage Four: A study of the grid size problem has been made and optimization of the number of grid blocks was done. It has been found that , for the specific case studied, the optimum number of grid blocks is ten.

Stage Five: A comprehensive study of the effect of the main input parameters on the performance characteristics of the in-situ combustion process has been carried out. This study has been made in order to indicate which data are "important" in the sense of having a significant influence on the performance characteristics of the process and to test the robustness of the program to changes in the input parameters.

The effect of the following parameters has been studied;

- Injection gas flow rate.
- Oxygen concentration in the injected gas.
- Relative permeability of oil,gas, and water.
- Absolute permeability.
- Oil viscosity.
- Porosity.

* Thermal conductivity.

* Heats of reactions.

* Reaction kinetics (Arrhenius constants and the activation energy.

Several conclusions can be drawn from this parametric study;

1. The flow rate and oxygen concentration of the injected gas are very important factors in controlling the time needed to achieve maximum recovery. System pressure was also affected. Oxygen concentration of the injected gas can dramatically effect the peak temperatures.

2. Oil relative permeability affects the time needed to achieve maximum recovery as well as the peak temperatures and the system pressure.

3. The performance characteristics (except for pressure) of in-situ combustion are found to be insensitive to water and gas relative permeability.

4. Absolute permeability is found to affect light oil recovery, the time needed to achieve maximum recovery, and the system pressure.

5. Oil viscosity is very important in determining the amount of oil that can be recovered as well as the time needed to achieve maximum recovery. It was also found to affect the peak temperature and the system pressure.

6. The porosity affects the amount of oil which can be recovered as

well as the time needed to achieve maximum recovery. It was also found that the porosity dramatically affects the peak temperature.

7. Thermal conductivity is found to be important in controlling the amount of heat lost to the surrounding strata and consequently the peak temperatures.

8. Reaction enthalpies as well as the kinetic parameters determine the amount of heat generated by the reactions which in turn affects the amount of oil that can be recovered and the time needed to achieve maximum recovery. Also, the heat generated by the reactions controls the overall temperatures of the formation.

Stage Six: The model has been developed to include multi-dimensional flow. This development was necessary in order to study the effect of viscous fingering and gravity override in field combustion projects. Also, such a model can be used to study and optimize well positioning as well as to provide a more detailed picture of what is happening in a complex process like in-situ combustion.

The multi-dimensional model requires a considerably larger amount of CPU time compared with the one-dimensional model. Access to Manchester university supercomputer was limited by the SERC allocation for the project. Thus, there was a restriction on the number of cases to which the multi-dimensional model to be applied. But nevertheless the results obtained from the cases presented demonstrated the importance of the injection location in a given reservoir formation.

6.2 Recommendation for Future Work

The one-dimensional model could be developed further into a more general form in which any number of components can be used as well as any number of reactions. Also, the model can be developed to accept any form of correlations or tabulation to calculate the physical properties. By this means, any data taken from experiments can be used directly in the model.

In addition, the importance of the grid size in field scale simulation needs to be studied comprehensively and an efficient method to tackle this problem needs to be adapted and added to the simulator.

The problems arising in the multi-dimensional model need to be studied comprehensively and the methods used to construct and solve the Jacobian matrix must be modified in order to solve these problems.

In addition, the multi-dimensional model should be developed into a more general form to include any number of components and reactions. This model can further be modified to work in the radial and curvilinear dimensions.

Work should be done to improve the efficiency of the model. This can be done by adopting faster solution techniques.

REFERENCES

Acharya, U. K.: "Theoretical study of In-Situ Combustion in Thick Inclined Oil Reservoirs.", *SPE paper no. 7967*, presented at the 1979 california Meeting of the SPE of AIME held in Ventura, California, April 18-20, 1979.

Adler, G.: "A Linear Model and a Related Very Stable Numerical Method for Thermal Secondary Oil Recovery", *J. Can. Pet. Tech.*, (July-Sept., 1975)56-65.

Anis, M., Hwang, M. K., and Odeh, A. S.: "A Sensitivity Study on the Effect of Parameters on Results From an In-Situ Combustion Simulator", *Soc. Pet. Eng. J.*, (April 1983), 259-264.

Bailey, H. R., and Larkin, B. K.: "Heat Conduction in Underground Combustion", *Trans., AIME*, (1959) Vol 216, 123.

Bailey, H. R., and Larkin, B. K.: "Heat Convection in Underground Combustion", *Trans., AIME*, (1960) Vol 216, 312.

Baker, P. B.: "Temperature Profile in Underground Combustion", *Soc. Pet. Eng. J.*, (March 1962)21.

Belgrave, J. D. M., Moore, R. G. Ursenbach, M. G., and Bennion, D. W.: "A Comprehensive Approach to In-Situ Combustion Modeling", Paper presented for the SPE/DOE Seventh Symposium on Enhanced Oil Recovery held in Tuisa, Oklahoma, April 22-25, 1990.

Camey, J. P., and Emanuel, A. S.: "Effect of Grid Size in the Compositional Simulation of CO_2 Injection", *SPE paper no. 6894,* presented at the SPE Annual Conference and exhibition held in Denver, CO, 1977.

Chu, C.: "Two-Dimensional Analysis of Radial Heat Wave",*J. Pet. Tech.,* (Oct. 1963)1137.

Coats, K. H.: "In-Situ Combustion Model", *Soc. Pet. Eng. J* (Dec., 1980)533.

Couch, E. J., and Rodriguesz, H. V.: "Effects of Porosity and Permeability on In-Situ Combustion Fuel Consumption.", *SPE paper no. 2873,* presented at the 45th Annual Fall Meeting of SPE, Houston, 1970.

Crookston, R. B., Culham, W. E., and Chen, W. H.: "Numerical Simulation Model for Thermal Recovery Process", *Soc. Pet. Eng. J.,*(Feb.,1979)37-58.

Crout, P. D.: "A Short Method for Evaluating Determinants and Solving Systems of Linear Equations With Real or Complex Coefficients", *Trans. AIEE,* (1941), 60, 1235.

Davies, R.: "Numerical Simulation of In-Situ Combustion", Ph.D. Thesis, University of Bath, 1988.

Eggenschwiler, M. E., and Farouq Ali, S. M.: "Two-Dimensional, Single Phase Simulation of Fireflood", Can-Venezuela Oil Sands 1977.

El-Khatib, N.: "A Mathematical Model of Thermal Oil Recovery in a Forward In-Situ Combustion Process", Ph.D Thesis, Univ. of Pittsburgh (1973).

Ewing, R.E.: " Efficient Adaptive For Fluid Flow Applications", *Comp. Meth. Appl. Mech. Eng.* 55(1986), 89-103.

Ewing, R. E., and Lazarov, R. D.: "Adaptive Local Grid Refinement", *SPE paper No. 17806,* presented at the SPE Rocky Mountain Regional Meet., held in Casper, WY, 1988.

Farouq Ali, S. M.: "Multiphase, Multidimensional Simulation of In Situ Combustion", *Paper SPE 6896,* presented at the 52nd Annual Fall Meeting of SPE, Denver, Oct.9-12, 1977.

Genrich, J. F., and Pope, G. A.: "A Simplified Performance - Predictive Model for In-Situ Combustion Process", *SPE Res. Eng.,*(May, 1988)410.

Gottfried, B. S.: "A Mathematical Model Of Thermal Oil Recovery in Linear Systems", *Soc. Pet. Eng. J.,* (September 1965),196-210.

Grabowski, J. W., Vinsome, P. K., Behie, A., and Rubin, B.: "A Fully Implicit General Purpose Finite Difference Thermal Model for In-Situ Combustion and Steam", *SPE paper No. 8396,* presented at the 54th Annual Fall Meeting of SPE, Las Vegas, Sep. 23-26, 1979.

Han, D. K., Han, D. L., Yan, C. Z., and Peng, L. T.: "A More Flexible Approach of Dynamic Lacal Refinement for Reservoir Modeling", *SPE* *paper No. 16014,* presented at the Ninth SPE Symp. on Res. Sim. held in San Antonio, Texas, 1987.

Heinemann, Z. E., Gerken, G., and von Hantelmann, G.: "Using Grid Refinement in a Multiple-Application Reservoir Simulator", *SPE paper No. 12255,* presented at the Reservoir Simulation Symp. held in San Francisco, CA, 1983.

Ito, Y., and Chow, A. K.: "A Field Scale In-Situ Combustion Simulator with Channeling Considerations", *SPE Res. Eng.* (May 1988), 419-430.

Hwang, M. K., Jines W. R., Odeh A. S.: "An In-Situ Combustion Process Simulator", *SPE paper 9450,* 55th Annual SPE Fall Conference, Dallas, 1980.

Latil, M., Bardon, C., Burger, J., and Sourieau, P.: "Enhanced Oil Recovery", Edition Techinp, Paris (1980).

Lin, C. Y., Chen, W. H., Lee, S. T., and Culham, W. E.: "Numerical Simulation of Combustion Tube Experiments and the Associated Kinetics of In-Situ Combustion Processes", *Soc. Pet. Eng. J,* December (1984) 657-666.

Ramey, H. J., Jr: "Transient Heat Conduction During Radial Movement Of Cylindrical Heat Source - Applications Of the Thermal Recovery Process", *Trans. AIME,* (1959)Vol 216,75.

Rubin, B., and Buchannan, W. L.: "A General Purpose Thermal Model", *Soc. Pet. Eng. J.,* (April 1985), 202-214.

Rubin, B., and Vinsome, P. K. W.: "The Simulation of the In-Situ Combustion Process in One Dimensional Using a Highly Implicit Finite Difference Scheme", *J. Can. Pet. Tech.,* (Oct.-Dec.,1980)68-76.

Smith, J. t., and Farouq Ali, S. M.: "Simulation of In-Situ Combustion in a Two-Dimensional System", Paper 3594, presented at the 46th Annual Fall Meeting of SPE, New Orleans, October 1971.

Thies, P. L., and Lemmonier, P.: "An In-Situ Combustion Reservoir Simulator with a New Representation of Chemical Reactions", *SPE paper no. 17416,* presented at the SPE California Regional Meeting held in Long Beach, CA, 1988.

Thomas, G. W.: " A Study of Forward Combustion in a Redial System Bounded by Permeable Media", *J. Pet. Tech.,* (October, 1963)1145.

Vaughn, P.: " A Numerical Model for Thermal Recovery Process In Tar Sand", Report no. DOE/FE/60177-2219, April 1986.

Vinsome, P. K. W., and Westerveld, J.: "A Simple Method for Predicting Cap and Base Rock Heat Losses in Thermal Reservoir Simulators", *J. Can. Pet. Tech.* (July-September 1980),87-90.

Watson, K. M.: "Thermodynamic of the Lquid State", *Ind. and Eng. Chem.,* 35,(1943), 393-406.

Wilson, L. A., Wygal, R. J., Reed, D. W., Gergins R. L., and Henderson, J. H.: "Fluid Dynamics During an Underground Combustion Process", *Trans., AIME,* (1958)Vol 213,146.

Youngren, G. K.: "Development and Application of an In-Situ Combustion Reservoir Simulator", *Soc. Pet. Eng. J.,* (Feb., 1980)39-51.

NOMENCLATURE

A_A	Arrhenius constant in the rate equation of light oil oxidation	$Day^{-1}kPa^{-1}$
A_B	Arrhenius constant in the rate equation of heavy oil oxidation	$Day^{-1}kPa^{-1}$
A_C	Arrhenius constant in the rate equation of coke formation	Day^{-1}
A_D	Arrhenius constant in the rate equation of coke oxidation	$Day^{-1}kPa^{-1}$
C_c	Coke Concentration	$gmol/m^3$
C_{cmax}	Maximum coke concentration	$gmol/m^3$
Cp_c	Coke heat capacity	kJ/gmol.K
Cp_g	Gas phase heat capacity	kJ/gmol.K
Cp_o	Oil phase heat capacity	kJ/gmol.K
Cp_r	Rock heat capacity	kJ/gmol.K
Cp_w	Water phase heat capacity	kJ/gmol.K
E_A	Activation energy of the light oil oxidation reaction	kJ/gmol
E_B	Activation energy of the heavy oil oxidation reaction	kJ/gmol
E_C	Activation energy of the coke formation reaction	kJ/gmol
E_D	Activation energy of the coke oxidation reaction	kJ/gmol
h	Thickness	m
h_j	Phase j enthalpy (j=g,w,o,c,r)	kJ/gmol
h_j^*	Phase j injection/production enthalpy (j=o,g,w)	kJ/gmol

K	Absolute Permeability	m^2
K_{rg}	Gas relative permeability	--
K_{rw}	Water relative permeability	--
K_{ro}	Oil relative permeability	--
K_{rog}	Oil-gas two phase relative permeability to oil	--
K_{row}	Oil-water two phase relative permeability to oil	--
K_{rgro}	Oil-gas two phase relative permeability to water at residual oil saturation	--
K_{rwro}	Oil-water two phase relative permeability to gas at residual oil saturation	--
K_{rocw}	Oil-water two-phase relative permeability to oil at connate water saturation	--
K_j	Equilibrium constant (j=3,4,5)	--
K_T	Thermal conductivity of the formation	kJ/Day.m.K
P_{cg}	Gas/oil capillary pressure	kPa
P_{cw}	Oil/water capillary pressure	kPa
P_g	Gas phase pressure	kPa
P_o	Oil phase pressure	kPa
P_w	Water phase pressure	kPa
P_{well}	Well pressure	kPa
q_j	Injection/production rate (j=o,w,g)	$gmol/m^3Day$
q_i	Component injection/production rate (i=1,2,3,4,5)	$gmol/m^3Day$
r_A	Reaction rate of light oil oxidation	$gmol/m^3Day$
r_B	Reaction rate of heavy oil oxidation	$gmol/m^3Day$
r_C	Reaction rate of coke formation	$gmol/m^3Day$
r_D	Reaction rate of coke oxidation	$gmol/m^3Day$

R	Standard gas constant	kJ/gmol.K
s_j	Reactions stoichiometric coefficients (j=1,2,...,12)	--
S_g	Gas phase saturation	--
S_o	Oil phase saturation	--
S_w	Water Phase saturation	--
S_{wc}	Connate water saturation	--
S_{gc}	Critical gas saturation	--
S_{org}	Residual oil saturation for oil-gas system	--
S_{orw}	Residual oil saturation for oil-water system	--
t	Time	Day
T	Temperature	K
U_j	Phase enternal energy (j=o,w,g,r,c)	kJ/gmol
V	Grid block volume	m^3
V_g	Gas phase velocity	m/Day
V_o	Oil phase velocity	m/Day
V_w	Water phase velocity	m/Day
x	Spatial variable	m
X_3	Light oil mole fraction in the oil phase	--
X_4	Heavy oil mole fraction in the oil phase	--
y	Spatial variable	m
Y_j	Component j mole fraction in the gas phase (j=1,2,....,5)	--
z	Spatial Variable	m
Z	Compressibility factor	--
Z_i	Fitting constants in the relative permeability equations i=g,w,og,ow	--
α	Productivity index	
α_o	Oil compressibility factor	kPa^{-1}

α_w	Water compressibility factor	kPa^{-1}
β_o	Oil thermal expansion	K^{-1}
β_w	Water thermal expansion	K^{-1}
ε	Constant in the saturation function of the equilibrium constants equations	——
ϕ	Porosity	——
ρ_j	Phase density (j=g,o,w,r)	$gmol/m^3$
ρ_{oR}	Oil dansity at reference pressure and temperature	$gmol/m^3$
ρ^3_{oR}	light oil dansity at reference pressure and temperature	$gmol/m^3$
ρ^4_{oR}	Heavy oil dansity at reference pressure and temperature	$gmol/m^3$
ρ_{ow}	Water dansity at reference pressure and temperature	$gmol/m^3$
μ_j	Phase viscosity (j=o,g,w)	kPa Day
μ_{o3}	Light oil viscosity	kPa.Day
μ_{o4}	Heavy oil viscosity	kPa.Day
ω	Damping factor in the RHS of Newtons equation	--

APPENDIX A

ELIMINATION OF COKE EQUATION

The coke conservation equation is discretized to :

$$\frac{C_C^{n+1} - C_C^{n}}{\Delta t} = s_8 r_C^{n+1} - r_D^{n+1} \qquad \text{A.1}$$

$$r_C^{n+1} = [A_C e^{(-E_C/RT)} (\phi S_o \rho_o X_4)(1 - (\frac{C_C}{C_{Cmax}})^5)]^{n+1} \qquad \text{A.2}$$

$$r_D^{n+1} = [A_D \, e^{(-E_D/RT)} (\rho_g Y_1)(C_C)]^{n+1} \qquad \text{A.3}$$

Substitution of A.2 and A.3 in to A.1 yields:

$$\begin{aligned} C_C^{n+1} = C_C^{n} &+ [s_8 A_C e^{(-E_C/RT)}(\phi S_o \rho_o X_4)(1-(\frac{C_C}{C_{Cmax}})^5)]\Delta t \\ &- [\, A_D e^{(-E_D/RT)}(P_g Y_1)(C_C)]^{n+1}\Delta t \end{aligned} \qquad \text{A.4}$$

Equation A.4 is a nonlinear algebraic equation solved using the Newton Raphson method.

APPENDIX -B-

DESCRETIZATION OF THE CONSERVATION EQUATIONS

The six primary conservation equations are discretized using single point upstream weighting in space and a backward Euler scheme in time. The following is the result of the finite difference presentation after substituting the velocity equations in the conservation equations:

1- Inert Mass Balance Equation:

$$\left(\frac{\rho_g Y_2 KK_{rg}}{\mu_g}\right)_l^{n+1}\left(\frac{P_{gl+1}-P_{gl}}{\Delta x}\right)^{n+1}+\left(\frac{\rho_g Y_2 KK_{rg}}{\mu_g}\right)_{l-1}^{n+1}\left(\frac{P_{gl-1}-P_{gl}}{\Delta x}\right)^{n+1}$$

$$+\ q_2^{n+1}-(s_1r_A+s_5r_B+s_9r_C+s_{11}r_D)_l^{n+1}=\left(\frac{(\phi S_g\rho_g Y_2)_l^{n+1}-(\phi S_g\rho_g Y_2)_l^{n}}{\Delta t}\right)$$

B.1

2- Water Mass Balance Equation:

$$\left(\frac{\rho_g K_5 KK_{rg}}{\mu_g}\right)_l^{n+1}\left(\frac{P_{gl+1}-P_{gl}}{\Delta x}\right)^{n+1}+\left(\frac{\rho_g K_5 KK_{rg}}{\mu_g}\right)_{l-1}^{n+1}\left(\frac{P_{gl-1}-P_{gl}}{\Delta x}\right)^{n+1}$$

$$+\left(\frac{\rho_w KK_{rw}}{\mu_w}\right)_l^{n+1}\left(\frac{P_{wl+1}-P_{wl}}{\Delta x}\right)^{n+1}+\left(\frac{\rho_w KK_{rw}}{\mu_w}\right)_{l-1}^{n+1}\left(\frac{P_{wl-1}-P_{wl}}{\Delta x}\right)^{n+1}$$

$$+\ q_5^{n+1}-(s_3r_A+s_6r_B+s_{12}r_D)_l^{n+1}=\left(\frac{(\phi S_g\rho_g K_5)_l^{n+1}-(\phi S_g\rho_g K_5)_l^{n}}{\Delta t}\right)$$

$$+\left(\frac{(\phi S_w\rho_w)_l^{n+1}-(\phi S_w\rho_w)_l^{n}}{\Delta t}\right)$$

B.2

3- Light Oil Mass Balance Equation:

$$\left(\frac{\rho_g K_3 X_3 KK_{rg}}{\mu_g}\right)_l^{n+1}\left(\frac{P_{gl+1}-P_{gl}}{\Delta x}\right)^{n+1}+\left(\frac{\rho_g K_3 X_3 KK_{rg}}{\mu_g}\right)_{l-1}^{n+1}\left(\frac{P_{gl-1}-P_{gl}}{\Delta x}\right)^{n+1}$$

$$+\left(\frac{\rho_o X_3 KK_{ro}}{\mu_o}\right)_l^{n+1}\left(\frac{P_{ol+1}-P_{ol}}{\Delta x}\right)^{n+1}+\left(\frac{\rho_o X_3 KK_{ro}}{\mu_o}\right)_{l-1}^{n+1}\left(\frac{P_{ol-1}-P_{ol}}{\Delta x}\right)^{n+1}$$

$$+ q_3^{n+1} - (r_A - s_7 r_C)_l^{n+1} = \left(\frac{(\phi S_g \rho_g K_3 X_3)_l^{n+1} - (\phi S_g \rho_g K_3 X_3)_l^{n}}{\Delta t}\right)$$

$$+ \left(\frac{(\phi S_o \rho_o X_3)_l^{n+1} - (\phi S_o \rho_o X_3)_l^{n}}{\Delta t}\right) \qquad \textbf{B.3}$$

4- Energy Balance Equation:

$$\frac{K_T}{\Delta x^2}\left[(T_{l-1}-T_l)+(T_{l+1}-T_l)\right]^{n+1} + \Bigg\{\left(\frac{\rho_g h_g KK_{rg}}{\mu_g}\right)_{l-1}^{n+1}(P_{gl-1}-P_{gl})^{n+1}$$

$$+\left(\frac{\rho_g h_g KK_{rg}}{\mu_g}\right)_{l}^{n+1}(P_{gl+1}-P_{gl})^{n+1}+\left(\frac{\rho_w h_w KK_{rw}}{\mu_w}\right)_{l-1}^{n+1}(P_{wl-1}-P_{wl})^{n+1}$$

$$+\left(\frac{\rho_w h_w KK_{rw}}{\mu_w}\right)_{l}^{n+1}(P_{wl+1}-P_{wl})^{n+1}+\left(\frac{\rho_o h_o KK_{ro}}{\mu_o}\right)_{l-1}^{n+1}(P_{ol-1}-P_{ol})^{n+1}$$

$$+\left(\frac{\rho_o h_o KK_{ro}}{\mu_o}\right)_{l}^{n+1}(P_{ol+1}-P_{ol})^{n+1}\Bigg\}\frac{1}{\Delta x^2}$$

$$+ (q_g h_g + q_w h_w + q_o h_o)_l^{n+1} + (H_A r_A + H_B r_B + H_C r_C + H_D r_D)_l^{n+1} =$$

$$\frac{\left[(1-\phi)\rho_r U_r + \phi(S_g \rho_g U_g + S_w \rho_w U_w + S_o \rho_o U_o) + C_c U_c\right]_l^{n+1}}{\Delta t} -$$

$$\frac{[(1-\phi)\rho_r U_r + \phi(S_g \rho_g U_g + S_w \rho_w U_w + S_o \rho_o U_o) + C_c U_c]_l^n}{\Delta t} \qquad \text{B.4}$$

5- Oxygen Mass Balance Equation:

$$\left(\frac{\rho_g Y_1 K K_{rg}}{\mu_g}\right)_l^{n+1}\left(\frac{P_{gl+1} - P_{gl}}{\Delta x}\right)^{n+1} + \left(\frac{\rho_g Y_1 K K_{rg}}{\mu_g}\right)_{l-1}^{n+1}\left(\frac{P_{gl-1} - P_{gl}}{\Delta x}\right)^{n+1}$$

$$+\, q_1^{n+1} - (s_1 r_A + s_4 r_B + s_{10} r_D)_l^{n+1} = \left(\frac{(\phi S_g \rho_g Y_1)_l^{n+1} - (\phi S_g \rho_g Y_1)_l^n}{\Delta t}\right)$$

$$\text{B.5}$$

6- Heavy Oil Mass Balance:

$$\left(\frac{\rho_g K_4 X_4 K K_{rg}}{\mu_g}\right)_l^{n+1}\left(\frac{P_{gl+1} - P_{gl}}{\Delta x}\right)^{n+1} + \left(\frac{\rho_g K_4 X_4 K K_{rg}}{\mu_g}\right)_{l-1}^{n+1}\left(\frac{P_{gl-1} - P_{gl}}{\Delta x}\right)^{n+1}$$

$$+\left(\frac{\rho_o X_4 K K_{ro}}{\mu_o}\right)_l^{n+1}\left(\frac{P_{ol+1} - P_{ol}}{\Delta x}\right)^{n+1} + \left(\frac{\rho_o X_4 K K_{ro}}{\mu_o}\right)_{l-1}^{n+1}\left(\frac{P_{ol-1} - P_{ol}}{\Delta x}\right)^{n+1}$$

$$+\, q_3^{n+1} - (r_B + r_C)_l^{n+1} = \left(\frac{(\phi S_g \rho_g K_4 X_4)_l^{n+1} - (\phi S_g \rho_g K_4 X_4)_l^n}{\Delta t}\right)$$

$$+\left(\frac{(\phi S_o \rho_o X_4)_l^{n+1} - (\phi S_o \rho_o X_4)_l^n}{\Delta t}\right) \qquad \text{B.6}$$

APPENDIX C

C.1 Introduction

This Appendix describes the structure of the computer program written to model the in situ combustion process. The program is named **SUISCOM** for Salford University In Situ COMbustion. It is written totally in FORTRAN77. Efforts were made to exclude any extension commands to the standard FORTRAN77 commands. Therefore, the user should be able to run the program on any FORTRAN77 compiler without any modifications. The program has been tested on the following machines;

*Manchester University's Amdahl VP1200 super computer on its two FORTRAN77 compilers i.e. the vector and the scalar using the time allocated under the SERC grant.

*Salford University's Prime computer on its two FORTRAN77 compilers (F77 and FTN77).

*IBM PC compatible 80386 running at 20 MHz and 80387 math coprocessor with the Salford University FORTRAN77 compiler called (FTN77).

No difficulties were encountered from running the program on any of these machines.

The next section describes the program. All efforts were made to keep the structure as simple as possible so that any future modifications can be done without difficulties.

C.2 The Program

The program SUISCOM starts by calling the main subroutine SYSTEM.

This routine controls the main flow of the simulation process. It starts by initializing the variables which need to be initialized at the beginning of the simulation. Then it calls the routine INPUT which contains all the input variables of the particular simulation run.

After INPUT, the routine CONST is called. This routine calculates all the constants needed for the process using the values of the variables read by INPUT. Then the routine SETUP is called. This routine initializes all the variables and physical properties. Then a first call to the routine OUTPUT is made to report all the initialization variables.

All is now set to start the simulation process. The time is updated by adding the time step to it then a call to the routine GUESS is made. This routine guesses the values of the primary variables at the new time step. After GUESS, the routine MNEWT is called. This routine controls the solution of the system at the particular time step.

Then, depending on the outcome of MNEWT, the next call is made. Hence, if the solution fails a call to the routine UPDATE is made. This routine updates the time and goes back to repeat the solution of the system. If the solution is successful, a call to the routine NWSTEP is made. This routine calculates a new time step size. Then, a call to the routine OUTPUT is made to report the values of the new variables. Finally, a check is made to determine whether the simulation is finished. If it is, the program stops normally with the message "The Reservoir Is Empty Now". If it is not, the control goes back to the step where the time is updated and the whole process is repeated. Figure C.1 Shows a diagram of the main operations in subroutine SYSTEM.

C.3 Subroutine INPUT

This routine reads the following variables from a data file;

*The starting size of the time step.

*The maximum size of the time step.

*The maximum number of steps allowed for the whole simulation.

*The value of the underrelaxation constant URMX

*The value of the productivity index.

*The values of the maximum changes in the primary variables expected at any time steps.

*The maximum number of iterations allowed before the solution is terminated and the time is halved.

*The frequency in which the output is written to the output file.

*The tolerance allowed for each of the primary variables.

*The method chosen to calculate the mobility term in the flow equations.

*The reservoir length, width, and height.

*The reservoir initial pressure, temperature, oil saturation, water saturation, heavy oil concentration, oxygen concentration, and coke concentration.

*The flow rate, concentration and temperature of the injected gas.

*The dimension that the solution is required for.

*The number of grid blocks required for each dimension.

*Whether heat loss occurs or not.

The rest of the data required are in block data which can be modified according to the particular run.

C.4 Subroutine CONST

This routine calculates the size of the spatial variables i.e. Δx, Δy, and Δz. Then, it calculates the volume of the grid block.

It also calculates the enthalpy of the injected gas, the thermal diffusivity, and the productivity constant.

C.5 Subroutine SETUP

This routine initializes all the variables and the physical properties. The secondary variables and the physical properties calculations are done by calling the routine PHYPRO.

SETUP also uses the physical properties to calculate the volumetric amount of the original oil in place (VOOIP) and the volumetric amount of the original water in place (VOWIP).

C.6 Subroutine GUESS

This routine guesses the values of the primary variables at the new time step. This is actually done by assigning to them their values at the last successful solution.

C.7 Subroutine MNEWT

This is the main system solution routine. It starts by initializing the Jacobian matrix. Then, it call subroutine JACOBI which calculates the Jacobian matrix. After the Jacobian calculation, subroutine LUDCMP is called. This routine performs the LU decomposition of the Jacobian matrix. Subroutine LUBKSB is called then to perform the backward substitution. The new primary variables are checked for physical significance and compared with the previous iteration variables. If the solution converges, the control goes back to SYSTEM reporting a successful solution. If the solution does not converge, the solution goes back for another iteration and if the maximum number of iterations is exceeded, the control goes back to system reporting unsuccessful solution. Figure C.2 show a diagram of the main operations in MNEWT.

C.8 Subroutine UPDATE

This routine is called by SYSTEM when MNEWT reports a failed solution in order to update the time.

UPDATE calculates the new time step as the half of the previous step with which the solution was failed. Then, it checks to see if the new step is not less the minimum step size allowed. If it is, the simulation is terminated reporting "DT is too small". If it is not less than the minimum step size, the control goes back to SYSTEM.

C.9 Subroutine NWSTEP

This routine is called if MNEWT reports a successful solution. It first calculates the accumulated volumetric production of each component in each phase then, it calculates the percentage of the accumulated production rate of the oil and water to the original oil and water in place.

NWSTEP also calculates the total heat of reactions as well as the total heat losses to the surrounding strata.

A new time step size is calculated then the control goes back to SYSTEM.

C.10 Subroutine PHYPRO

This routine calculates the secondary variables and the physical properties. It calls the following routines;

*Subroutine EQUILB to calculate the equilibrium constants.

*Subroutine CAPEL to calculate the capillary pressures.

*Subroutine SECOND to calculate the secondary variables.

*Subroutine DENSTY to calculate each phase density.

*Subroutine ENTHAL to calculate each phase enthalpy and internal energy.

*Subroutine VISCOS to calculate each phase viscosity.

*Subroutine PERM to calculate each phase relative permeability.

C.11 Subroutine JACOBI

This routine calculates the Jacobian matrix of the system. It calls subroutine FUNC which calculates the material and energy balance equations.

C.12 Subroutine FUNC

This routine calculates the material and energy balance equations. This is done by first calling subroutine PHYPRO to calculate the physical properties and then subroutine EQUAT to calculate the equations.

C.13 Subroutine EQUAT

This routine calculates the mass and energy balance equations. It starts by calling subroutine DARCY which calculates the mobility term depending on the method chosen for this calculation. Then, the routine COKE is called to calculate the coke concentration. This call is followed by a call to subroutine REACT to calculate the rate of each of the reactions. Then, the following routines are called;

*Subroutine INERTS which calculates the inerts gas mass balance equation.

*Subroutine WATER which calculates the water mass balance equation.

*Subroutine LIGHT which calculates the light oil mass balance equation.

*Subroutine OXYGEN which calculates the oxygen mass balance equation.

*Subroutine HEAVY which calculates the heavy oil mass balance equation.

*Subroutine ENERGY which calculates the energy balance equation.

```
      PROGRAM SUISCOM
C**********************************************************
C***** THIS PROGRAM IS WRITTEN TO SIMULATE THE INSITU COMBUST
C***** PROCESS.

      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
C
C***** CALL THE MAIN SUBROUTINE *****************************
C
      CALL SYSTEM

      STOP
      END

      SUBROUTINE SYSTEM

C***** This is the main Subroutine
C
C Variable      Definition
C
C  AOTXXX        TOTAL MOLAR RECOVERY
C  CAPEN         CONSTANT IN THE HEAT LOSS SUBROUTINE
C  DT            TIME STEP
C  DTST          STARTING TIME STEP
C  ICOUNT        NUMBER OF TIME STEPS COUNTER
C  IFLAG1        SOLUTION REPORT FLAG
C  IFLAG3        OUTPUT FREQUENCY FLAG
C  NY            NUMBER OF GRID BLOCKS IN THE VERTICAL AXIS
C  PHL           CONSTANT IN THE HEAT LOSS SUBROUTINE
C  QHL           CONSTANT IN THE HEAT LOSS SUBROUTINE
C  SO            OIL SATURATION
C  VOTXXX        TOTAL VOLUMETRIC RECOVERY
C
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

        OPEN(UNIT=14,FILE='C:\PROJECT\OUTPUT\SUMM3D.DAT')

      ICOUNT = 0
      IFLAG1 = 0
```

```
      IFLAG3 = 1
      DO 113 KZ = 1 , NZ
      DO 111 JY = 1 , NY
      DO 112 IX = 1 , NX
C        AIHL(IX,JY,KZ) = 0.0
         PHL(IX,JY,KZ) = 0.0
         QHL(IX,JY,KZ) = 0.0
         CAPEN(IX,JY,KZ) = 0.0
112    CONTINUE
111    CONTINUE
113    CONTINUE
        DIFL = 0.0
      TOTHET = 0.0
      TOTLOS = 0.0
      AOTNR = 0.0
      AOTWTV = 0.0
      AOTWTL = 0.0
      AOTLTV = 0.0
      AOTLTL = 0.0
      AOTOX = 0.0
      AOTHVV = 0.0
      AOTHVL = 0.0
      VOTNR = 0.0
      VOTWTV = 0.0
      VOTWTL = 0.0
      VOTLTV = 0.0
      VOTLTL = 0.0
      VOTOX = 0.0
      VOTHVV = 0.0
      VOTHVL = 0.0

C READ THE INPUT VARIABLES **********************
      CALL INPUT

C CALL THE SUBROUTINE WHICH GIVES THE INLET FLOWRATES &ENTH.**
      CALL CONST

C GIVE THE INITIAL CONDITIONS AND THE PHYSICAL PROPERTIES ****
      CALL SETUP
```

```
C GUESS THE VARIABLES VALUES AT THE FIRST TIME STEP ********
      CALL PRIME
      TIM = 0.0
      DT = DTST
9999   TIM = TIM + DT
      ICOUNT = ICOUNT + 1
      TIME(ICOUNT) = TIM
       WRITE(*,*)'ICOUNT',ICOUNT,'TIME=',TIM,'DT=',DT
      CALL GUESS
C SOLVE THE SYSTEM BY NEWTON RAPHSON METHOD *********
      CALL MNEWT

      IF (IFLAG1 .EQ. 1) THEN
         CALL UPDATE
         GOTO 9999
      ENDIF
      IF(ICOUNT .LT. MAXSTP) THEN
        CALL NWSTEP
      IF (IFLAG3 .EQ. IFREQ) THEN
      IFLAG3 = 1
      CALL PRIME
      ELSE
      IFLAG3 = IFLAG3 + 1
      ENDIF
      DO 213 JY = 1 , NY
      IF (SO(NX,JY,NZ) .LT. 1.0D-04)THEN
        WRITE(14,*) 'THE RESERVIOR IS EMPTY NOW'
        STOP
      ENDIF
        GOTO 9999
213    CONTINUE
      ENDIF
      STOP
      END

      SUBROUTINE UPDATE

C***** THIS SUBROUTINE UPDATES THE TIME WHEN THE SOLUTION FAILS
```

```
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      TIM = TIM - DT
      TIME(ICOUNT) = TIM
      DT = DT/2.0
      IF(DT .LT. 1.0D-06) THEN
        WRITE(14,*)'DT IS TO SMALL'
        STOP
      ENDIF
      ICOUNT = ICOUNT - 1
      WRITE(14,*)'UPDATE ',ICOUNT+1,' DT=',DT,' TIME=',TIM+DT

      RETURN
      END

      SUBROUTINE NWSTEP

C***** THIS SUBROUTINE CALCULATES NEW TIME STEP AFTER SUCCESFUL
C         SOLUTION

C   Variable        Definition

C    DIFL           DIFFUSION LENGTH IN THE HEAT LOSS EQUATION
C    DX             PHASE DENSITY
C    OTXXX          MOLAR PRODUCTION RATE
C    PX             PHASE PRESSURE
C    ROTXXX         TOTAL RECOVER (% OF OOIP OR OWIP)
C    RPX            PHASE RELATIVE PERMEABILITY
C    SX             PHASE SATURATION
C    TM             TEMPERATURE
C    TOTHET         TOTAL HEAT GENERATED BY THE REACTIONS
C    TOTLOS         TOTAL HEAT LOSS
C    VTXXX          VOLUMETRIC PRODUCTION RATE
C    VX             PHASE VISCOSITY
C    XX             OIL COMPONENTS MOLE FRACTION IN THE OIL PHASE
C    YX             COMPONENTS MOLE FRACTION IN THE GAS PHASE
C
C
```

```
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
      DO 630 KZ = 1 , NZ
      DO 31 JY = 1 , NY
      DO 32 IX = 1 , NX
        CALL FUNC(IX,JY,KZ)
32     CONTINUE
31     CONTINUE
630    CONTINUE
      OLDDT = DT
      DO 633 KZ = 1 , NZ
      DO 33 JY = 1 , NY
       DO 34 IX = 1 , NX
      OTNR(IX,JY,KZ) = OTNR(IX,JY,KZ)*DT*VOLUMB
      OTWTV(IX,JY,KZ) = OTWTV(IX,JY,KZ)*DT*VOLUMB
      OTWTL(IX,JY,KZ) = OTWTL(IX,JY,KZ)*DT*VOLUMB
      OTLTL(IX,JY,KZ) = OTLTL(IX,JY,KZ)*DT*VOLUMB
      OTLTV(IX,JY,KZ) = OTLTV(IX,JY,KZ)*DT*VOLUMB
      OTOX(IX,JY,KZ) = OTOX(IX,JY,KZ)*DT*VOLUMB
      OTHVV(IX,JY,KZ) = OTHVV(IX,JY,KZ)*DT*VOLUMB
      OTHVL(IX,JY,KZ) = OTHVL(IX,JY,KZ)*DT*VOLUMB
      VTWTV = OTWTV(IX,JY,KZ)/STDWT
      VTWTL = OTWTL(IX,JY,KZ)/STDWT
      VTLTL = OTLTL(IX,JY,KZ)/STDLT
      VTLTV = OTLTV(IX,JY,KZ)/STDLT
      VTHVV = OTHVV(IX,JY,KZ)/STDHV
      VTHVL = OTHVL(IX,JY,KZ)/STDHV
      AOTNR = AOTNR + OTNR(IX,JY,KZ)
      AOTWTV = AOTWTV + OTWTV(IX,JY,KZ)
      AOTWTL = AOTWTL + OTWTL(IX,JY,KZ)
      AOTLTL = AOTLTL + OTLTL(IX,JY,KZ)
      AOTLTV = AOTLTV + OTLTV(IX,JY,KZ)
      AOTOX = AOTOX + OTOX(IX,JY,KZ)
      AOTHVV = AOTHVV + OTHVV(IX,JY,KZ)
      AOTHVL = AOTHVL + OTHVL(IX,JY,KZ)
      VOTWTV = VOTWTV + VTWTV
      VOTWTL = VOTWTL + VTWTL
      VOTLTL = VOTLTL + VTLTL
```

```
      VOTLTV = VOTLTV + VTLTV
      VOTHVV = VOTHVV + VTHVV
      VOTHVL = VOTHVL + VTHVL
34     CONTINUE
33     CONTINUE
633    CONTINUE
      ROTWTV = VOTWTV/VOWIP *100.0
      ROTWTL = VOTWTL/VOWIP *100.0
      ROTLTL = VOTLTL/VOOIP *100.0
      ROTLTV = VOTLTV/VOOIP *100.0
      ROTHVL = VOTHVL/VOOIP *100.0
      ROTHVV = VOTHVV/VOOIP *100.0
      DO 612 KZ = 1 , NZ
      DO 112 JY = 1 , NY
      DO 113 IX = 1 , NX
        TOTHET = TOTHET + HREAC(IX,JY,KZ)*DT*VOLUMB
        TOTLOS = TOTLOS + HLOSS(IX,JY,KZ)*VOLUMB
113    CONTINUE
112    CONTINUE
612    CONTINUE
      DO 610 KZ = 1 , NZ
      DO 11 JY = 1 , NY
      DO 10 IX = 1 , NX
        PGD(IX,JY,KZ) = ABS (PGP(IX,JY,KZ) - PG(IX,JY,KZ))
        SOD(IX,JY,KZ) = ABS (SOP(IX,JY,KZ) - SO(IX,JY,KZ))
        SWD(IX,JY,KZ) = ABS (SWP(IX,JY,KZ) - SW(IX,JY,KZ))
        YOD(IX,JY,KZ) = ABS (YOP(IX,JY,KZ) - YO(IX,JY,KZ))
        XHD(IX,JY,KZ) = ABS (XHP(IX,JY,KZ) - XH(IX,JY,KZ))
        TMD(IX,JY,KZ) = ABS (TMP(IX,JY,KZ) - TM(IX,JY,KZ))
10     CONTINUE
11     CONTINUE
610    CONTINUE

      PGDMAX = 0.0
      SODMAX = 0.0
      SWDMAX = 0.0
      YODMAX = 0.0
      XHDMAX = 0.0
      TMDMAX = 0.0
```

```
      DO 621 KZ = 1 , NZ
      DO 21 JY = 1 , NY
      DO 20 IX = 1 , NX
        IF(PGD(IX,JY,KZ) .GT. PGDMAX) PGDMAX = PGD(IX,JY,KZ)
        IF(SOD(IX,JY,KZ) .GT. SODMAX) SODMAX = SOD(IX,JY,KZ)
        IF(SWD(IX,JY,KZ) .GT. SWDMAX) SWDMAX = SWD(IX,JY,KZ)
        IF(YOD(IX,JY,KZ) .GT. YODMAX) YODMAX = YOD(IX,JY,KZ)
        IF(XHD(IX,JY,KZ) .GT. XHDMAX) XHDMAX = XHD(IX,JY,KZ)
        IF(TMD(IX,JY,KZ) .GT. TMDMAX) TMDMAX = TMD(IX,JY,KZ)
20    CONTINUE
21    CONTINUE
621   CONTINUE

      DEL(1) = 2.0*DNGP/(DNGP + PGDMAX) * DT
      DEL(2) = 2.0*DNOL/(DNOL + SODMAX) * DT
      DEL(3) = 2.0*DNWT/(DNWT + SWDMAX) * DT
      DEL(4) = 2.0*DNOX/(DNOX + YODMAX) * DT
      DEL(5) = 2.0*DNHL/(DNHL + XHDMAX) * DT
      DEL(6) = 2.0*DNTE/(DNTE + TMDMAX) * DT

      DT = 5.0
C     DEL(4) = DT
      DO 30 I = 1 , 6
        IF(DEL(I) .LT. DT) DT = DEL(I)
30    CONTINUE

      IF (DT .GT. DTMAX) DT = DTMAX
      DO 932 KZ = 1 , NZ
      DO 332 JY = 1 , NY
      DO 331 IX = 1 , NX
       PGP(IX,JY,KZ) = PG(IX,JY,KZ)
       TMP(IX,JY,KZ) = TM(IX,JY,KZ)
       SGP(IX,JY,KZ) = SG(IX,JY,KZ)
       SOP(IX,JY,KZ) = SO(IX,JY,KZ)
       SWP(IX,JY,KZ) = SW(IX,JY,KZ)
       YWP(IX,JY,KZ) = YW(IX,JY,KZ)
       YOP(IX,JY,KZ) = YO(IX,JY,KZ)
       YNP(IX,JY,KZ) = YN(IX,JY,KZ)
```

```
      YLP(IX,JY,KZ) = YL(IX,JY,KZ)
      YHP(IX,JY,KZ) = YH(IX,JY,KZ)
      XHP(IX,JY,KZ) = XH(IX,JY,KZ)
      XLP(IX,JY,KZ) = XL(IX,JY,KZ)
      DGP(IX,JY,KZ) = DG(IX,JY,KZ)
      DOP(IX,JY,KZ) = DO(IX,JY,KZ)
      DWP(IX,JY,KZ) = DW(IX,JY,KZ)
      CCP(IX,JY,KZ) = CC(IX,JY,KZ)
      ERP(IX,JY,KZ) = ER(IX,JY,KZ)
      EGP(IX,JY,KZ) = EG(IX,JY,KZ)
      EOP(IX,JY,KZ) = EO(IX,JY,KZ)
      EWP(IX,JY,KZ) = EW(IX,JY,KZ)
      HCP(IX,JY,KZ) = HC(IX,JY,KZ)
      PPHL(IX,JY,KZ) = PHL(IX,JY,KZ)
      PQHL(IX,JY,KZ) = QHL(IX,JY,KZ)
      PCAPEN(IX,JY,KZ) = CAPEN(IX,JY,KZ)
      VGP(IX,JY,KZ) = VG(IX,JY,KZ)
      VWP(IX,JY,KZ) = VW(IX,JY,KZ)
      VOP(IX,JY,KZ) = VO(IX,JY,KZ)
      RPGP(IX,JY,KZ) = RPG(IX,JY,KZ)
      RPOP(IX,JY,KZ) = RPO(IX,JY,KZ)
      RPWP(IX,JY,KZ) = RPW(IX,JY,KZ)
331    CONTINUE
332    CONTINUE
932    CONTINUE
      PDIFL = DIFL
      WRITE(14,*)'ICOUNT ',ICOUNT+1,' DT= ',DT, ' TIME= ',TIM+DT
      RETURN
      END

      SUBROUTINE GUESS

C***** THIS SUBROUTINE GUESSES THE NEW VARIABLES VALUES


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      DO 511 KZ = 1 , NZ
      DO 11 JY = 1 , NY
```

```
      DO 10 IX = 1 , NX
        I = (IX-1)*6 + 1 + (JY-1)*NN + (KZ-1)*NNZ
        X(I)   = PGP(IX,JY,KZ)
        X(I+1) = SWP(IX,JY,KZ)
        X(I+2) = SOP(IX,JY,KZ)
        X(I+3) = TMP(IX,JY,KZ)
        X(I+4) = YOP(IX,JY,KZ)
        X(I+5) = XHP(IX,JY,KZ)
10     CONTINUE
11     CONTINUE
511    CONTINUE
       IFLAG1 = 0
      DO 5001 KZ = 1 , NZ
      DO 3001 JY = 1 , NY
      DO 3000 IX = 1 , NX
        I = (IX-1)*6 + 1 + (JY-1)*NN + (KZ-1)*NNZ
        PG(IX,JY,KZ) = X(I)
        SW(IX,JY,KZ) = X(I+1)
        SO(IX,JY,KZ) = X(I+2)
        TM(IX,JY,KZ) = X(I+3)
        YO(IX,JY,KZ) = X(I+4)
        XH(IX,JY,KZ) = X(I+5)
3000   CONTINUE
3001   CONTINUE
5001   CONTINUE

      RETURN
      END

      SUBROUTINE SETUP

C***** THIS SUBROUTINE INITIALIZE SECONDARY VARIABLES AND
C      PHYSICAL PROPERTIES AT TIME =0.0


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      DO 711 KZ = 1 , NZ
      DO 11 JY = 1 , NY
```

```
      DO 10 IX = 1 , NX
       CALL PHYPRO(IX,JY,KZ)
10     CONTINUE
11     CONTINUE
711    CONTINUE
      DO 732 KZ = 1 , NZ
      DO 32 JY = 1 , NY
      DO 31 IX = 1 , NX
       PGP(IX,JY,KZ) = PG(IX,JY,KZ)
       TMP(IX,JY,KZ) = TM(IX,JY,KZ)
       SGP(IX,JY,KZ) = SG(IX,JY,KZ)
       SOP(IX,JY,KZ) = SO(IX,JY,KZ)
       SWP(IX,JY,KZ) = SW(IX,JY,KZ)
       YWP(IX,JY,KZ) = YW(IX,JY,KZ)
       YOP(IX,JY,KZ) = YO(IX,JY,KZ)
       YNP(IX,JY,KZ) = YN(IX,JY,KZ)
       YLP(IX,JY,KZ) = YL(IX,JY,KZ)
       YHP(IX,JY,KZ) = YH(IX,JY,KZ)
       XHP(IX,JY,KZ) = XH(IX,JY,KZ)
       XLP(IX,JY,KZ) = XL(IX,JY,KZ)
       DGP(IX,JY,KZ) = DG(IX,JY,KZ)
       DOP(IX,JY,KZ) = DO(IX,JY,KZ)
       DWP(IX,JY,KZ) = DW(IX,JY,KZ)
       CCP(IX,JY,KZ) = CC(IX,JY,KZ)
       ERP(IX,JY,KZ) = ER(IX,JY,KZ)
       EGP(IX,JY,KZ) = EG(IX,JY,KZ)
       EOP(IX,JY,KZ) = EO(IX,JY,KZ)
       EWP(IX,JY,KZ) = EW(IX,JY,KZ)
       HCP(IX,JY,KZ) = HC(IX,JY,KZ)
C      PAIHL(IX,JY,KZ) = AIHL(IX,JY,KZ)
       PPHL(IX,JY,KZ) = PHL(IX,JY,KZ)
       PQHL(IX,JY,KZ) = QHL(IX,JY,KZ)
       PCAPEN(IX,JY,KZ) = CAPEN(IX,JY,KZ)
       VGP(IX,JY,KZ) = VG(IX,JY,KZ)
       VWP(IX,JY,KZ) = VW(IX,JY,KZ)
       VOP(IX,JY,KZ) = VO(IX,JY,KZ)
       RPGP(IX,JY,KZ) = RPG(IX,JY,KZ)
       RPOP(IX,JY,KZ) = RPO(IX,JY,KZ)
       RPWP(IX,JY,KZ) = RPW(IX,JY,KZ)
```

```
31    CONTINUE
32    CONTINUE
732    CONTINUE
       PDIFL = DIFL

      OOIPL = SO(1,1,1)*POROS*VOLUMB*NX*DO(1,1,1)*NY*NZ
      OLTPV = SG(1,1,1)*POROS*VOLUMB*NX*(YL(1,1,1))*DG(1,1,1)*NY*N
      OHVPV = SG(1,1,1)*POROS*VOLUMB*NX*(YH(1,1,1))*DG(1,1,1)*NY*N
      VOOIP = (OOIPL + OHVPV)/STDHV + OLTPV/STDLT
      OWIPL = SW(1,1,1)*POROS*VOLUMB*NX*DW(1,1,1)*NY*NZ
      OWIPV = SG(1,1,1)*POROS*VOLUMB*NX*YW(1,1,1)*DG(1,1,1)*NY*NZ
      VOWIP  = (OWIPL + OWIPV)/STDWT
      RETURN
      END

      SUBROUTINE CONST

C***** THIS SUBROUTINE CALCULATES SOME OF THE CONSTANTS AT THE
C      BEGINNING OF THE SIMULATION

C   Variable          Definition

C   ARK               ROCK HEAT CAPACITY
C   DX,DY,DZ           GRID BLOCK SIZE
C   HIN               INJECTION GAS ENTHALPY
C   OTCN              PRODUCTIVITY CONSTANT
C   QX                INJECTION FLOW RATE
C   RESL,RESD,RESW     RESERVOIR SIZE
C   THCCAP             CAP ROCK THERMAL CONDUCTIVITY
C   THDIF              THERMAL DIFFUSIVITY
C   VOLUMB             GRID BLOCK VOLUME


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

        DX=RESL/NX
        DY=RESD/NY
        DZ=RESW/NZ
        VOLUMB = DX*DZ*DY
```

```
C        WRITE(14,*)'DX,DY,VOLUMB,DY',DX,DY,VOLUMB,DY
        DO 811 KZ = 1 , NZ
        DO 11 JY = 1 , NY
        DO 10 IX = 1 , NX-1
        OTCN(IX,JY,KZ) = 0.00
10     CONTINUE
       OTCN(NX,JY,KZ) = PRCN/(DX*DZ)
11     CONTINUE
811     CONTINUE
       DO 821 KZ = 1 , NZ
       DO 21 JY = 1 , NY
       DO 20 IX = 1 , NX
        QW(IX,JY,KZ) = 0.00
        QH(IX,JY,KZ) = 0.00
        QL(IX,JY,KZ) = 0.00
        QN(IX,JY,KZ) = 0.00
        QO(IX,JY,KZ) = 0.00
        HIN(IX,JY,KZ) = 0.00
C        TI(IX,JY,KZ) = TI(1)
20     CONTINUE
21     CONTINUE
821     CONTINUE
         DO 5445 JY = 3, 3
          QW(1,JY,1) = QW1/(VOLUMB*NY)
          QH(1,JY,1) = QH1/(VOLUMB*NY)
          QL(1,JY,1) = QL1/(VOLUMB*NY)
          QN(1,JY,1) = QN1/(VOLUMB*NY)
          QO(1,JY,1) = QO1/(VOLUMB*NY)
5445  CONTINUE
       DO 953 KZ = 1 , NZ
       DO 53 JY = 1 , NY
         DO 54 IX = 1 , NX
            PG(IX,JY,KZ) = PG(IX,1,1)
            TM(IX,JY,KZ) = TM(IX,1,1)
            SO(IX,JY,KZ) = SO(IX,1,1)
            SW(IX,JY,KZ) = SW(IX,1,1)
            YO(IX,JY,KZ) = YO(IX,1,1)
            XH(IX,JY,KZ) = XH(IX,1,1)
            CC(IX,JY,KZ) = CC(IX,1,1)
```

```
54      CONTINUE
53     CONTINUE
953    CONTINUE
       DO 862 KZ = 1 , NZ
       DO 62 JY = 1 , NY
         DO 63 IX = 1 , NX
          TI(IX,JY,KZ) = TM(IX,JY,KZ)
63      CONTINUE
62     CONTINUE
862    CONTINUE

       DO 6556 JY = 3 , 3
       HIN(1,JY,1) = (QW(1,JY,1)*AWT+QH(1,JY,1)*AHV+QL(1,JY,1)*ALT
     +          + QN(1,JY,1)*ANR+QO(1,JY,1)*AOX)*(TII-TREF)
6556   CONTINUE

        THDIF = THCCAP/ARK

         DO 8877 KZ = 1 , NZ
         DO 877 JY = 1 , NY
          DO 878 IX = 1 , NX
            AAX(IX,JY,KZ) = 0.5
            AAY(IX,JY,KZ) = 0.5
            AAZ(IX,JY,KZ) = 0.5
878       CONTINUE
877    CONTINUE
8877   CONTINUE
       RETURN
       END


       SUBROUTINE MNEWT

C***** THIS THE MAIN SOLUTION SUBROUTINE

C  Variable        Definition

C   ITER           NO OF ITERATION
C   RHS            RIGHT HAND SIDE OF THE BALANCE EQUATIONS
```

```
C    URMX              UNDERRELAXATION FACTOR


     INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
       DO 2 I = 1 , N
        DO 3 J = 1 , N
         A(I,J) = 0.0
3         CONTINUE
2        CONTINUE

     DO 13  K=1,MAXITR
       ITER = K
       IFLAG1 = 0
       IFLAG4 = 0
       CALL JACOBI
       IF (ITER .EQ. 1) DAMP=0.15*URMX
       IF (ITER .EQ. 2) DAMP=0.2*URMX
       IF (ITER .EQ. 3) DAMP=0.3*URMX
       IF (ITER .EQ. 4) DAMP=0.4*URMX
       IF (ITER .EQ. 5) DAMP=0.5*URMX
       IF (ITER .EQ. 6) DAMP=0.6*URMX
       IF (ITER .EQ. 7) DAMP=0.7*URMX
       IF (ITER .EQ. 8) DAMP=0.8*URMX
       IF (ITER .EQ. 9) DAMP=0.9*URMX
       IF (ITER .GE. 10) DAMP=1.0*URMX
       DO 7 I = 1 , N
         RHS(I) = DAMP*RHS(I)
7       CONTINUE

       CALL LUDCMP
       CALL LUBKSB

     DO 51 I = 1 , N
        IF(ABS(BETA(I)) .LT. 1.0D-18) BETA(I) = 0.000
51    CONTINUE
     DO 12 I=1,N
        X(I)=X(I)+BETA(I)
12    CONTINUE
     DO 5001 KZ = 1 , NZ
```

```
      DO 3001 JY = 1 , NY
      DO 3000 IX = 1 , NX
      I = (IX-1)*6 + 1 + (JY-1)*NN + (KZ-1)*NNZ
        PG(IX,JY,KZ) = X(I)
        SW(IX,JY,KZ) = X(I+1)
        SO(IX,JY,KZ) = X(I+2)
        TM(IX,JY,KZ) = X(I+3)
        YO(IX,JY,KZ) = X(I+4)
        XH(IX,JY,KZ) = X(I+5)
3000  CONTINUE
3001  CONTINUE
5001  CONTINUE
       DO 17 I = 1 , N , NV
         BETA(I) = BETA(I)/X(I)
17    CONTINUE
      DO 18 I = 4 , N , NV
        BETA(I) = BETA(I)/X(I)
18    CONTINUE
      DO 9111 KZ = 1 , NZ
      DO 111 JY = 1 , NY
      DO 211 IX = 1 , NX
       IF(SW(IX,JY,KZ) .LT. 0.000) THEN
        IF(SWP(IX,JY,KZ) .LT. 1.0D-04) THEN
      I = (IX-1)*6 + 1 + (JY-1)*NN + (KZ-1)*NNZ
C        SW(IX,JY,KZ) = DSQRT(SWP(IX,JY,KZ)*1.0D-05)
         SW(IX,JY,KZ) = SWP(IX,JY,KZ)*0.5
         BETA(I+1) = 0.00
        ELSE
         IFLAG1 = 1
        ENDIF
       ENDIF
211   CONTINUE
111   CONTINUE
9111  CONTINUE
      DO 9112 KZ = 1 , NZ
      DO 112 JY = 1 , NY
      DO 212 IX = 1 , NX
       IF(SO(IX,JY,KZ) .LT. 0.0000) THEN
        IF(SOP(IX,JY,KZ) .LT. 1.0D-04) THEN
```

```
      I = (IX-1)*6 + 1 + (JY-1)*NN + (KZ-1)*NNZ
C      SO(IX,JY,KZ) = DSQRT(SOP(IX,JY,KZ)*1.0D-05)
       SO(IX,JY,KZ) = SOP(IX,JY,KZ)*0.5
       BETA(I+2) = 0.0
      ELSE
       IFLAG1 = 1
      ENDIF
     ENDIF
212  CONTINUE
112  CONTINUE
9112 CONTINUE

    DO 9113 KZ = 1 , NZ
    DO 113 JY = 1 , NY
      DO 213 IX = 1 , NX
     IF(YO(IX,JY,KZ) .LT. 0.0000) THEN
      IF(YOP(IX,JY,KZ) .LT. 1.0D-04) THEN
      I = (IX-1)*6 + 1 + (JY-1)*NN + (KZ-1)*NNZ
C      YO(IX,JY,KZ) = DSQRT(YOP(IX,JY,KZ)*1.0D-05)
       YO(IX,JY,KZ) = YOP(IX,JY,KZ)
       BETA(I+4) = 0.0
      ELSE
       IFLAG1 = 1
      ENDIF
     ENDIF
     IF(YO(IX,JY,KZ) .GT. 1.0000) THEN
      IF(YOP(IX,JY,KZ) .GT. 0.9990) THEN
      I = (IX-1)*6 + 1 + (JY-1)*NN + (KZ-1)*NNZ
       YO(IX,JY,KZ) = DSQRT(YOP(IX,JY,KZ))
       BETA(I+4) = 0.00
      ELSE
       IFLAG1 = 1
      ENDIF
     ENDIF
213  CONTINUE
113  CONTINUE
9113 CONTINUE

    DO 9114 KZ = 1 , NZ
```

```
      DO 114 JY  = 1 , NY
        DO 214 IX = 1 , NX
       IF(XH(IX,JY,KZ) .GT. 1.00) THEN
        IF(XHP(IX,JY,KZ) .GT. 0.99) THEN
        I = (IX-1)*6 + 1 + (JY-1)*NN + (KZ-1)*NNZ
         XH(IX,JY,KZ) = DSQRT(XHP(IX,JY,KZ))
         BETA(I+5) = 0.00
        ELSE
         IFLAG1 = 1
        ENDIF
       ENDIF
214    CONTINUE
114    CONTINUE
9114   CONTINUE
      DO 9215 KZ = 1 , NZ
      DO 215 JY = 1 , NY
      DO 115 IX = 1 , NX
       IF(PG(IX,JY,KZ) .LT. 410.00 .OR. TM(IX,JY,KZ) .LT. 300.0) TH
        IFLAG1 = 1
       ENDIF
115    CONTINUE
215    CONTINUE
9215   CONTINUE
      IF(IFLAG1 .EQ. 1)GOTO 999
      DO 20 I = 1 , NV
       UMAX(I) = 0.0
20     CONTINUE
      DO 19 I = 1 , NV
       DO 21 J = I , N , NV
        IF(DABS(BETA(J)) .GT. UMAX(I))UMAX(I)=DABS(BETA(J))
21      CONTINUE
19     CONTINUE
      DO 22 IV = 1 , NV
       IF(UMAX(IV) .GT. TOLX(IV))THEN
        IFLAG4 = 1
       ENDIF
22     CONTINUE
      IF(IFLAG4 .NE. 1) GOTO 999
C      IF(ITER .GT. 20) THEN
```

```
C        IFLAG1 = 1
C      ELSE
C       GOTO 13
C      ENDIF
C     ENDIF
13   CONTINUE
     WRITE(14,*) 'NO CONVERGANCE REACHED'
     IFLAG1 = 1
999   CONTINUE
      WRITE(14,*)'ICOUNT=',ICOUNT,' ITER = ',ITER

     RETURN
     END


     SUBROUTINE JACOBI

C***** THIS SUBROUTINE CALCULATES THE JACOBIAN MATRIX
C

     INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

     MELIMF = 1
     DO 4006 KZ = 1 , NZ
     DO 6 JY = 1 , NY
       DO 5 J  = 1 , NX
      IFLAG5 = 0
      CALL FUNC(J,JY,KZ)
      IFLAG5 = 1
      I = (J-1)*6 + 1 + (JY-1)*NN + (KZ-1)*NNZ
      RHS(I)   = -EALPG
      RHS(I+1) = -EALSW
      RHS(I+2) = -EALSO
      RHS(I+3) = -EALTM
      RHS(I+4) = -EALYO
      RHS(I+5) = -EALXH

      IF(ITER .GE. 2) GOTO 13
       MELIMF = 0
```

```
IF(J .EQ. 1) GOTO 12
PSH = PG(J-1,JY,KZ)/5.0D01
PG(J-1,JY,KZ) = PG(J-1,JY,KZ) + PSH
CALL FUNC(J,JY,KZ)
A(I,I-6)   = (RHS(I) + EALPG)/PSH
A(I+1,I-6) = (RHS(I+1) + EALSW)/PSH
A(I+2,I-6) = (RHS(I+2) + EALSO)/PSH
A(I+3,I-6) = (RHS(I+3) + EALTM)/PSH
A(I+4,I-6) = (RHS(I+4) + EALYO)/PSH
A(I+5,I-6) = (RHS(I+5) + EALXH)/PSH
PG(J-1,JY,KZ) = PG(J-1,JY,KZ) - PSH
PSH = 0.0

IF(SWP(J-1,JY,KZ) .LT. 0.05) SWSH = SWP(J-1,JY,KZ)/2.0
IF(SWP(J-1,JY,KZ).GE.0.05)SWSH=0.01
SW(J-1,JY,KZ) = SW(J-1,JY,KZ) + SWSH
CALL FUNC(J,JY,KZ)
A(I,I-5)   = (RHS(I) + EALPG)/SWSH
A(I+1,I-5) = (RHS(I+1) + EALSW)/SWSH
A(I+2,I-5) = (RHS(I+2) + EALSO)/SWSH
A(I+3,I-5) = (RHS(I+3) + EALTM)/SWSH
A(I+4,I-5) = (RHS(I+4) + EALYO)/SWSH
A(I+5,I-5) = (RHS(I+5) + EALXH)/SWSH
SW(J-1,JY,KZ) = SW(J-1,JY,KZ) - SWSH
SWSH = 0.0

IF(SOP(J-1,JY,KZ) .LT. 0.008) SOSH = SOP(J-1,JY,KZ)/2.0
IF(SOP(J-1,JY,KZ).GE.0.008)SOSH=0.01
SO(J-1,JY,KZ) = SO(J-1,JY,KZ) + SOSH
CALL FUNC(J,JY,KZ)
A(I,I-4) = (RHS(I) + EALPG)/SOSH
A(I+1,I-4) = (RHS(I+1) + EALSW)/SOSH
A(I+2,I-4) = (RHS(I+2) + EALSO)/SOSH
A(I+3,I-4) = (RHS(I+3) + EALTM)/SOSH
A(I+4,I-4) = (RHS(I+4) + EALYO)/SOSH
A(I+5,I-4) = (RHS(I+5) + EALXH)/SOSH
SO(J-1,JY,KZ) = SO(J-1,JY,KZ) - SOSH
SOSH = 0.0
```

```
      TSH = TM(J-1,JY,KZ)/2.0D+1
      TM(J-1,JY,KZ) = TM(J-1,JY,KZ) + TSH
      CALL FUNC(J,JY,KZ)
      A(I,I-3) = (RHS(I) + EALPG)/TSH
      A(I+1,I-3) = (RHS(I+1) + EALSW)/TSH
      A(I+2,I-3) = (RHS(I+2) + EALSO)/TSH
      A(I+3,I-3) = (RHS(I+3) + EALTM)/TSH
      A(I+4,I-3) = (RHS(I+4) + EALYO)/TSH
      A(I+5,I-3) = (RHS(I+5) + EALXH)/TSH
      TM(J-1,JY,KZ) = TM(J-1,JY,KZ) - TSH
      TSH = 0.0

C      IF(YOP(J-1,JY,KZ) .GT. 0.98)YOSH=(1.0-YOP(J-1,JY,KZ))/2.0
      IF(YOP(J-1,JY,KZ) .GT. 0.98)YOSH=-1.0D-04
      IF(YOP(J-1,JY,KZ) .LE. 0.98 .OR. YOP(J-1,JY,KZ).GE.1.0)YOSH=
C     IF(YOP(J-1,JY,KZ) .LT. 0.0001) YOSH = -1.0D-04
      IF(YOP(J-1,JY,KZ) .LT. 0.01) YOSH = YOP(J-1,JY,KZ)/2.0
      IF(YOP(J-1,JY,KZ) .LT. 0.0001) YOSH = 1.0D-04
      YO(J-1,JY,KZ) = YO(J-1,JY,KZ) + YOSH
      CALL FUNC(J,JY,KZ)
      A(I,I-2) = (RHS(I) + EALPG)/YOSH
      A(I+1,I-2) = (RHS(I+1) + EALSW)/YOSH
      A(I+2,I-2) = (RHS(I+2) + EALSO)/YOSH
      A(I+3,I-2) = (RHS(I+3) + EALTM)/YOSH
      A(I+4,I-2) = (RHS(I+4) + EALYO)/YOSH
      A(I+5,I-2) = (RHS(I+5) + EALXH)/YOSH
      YO(J-1,JY,KZ) = YO(J-1,JY,KZ) - YOSH
      YOSH = 0.0

      IF(XHP(J-1,JY,KZ) .LT. 0.008)XHSH=- XHP(J-1,JY,KZ)/2.0
      IF(XHP(J-1,JY,KZ).GT.0.008)XHSH=0.0001
C     IF(XHP(J-1,JY,KZ) .GE. 1.000) XHSH = - 0.01
      IF(XHP(J-1,JY,KZ) .GT. 0.990) XHSH = - 0.0001
      XH(J-1,JY,KZ) = XH(J-1,JY,KZ) + XHSH
      CALL FUNC(J,JY,KZ)
      A(I,I-1) = (RHS(I) + EALPG)/XHSH
      A(I+1,I-1) = (RHS(I+1) + EALSW)/XHSH
      A(I+2,I-1) = (RHS(I+2) + EALSO)/XHSH
      A(I+3,I-1) = (RHS(I+3) + EALTM)/XHSH
```

```
      A(I+4,I-1) = (RHS(I+4) + EALYO)/XHSH
      A(I+5,I-1) = (RHS(I+5) + EALXH)/XHSH
      XH(J-1,JY,KZ) = XH(J-1,JY,KZ) - XHSH
      XHSH = 0.0
12    CONTINUE

      PSH = PG(J,JY,KZ)/5.0D01
      PG(J,JY,KZ) = PG(J,JY,KZ) + PSH
      CALL FUNC(J,JY,KZ)
      A(I,I)   = (RHS(I) + EALPG)/PSH
      A(I+1,I) = (RHS(I+1) + EALSW)/PSH
      A(I+2,I) = (RHS(I+2) + EALSO)/PSH
      A(I+3,I) = (RHS(I+3) + EALTM)/PSH
      A(I+4,I) = (RHS(I+4) + EALYO)/PSH
      A(I+5,I) = (RHS(I+5) + EALXH)/PSH
      PG(J,JY,KZ) = PG(J,JY,KZ) - PSH
      PSH = 0.0

      IF(SWP(J,JY,KZ) .LT. 0.05) SWSH = SWP(J,JY,KZ)/2.0
      IF(SWP(J,JY,KZ) .GE. 0.05)SWSH = 0.01
      SW(J,JY,KZ) = SW(J,JY,KZ) + SWSH
      CALL FUNC(J,JY,KZ)
      A(I,I+1)   = (RHS(I) + EALPG)/SWSH
      A(I+1,I+1) = (RHS(I+1) + EALSW)/SWSH
      A(I+2,I+1) = (RHS(I+2) + EALSO)/SWSH
      A(I+3,I+1) = (RHS(I+3) + EALTM)/SWSH
      A(I+4,I+1) = (RHS(I+4) + EALYO)/SWSH
      A(I+5,I+1) = (RHS(I+5) + EALXH)/SWSH
      SW(J,JY,KZ) = SW(J,JY,KZ) - SWSH
      SWSH = 0.0

      IF(SOP(J,JY,KZ) .LT. 0.008) SOSH = SOP(J,JY,KZ)/2.0
      IF(SOP(J,JY,KZ) .GE. 0.008)SOSH=0.01
      SO(J,JY,KZ) = SO(J,JY,KZ) + SOSH
      CALL FUNC(J,JY,KZ)
      A(I,I+2) = (RHS(I) + EALPG)/SOSH
      A(I+1,I+2) = (RHS(I+1) + EALSW)/SOSH
      A(I+2,I+2) = (RHS(I+2) + EALSO)/SOSH
      A(I+3,I+2) = (RHS(I+3) + EALTM)/SOSH
```

```
     A(I+4,I+2) = (RHS(I+4) + EALYO)/SOSH
     A(I+5,I+2) = (RHS(I+5) + EALXH)/SOSH
     SO(J,JY,KZ) = SO(J,JY,KZ) - SOSH
     SOSH = 0.0

     TSH = TM(J,JY,KZ)/2.0D+1
     TM(J,JY,KZ) = TM(J,JY,KZ) + TSH
     CALL FUNC(J,JY,KZ)
     A(I,I+3) = (RHS(I) + EALPG)/TSH
     A(I+1,I+3) = (RHS(I+1) + EALSW)/TSH
     A(I+2,I+3) = (RHS(I+2) + EALSO)/TSH
     A(I+3,I+3) = (RHS(I+3) + EALTM)/TSH
     A(I+4,I+3) = (RHS(I+4) + EALYO)/TSH
     A(I+5,I+3) = (RHS(I+5) + EALXH)/TSH
     TM(J,JY,KZ) = TM(J,JY,KZ) - TSH
     TSH = 0.0

C     IF(YOP(J,JY,KZ) .GT. 0.98) YOSH = (1.0-YOP(J,JY,KZ))/2.0
     IF(YOP(J,JY,KZ) .GT. 0.98)YOSH= -1.0D-04
     IF(YOP(J,JY,KZ) .LE. 0.98 .OR. YOP(J,JY,KZ) .GE. 1.0)YOSH=-
C    IF(YOP(J,JY,KZ) .LT. 0.0001) YOSH = -1.0D-04
     IF(YOP(J,JY,KZ) .LT. 0.01) YOSH = YOP(J,JY,KZ)/2.0
     IF(YOP(J,JY,KZ) .LT. 0.0001) YOSH = 1.0D-04
     YO(J,JY,KZ) = YO(J,JY,KZ) + YOSH
     CALL FUNC(J,JY,KZ)
     A(I,I+4) = (RHS(I) + EALPG)/YOSH
     A(I+1,I+4) = (RHS(I+1) + EALSW)/YOSH
     A(I+2,I+4) = (RHS(I+2) + EALSO)/YOSH
     A(I+3,I+4) = (RHS(I+3) + EALTM)/YOSH
     A(I+4,I+4) = (RHS(I+4) + EALYO)/YOSH
     A(I+5,I+4) = (RHS(I+5) + EALXH)/YOSH
     YO(J,JY,KZ) = YO(J,JY,KZ) - YOSH
     YOSH = 0.0

     IF(XHP(J,JY,KZ) .LT. 0.008) XHSH = - XHP(J,JY,KZ)/2.0
     IF(XHP(J,JY,KZ) .GT. 0.008)XHSH=0.0001
C    IF(XHP(J,JY,KZ) .GE. 1.000) XHSH = - 0.01
     IF(XHP(J,JY,KZ) .GT. 0.990) XHSH = - 0.0001
     XH(J,JY,KZ) = XH(J,JY,KZ) + XHSH
```

```
CALL FUNC(J,JY,KZ)
A(I,I+5) = (RHS(I) + EALPG)/XHSH
A(I+1,I+5) = (RHS(I+1) + EALSW)/XHSH
A(I+2,I+5) = (RHS(I+2) + EALSO)/XHSH
A(I+3,I+5) = (RHS(I+3) + EALTM)/XHSH
A(I+4,I+5) = (RHS(I+4) + EALYO)/XHSH
A(I+5,I+5) = (RHS(I+5) + EALXH)/XHSH
XH(J,JY,KZ) = XH(J,JY,KZ) - XHSH
XHSH = 0.0

IF (J .EQ. NX) GOTO 15
PSH = PG(J+1,JY,KZ)/5.0D01
PG(J+1,JY,KZ) = PG(J+1,JY,KZ) + PSH
CALL FUNC(J,JY,KZ)
A(I,I+6)   = (RHS(I) + EALPG)/PSH
A(I+1,I+6) = (RHS(I+1) + EALSW)/PSH
A(I+2,I+6) = (RHS(I+2) + EALSO)/PSH
A(I+3,I+6) = (RHS(I+3) + EALTM)/PSH
A(I+4,I+6) = (RHS(I+4) + EALYO)/PSH
A(I+5,I+6) = (RHS(I+5) + EALXH)/PSH
PG(J+1,JY,KZ) = PG(J+1,JY,KZ) - PSH
PSH = 0.0

IF(SWP(J+1,JY,KZ) .LT. 0.05) SWSH = SWP(J+1,JY,KZ)/2.0
IF(SWP(J+1,JY,KZ).GE.0.05)SWSH=0.01
SW(J+1,JY,KZ) = SW(J+1,JY,KZ) + SWSH
CALL FUNC(J,JY,KZ)
A(I,I+7)   = (RHS(I) + EALPG)/SWSH
A(I+1,I+7) = (RHS(I+1) + EALSW)/SWSH
A(I+2,I+7) = (RHS(I+2) + EALSO)/SWSH
A(I+3,I+7) = (RHS(I+3) + EALTM)/SWSH
A(I+4,I+7) = (RHS(I+4) + EALYO)/SWSH
A(I+5,I+7) = (RHS(I+5) + EALXH)/SWSH
SW(J+1,JY,KZ) = SW(J+1,JY,KZ) - SWSH
SWSH = 0.0

IF(SOP(J+1,JY,KZ) .LT. 0.008) SOSH = SOP(J+1,JY,KZ)/2.0
IF(SOP(J+1,JY,KZ) .GE. 0.008)SOSH=0.01
SO(J+1,JY,KZ) = SO(J+1,JY,KZ) + SOSH
```

```
      CALL FUNC(J,JY,KZ)
      A(I,I+8) = (RHS(I) + EALPG)/SOSH
      A(I+1,I+8) = (RHS(I+1) + EALSW)/SOSH
      A(I+2,I+8) = (RHS(I+2) + EALSO)/SOSH
      A(I+3,I+8) = (RHS(I+3) + EALTM)/SOSH
      A(I+4,I+8) = (RHS(I+4) + EALYO)/SOSH
      A(I+5,I+8) = (RHS(I+5) + EALXH)/SOSH
      SO(J+1,JY,KZ) = SO(J+1,JY,KZ) - SOSH
      SOSH = 0.0

      TSH = TM(J+1,JY,KZ)/2.0D+1
      TM(J+1,JY,KZ) = TM(J+1,JY,KZ) + TSH
      CALL FUNC(J,JY,KZ)
      A(I,I+9) = (RHS(I) + EALPG)/TSH
      A(I+1,I+9) = (RHS(I+1) + EALSW)/TSH
      A(I+2,I+9) = (RHS(I+2) + EALSO)/TSH
      A(I+3,I+9) = (RHS(I+3) + EALTM)/TSH
      A(I+4,I+9) = (RHS(I+4) + EALYO)/TSH
      A(I+5,I+9) = (RHS(I+5) + EALXH)/TSH
      TM(J+1,JY,KZ) = TM(J+1,JY,KZ) - TSH
      TSH = 0.0

C      IF(YOP(J+1,JY,KZ) .GT. 0.98) YOSH = (1.0-YOP(J+1,JY,KZ))/2.
      IF(YOP(J+1,JY,KZ) .GT. 0.98)YOSH= -1.0D-04
      IF(YOP(J+1,JY,KZ).LE.0.98 .OR. YOP(J+1,JY,KZ).GE.1.0)YOSH=-
C     IF(YOP(J+1,JY,KZ) .LT. 0.0001) YOSH = -1.0D-04
      IF(YOP(J+1,JY,KZ) .LT. 0.01) YOSH = YOP(J+1,JY,KZ)/2.0
      IF(YOP(J+1,JY,KZ) .LT. 0.0001) YOSH = 1.0D-04
      YO(J+1,JY,KZ) = YO(J+1,JY,KZ) + YOSH
      CALL FUNC(J,JY,KZ)
      A(I,I+10) = (RHS(I) + EALPG)/YOSH
      A(I+1,I+10) = (RHS(I+1) + EALSW)/YOSH
      A(I+2,I+10) = (RHS(I+2) + EALSO)/YOSH
      A(I+3,I+10) = (RHS(I+3) + EALTM)/YOSH
      A(I+4,I+10) = (RHS(I+4) + EALYO)/YOSH
      A(I+5,I+10) = (RHS(I+5) + EALXH)/YOSH
      YO(J+1,JY,KZ) = YO(J+1,JY,KZ) - YOSH
      YOSH = 0.0
```

```
      IF(XHP(J+1,JY,KZ) .LT. 0.008) XHSH = - XHP(J+1,JY,KZ)/2.0
      IF(XHP(J+1,JY,KZ) .GT.0.008)XHSH=0.0001
C      IF(XHP(J+1,JY,KZ) .GE. 1.000) XHSH = - 0.01
      IF(XHP(J+1,JY,KZ) .GT. 0.990) XHSH = - 0.0001
      XH(J+1,JY,KZ) = XH(J+1,JY,KZ) + XHSH
      CALL FUNC(J,JY,KZ)
      A(I,I+11) = (RHS(I) + EALPG)/XHSH
      A(I+1,I+11) = (RHS(I+1) + EALSW)/XHSH
      A(I+2,I+11) = (RHS(I+2) + EALSO)/XHSH
      A(I+3,I+11) = (RHS(I+3) + EALTM)/XHSH
      A(I+4,I+11) = (RHS(I+4) + EALYO)/XHSH
      A(I+5,I+11) = (RHS(I+5) + EALXH)/XHSH
      XH(J+1,JY,KZ) = XH(J+1,JY,KZ) - XHSH
      XHSH = 0.0

15     CONTINUE

      IF(JY .EQ. 1) GOTO 14
      PSH = PG(J,JY-1,KZ)/5.0D01
      PG(J,JY-1,KZ) = PG(J,JY-1,KZ) + PSH
      K = I - NN
      CALL FUNC(J,JY,KZ)
      A(I,K)   = (RHS(I) + EALPG)/PSH
      A(I+1,K) = (RHS(I+1) + EALSW)/PSH
      A(I+2,K) = (RHS(I+2) + EALSO)/PSH
      A(I+3,K) = (RHS(I+3) + EALTM)/PSH
      A(I+4,K) = (RHS(I+4) + EALYO)/PSH
      A(I+5,K) = (RHS(I+5) + EALXH)/PSH
      PG(J,JY-1,KZ) = PG(J,JY-1,KZ) - PSH
      PSH = 0.0

      IF(SWP(J,JY-1,KZ) .LT. 0.05) SWSH = SWP(J,JY-1,KZ)/2.0
      IF(SWP(J,JY-1,KZ) .GE. 0.05)SWSH=0.01
      SW(J,JY-1,KZ) = SW(J,JY-1,KZ) + SWSH
      CALL FUNC(J,JY,KZ)
      A(I,K+1)   = (RHS(I) + EALPG)/SWSH
      A(I+1,K+1) = (RHS(I+1) + EALSW)/SWSH
      A(I+2,K+1) = (RHS(I+2) + EALSO)/SWSH
      A(I+3,K+1) = (RHS(I+3) + EALTM)/SWSH
```

```
      A(I+4,K+1) = (RHS(I+4) + EALYO)/SWSH
      A(I+5,K+1) = (RHS(I+5) + EALXH)/SWSH
      SW(J,JY-1,KZ) = SW(J,JY-1,KZ) - SWSH
      SWSH = 0.0

      IF(SOP(J,JY-1,KZ) .LT. 0.008) SOSH = SOP(J,JY-1,KZ)/2.0
      IF(SOP(J,JY-1,KZ) .GE. 0.008)SOSH=0.01
      SO(J,JY-1,KZ) = SO(J,JY-1,KZ) + SOSH
      CALL FUNC(J,JY,KZ)
      A(I,K+2) = (RHS(I) + EALPG)/SOSH
      A(I+1,K+2) = (RHS(I+1) + EALSW)/SOSH
      A(I+2,K+2) = (RHS(I+2) + EALSO)/SOSH
      A(I+3,K+2) = (RHS(I+3) + EALTM)/SOSH
      A(I+4,K+2) = (RHS(I+4) + EALYO)/SOSH
      A(I+5,K+2) = (RHS(I+5) + EALXH)/SOSH
      SO(J,JY-1,KZ) = SO(J,JY-1,KZ) - SOSH
      SOSH = 0.0

      TSH = TM(J,JY-1,KZ)/2.0D+1
      TM(J,JY-1,KZ) = TM(J,JY-1,KZ) + TSH
      CALL FUNC(J,JY,KZ)
      A(I,K+3) = (RHS(I) + EALPG)/TSH
      A(I+1,K+3) = (RHS(I+1) + EALSW)/TSH
      A(I+2,K+3) = (RHS(I+2) + EALSO)/TSH
      A(I+3,K+3) = (RHS(I+3) + EALTM)/TSH
      A(I+4,K+3) = (RHS(I+4) + EALYO)/TSH
      A(I+5,K+3) = (RHS(I+5) + EALXH)/TSH
      TM(J,JY-1,KZ) = TM(J,JY-1,KZ) - TSH
      TSH = 0.0

C      IF(YOP(J,JY-1,KZ) .GT. 0.98) YOSH = (1.0-YOP(J,JY-1,KZ))/2.
      IF(YOP(J,JY-1,KZ) .GT. 0.98)YOSH= -1.0D-04
      IF(YOP(J,JY-1,KZ) .LE. 0.98 .OR. YOP(J,JY-1,KZ).GE.1.0)YOSH=
C     IF(YOP(J,JY-1,KZ) .LT. 0.0001) YOSH = -1.0D-04
      IF(YOP(J,JY-1,KZ) .LT. 0.01) YOSH = YOP(J,JY-1,KZ)/2.0
      IF(YOP(J,JY-1,KZ) .LT. 0.0001) YOSH = 1.0D-04
      YO(J,JY-1,KZ) = YO(J,JY-1,KZ) + YOSH
      CALL FUNC(J,JY,KZ)
      A(I,K+4) = (RHS(I) + EALPG)/YOSH
```

```
      A(I+1,K+4) = (RHS(I+1) + EALSW)/YOSH
      A(I+2,K+4) = (RHS(I+2) + EALSO)/YOSH
      A(I+3,K+4) = (RHS(I+3) + EALTM)/YOSH
      A(I+4,K+4) = (RHS(I+4) + EALYO)/YOSH
      A(I+5,K+4) = (RHS(I+5) + EALXH)/YOSH
      YO(J,JY-1,KZ) = YO(J,JY-1,KZ) - YOSH
      YOSH = 0.0

      IF(XHP(J,JY-1,KZ) .LT. 0.008) XHSH = - XHP(J,JY-1,KZ)/2.0
      IF(XHP(J,JY-1,KZ) .GT. 0.008)XHSH=0.0001
C     IF(XHP(J,JY-1,KZ) .GE. 1.000) XHSH = - 0.01
      IF(XHP(J,JY-1,KZ) .GT. 0.990) XHSH = - 0.0001
      XH(J,JY-1,KZ) = XH(J,JY-1,KZ) + XHSH
      CALL FUNC(J,JY,KZ)
      A(I,K+5) = (RHS(I) + EALPG)/XHSH
      A(I+1,K+5) = (RHS(I+1) + EALSW)/XHSH
      A(I+2,K+5) = (RHS(I+2) + EALSO)/XHSH
      A(I+3,K+5) = (RHS(I+3) + EALTM)/XHSH
      A(I+4,K+5) = (RHS(I+4) + EALYO)/XHSH
      A(I+5,K+5) = (RHS(I+5) + EALXH)/XHSH
      XH(J,JY-1,KZ) = XH(J,JY-1,KZ) - XHSH
      XHSH = 0.0

14    CONTINUE

      IF (JY .EQ. NY) GOTO 16
      K = I + NN
      PSH = PG(J,JY+1,KZ)/5.0D01
      PG(J,JY+1,KZ) = PG(J,JY+1,KZ) + PSH
      CALL FUNC(J,JY,KZ)
      A(I,K)   = (RHS(I) + EALPG)/PSH
      A(I+1,K) = (RHS(I+1) + EALSW)/PSH
      A(I+2,K) = (RHS(I+2) + EALSO)/PSH
      A(I+3,K) = (RHS(I+3) + EALTM)/PSH
      A(I+4,K) = (RHS(I+4) + EALYO)/PSH
      A(I+5,K) = (RHS(I+5) + EALXH)/PSH
      PG(J,JY+1,KZ) = PG(J,JY+1,KZ) - PSH
      PSH = 0.0
```

```
      IF(SWP(J,JY+1,KZ) .LT. 0.05) SWSH = SWP(J,JY+1,KZ)/2.0
      IF(SWP(J,JY+1,KZ) .GE. 0.05)SWSH=0.01
      SW(J,JY+1,KZ) = SW(J,JY+1,KZ) + SWSH
      CALL FUNC(J,JY,KZ)
      A(I,K+1)   = (RHS(I) + EALPG)/SWSH
      A(I+1,K+1) = (RHS(I+1) + EALSW)/SWSH
      A(I+2,K+1) = (RHS(I+2) + EALSO)/SWSH
      A(I+3,K+1) = (RHS(I+3) + EALTM)/SWSH
      A(I+4,K+1) = (RHS(I+4) + EALYO)/SWSH
      A(I+5,K+1) = (RHS(I+5) + EALXH)/SWSH
      SW(J,JY+1,KZ) = SW(J,JY+1,KZ) - SWSH
      SWSH = 0.0

      IF(SOP(J,JY+1,KZ) .LT. 0.008) SOSH = SOP(J,JY+1,KZ)/2.0
      IF(SOP(J,JY+1,KZ) .GE. 0.008)SOSH=0.01
      SO(J,JY+1,KZ) = SO(J,JY+1,KZ) + SOSH
      CALL FUNC(J,JY,KZ)
      A(I,K+2) = (RHS(I) + EALPG)/SOSH
      A(I+1,K+2) = (RHS(I+1) + EALSW)/SOSH
      A(I+2,K+2) = (RHS(I+2) + EALSO)/SOSH
      A(I+3,K+2) = (RHS(I+3) + EALTM)/SOSH
      A(I+4,K+2) = (RHS(I+4) + EALYO)/SOSH
      A(I+5,K+2) = (RHS(I+5) + EALXH)/SOSH
      SO(J,JY+1,KZ) = SO(J,JY+1,KZ) - SOSH
      SOSH = 0.0

      TSH = TM(J,JY+1,KZ)/2.0D+1
      TM(J,JY+1,KZ) = TM(J,JY+1,KZ) + TSH
      CALL FUNC(J,JY,KZ)
      A(I,K+3) = (RHS(I) + EALPG)/TSH
      A(I+1,K+3) = (RHS(I+1) + EALSW)/TSH
      A(I+2,K+3) = (RHS(I+2) + EALSO)/TSH
      A(I+3,K+3) = (RHS(I+3) + EALTM)/TSH
      A(I+4,K+3) = (RHS(I+4) + EALYO)/TSH
      A(I+5,K+3) = (RHS(I+5) + EALXH)/TSH
      TM(J,JY+1,KZ) = TM(J,JY+1,KZ) - TSH
      TSH = 0.0

C       IF(YOP(J,JY+1,KZ) .GT. 0.98) YOSH = (1.0-YOP(J,JY+1,KZ))/2.
```

```
      IF(YOP(J,JY+1,KZ) .GT. 0.98)YOSH= -1.0D-04
      IF(YOP(J,JY+1,KZ) .LE.0.98 .OR. YOP(J,JY+1,KZ).GE.1.0)YOSH=
C      IF(YOP(J,JY+1,KZ) .LT. 0.0001) YOSH = -1.0D-04
      IF(YOP(J,JY+1,KZ) .LT. 0.01) YOSH = YOP(J,JY+1,KZ)/2.0
      IF(YOP(J,JY+1,KZ) .LT. 0.0001) YOSH = 1.0D-04
      YO(J,JY+1,KZ) = YO(J,JY+1,KZ) + YOSH
      CALL FUNC(J,JY,KZ)
      A(I,K+4) = (RHS(I) + EALPG)/YOSH
      A(I+1,K+4) = (RHS(I+1) + EALSW)/YOSH
      A(I+2,K+4) = (RHS(I+2) + EALSO)/YOSH
      A(I+3,K+4) = (RHS(I+3) + EALTM)/YOSH
      A(I+4,K+4) = (RHS(I+4) + EALYO)/YOSH
      A(I+5,K+4) = (RHS(I+5) + EALXH)/YOSH
      YO(J,JY+1,KZ) = YO(J,JY+1,KZ) - YOSH
      YOSH = 0.0

      IF(XHP(J,JY+1,KZ) .LT. 0.008) XHSH = - XHP(J,JY+1,KZ)/2.0
      IF(XHP(J,JY+1,KZ) .GT.0.008)XHSH=0.0001
C      IF(XHP(J,JY+1,KZ) .GE. 1.000) XHSH = - 0.01
      IF(XHP(J,JY+1,KZ) .GT. 0.990) XHSH = - 0.0001
      XH(J,JY+1,KZ) = XH(J,JY+1,KZ) + XHSH
      CALL FUNC(J,JY,KZ)
      A(I,K+5) = (RHS(I) + EALPG)/XHSH
      A(I+1,K+5) = (RHS(I+1) + EALSW)/XHSH
      A(I+2,K+5) = (RHS(I+2) + EALSO)/XHSH
      A(I+3,K+5) = (RHS(I+3) + EALTM)/XHSH
      A(I+4,K+5) = (RHS(I+4) + EALYO)/XHSH
      A(I+5,K+5) = (RHS(I+5) + EALXH)/XHSH
      XH(J,JY+1,KZ) = XH(J,JY+1,KZ) - XHSH
      XHSH = 0.0
 16   CONTINUE
      IF(KZ .EQ. 1) GOTO 17
      PSH = PG(J,JY,KZ-1)/5.0D01
      PG(J,JY,KZ-1) = PG(J,JY,KZ-1) + PSH
      K = I - NNZ
      CALL FUNC(J,JY,KZ)
      A(I,K)   = (RHS(I) + EALPG)/PSH
      A(I+1,K) = (RHS(I+1) + EALSW)/PSH
      A(I+2,K) = (RHS(I+2) + EALSO)/PSH
```

```
A(I+3,K) = (RHS(I+3) + EALTM)/PSH
A(I+4,K) = (RHS(I+4) + EALYO)/PSH
A(I+5,K) = (RHS(I+5) + EALXH)/PSH
PG(J,JY,KZ-1) = PG(J,JY,KZ-1) - PSH
PSH = 0.0

IF(SWP(J,JY,KZ-1) .LT. 0.05) SWSH = SWP(J,JY,KZ-1)/2.0
IF(SWP(J,JY,KZ-1) .GE. 0.05)SWSH=0.01
SW(J,JY,KZ-1) = SW(J,JY,KZ-1) + SWSH
CALL FUNC(J,JY,KZ)
A(I,K+1)   = (RHS(I) + EALPG)/SWSH
A(I+1,K+1) = (RHS(I+1) + EALSW)/SWSH
A(I+2,K+1) = (RHS(I+2) + EALSO)/SWSH
A(I+3,K+1) = (RHS(I+3) + EALTM)/SWSH
A(I+4,K+1) = (RHS(I+4) + EALYO)/SWSH
A(I+5,K+1) = (RHS(I+5) + EALXH)/SWSH
SW(J,JY,KZ-1) = SW(J,JY,KZ-1) - SWSH
SWSH = 0.0

IF(SOP(J,JY,KZ-1) .LT. 0.008) SOSH = SOP(J,JY,KZ-1)/2.0
IF(SOP(J,JY,KZ-1) .GE. 0.008)SOSH=0.01
SO(J,JY,KZ-1) = SO(J,JY,KZ-1) + SOSH
CALL FUNC(J,JY,KZ)
A(I,K+2) = (RHS(I) + EALPG)/SOSH
A(I+1,K+2) = (RHS(I+1) + EALSW)/SOSH
A(I+2,K+2) = (RHS(I+2) + EALSO)/SOSH
A(I+3,K+2) = (RHS(I+3) + EALTM)/SOSH
A(I+4,K+2) = (RHS(I+4) + EALYO)/SOSH
A(I+5,K+2) = (RHS(I+5) + EALXH)/SOSH
SO(J,JY,KZ-1) = SO(J,JY,KZ-1) - SOSH
SOSH = 0.0

TSH = TM(J,JY,KZ-1)/2.0D+1
TM(J,JY,KZ-1) = TM(J,JY,KZ-1) + TSH
CALL FUNC(J,JY,KZ)
A(I,K+3) = (RHS(I) + EALPG)/TSH
A(I+1,K+3) = (RHS(I+1) + EALSW)/TSH
A(I+2,K+3) = (RHS(I+2) + EALSO)/TSH
A(I+3,K+3) = (RHS(I+3) + EALTM)/TSH
```

```
      A(I+4,K+3) = (RHS(I+4) + EALYO)/TSH
      A(I+5,K+3) = (RHS(I+5) + EALXH)/TSH
      TM(J,JY,KZ-1) = TM(J,JY,KZ-1) - TSH
      TSH = 0.0

C       IF(YOP(J,JY,KZ-1) .GT. 0.98) YOSH = (1.0-YOP(J,JY,KZ-1))/2.
      IF(YOP(J,JY,KZ-1) .GT. 0.98)YOSH= -1.0D-04
      IF(YOP(J,JY,KZ-1) .LE. 0.98 .OR. YOP(J,JY,KZ-1).GE.1.0)YOSH=
C      IF(YOP(J,JY,KZ-1) .LT. 0.0001) YOSH = -1.0D-04
      IF(YOP(J,JY,KZ-1) .LT. 0.01) YOSH = YOP(J,JY,KZ-1)/2.0
      IF(YOP(J,JY,KZ-1) .LT. 0.0001) YOSH = 1.0D-04
      YO(J,JY,KZ-1) = YO(J,JY,KZ-1) + YOSH
      CALL FUNC(J,JY,KZ)
      A(I,K+4) = (RHS(I) + EALPG)/YOSH
      A(I+1,K+4) = (RHS(I+1) + EALSW)/YOSH
      A(I+2,K+4) = (RHS(I+2) + EALSO)/YOSH
      A(I+3,K+4) = (RHS(I+3) + EALTM)/YOSH
      A(I+4,K+4) = (RHS(I+4) + EALYO)/YOSH
      A(I+5,K+4) = (RHS(I+5) + EALXH)/YOSH
      YO(J,JY,KZ-1) = YO(J,JY,KZ-1) - YOSH
      YOSH = 0.0

      IF(XHP(J,JY,KZ-1) .LT. 0.008) XHSH = - XHP(J,JY,KZ-1)/2.0
      IF(XHP(J,JY,KZ-1) .GT. 0.008)XHSH=0.0001
C      IF(XHP(J,JY,KZ-1) .GE. 1.000) XHSH = - 0.01
      IF(XHP(J,JY,KZ-1) .GT. 0.990) XHSH = - 0.0001
      XH(J,JY,KZ-1) = XH(J,JY,KZ-1) + XHSH
      CALL FUNC(J,JY,KZ)
      A(I,K+5) = (RHS(I) + EALPG)/XHSH
      A(I+1,K+5) = (RHS(I+1) + EALSW)/XHSH
      A(I+2,K+5) = (RHS(I+2) + EALSO)/XHSH
      A(I+3,K+5) = (RHS(I+3) + EALTM)/XHSH
      A(I+4,K+5) = (RHS(I+4) + EALYO)/XHSH
      A(I+5,K+5) = (RHS(I+5) + EALXH)/XHSH
      XH(J,JY,KZ-1) = XH(J,JY,KZ-1) - XHSH
      XHSH = 0.0

17    CONTINUE
```

```
IF(KZ .EQ. NZ) GOTO 18
PSH = PG(J,JY,KZ+1)/5.0D01
PG(J,JY,KZ+1) = PG(J,JY,KZ+1) + PSH
K = I + NNZ
CALL FUNC(J,JY,KZ)
A(I,K)   = (RHS(I) + EALPG)/PSH
A(I+1,K) = (RHS(I+1) + EALSW)/PSH
A(I+2,K) = (RHS(I+2) + EALSO)/PSH
A(I+3,K) = (RHS(I+3) + EALTM)/PSH
A(I+4,K) = (RHS(I+4) + EALYO)/PSH
A(I+5,K) = (RHS(I+5) + EALXH)/PSH
PG(J,JY,KZ+1) = PG(J,JY,KZ+1) - PSH
PSH = 0.0

IF(SWP(J,JY,KZ+1) .LT. 0.05) SWSH = SWP(J,JY,KZ+1)/2.0
IF(SWP(J,JY,KZ+1) .GE. 0.05)SWSH=0.01
SW(J,JY,KZ+1) = SW(J,JY,KZ+1) + SWSH
CALL FUNC(J,JY,KZ)
A(I,K+1)   = (RHS(I) + EALPG)/SWSH
A(I+1,K+1) = (RHS(I+1) + EALSW)/SWSH
A(I+2,K+1) = (RHS(I+2) + EALSO)/SWSH
A(I+3,K+1) = (RHS(I+3) + EALTM)/SWSH
A(I+4,K+1) = (RHS(I+4) + EALYO)/SWSH
A(I+5,K+1) = (RHS(I+5) + EALXH)/SWSH
SW(J,JY,KZ+1) = SW(J,JY,KZ+1) - SWSH
SWSH = 0.0

IF(SOP(J,JY,KZ+1) .LT. 0.008) SOSH = SOP(J,JY,KZ+1)/2.0
IF(SOP(J,JY,KZ+1) .GE. 0.008)SOSH=0.01
SO(J,JY,KZ+1) = SO(J,JY,KZ+1) + SOSH
CALL FUNC(J,JY,KZ)
A(I,K+2) = (RHS(I) + EALPG)/SOSH
A(I+1,K+2) = (RHS(I+1) + EALSW)/SOSH
A(I+2,K+2) = (RHS(I+2) + EALSO)/SOSH
A(I+3,K+2) = (RHS(I+3) + EALTM)/SOSH
A(I+4,K+2) = (RHS(I+4) + EALYO)/SOSH
A(I+5,K+2) = (RHS(I+5) + EALXH)/SOSH
SO(J,JY,KZ+1) = SO(J,JY,KZ+1) - SOSH
SOSH = 0.0
```

```
      TSH = TM(J,JY,KZ+1)/2.0D+1
      TM(J,JY,KZ+1) = TM(J,JY,KZ+1) + TSH
      CALL FUNC(J,JY,KZ)
      A(I,K+3) = (RHS(I) + EALPG)/TSH
      A(I+1,K+3) = (RHS(I+1) + EALSW)/TSH
      A(I+2,K+3) = (RHS(I+2) + EALSO)/TSH
      A(I+3,K+3) = (RHS(I+3) + EALTM)/TSH
      A(I+4,K+3) = (RHS(I+4) + EALYO)/TSH
      A(I+5,K+3) = (RHS(I+5) + EALXH)/TSH
      TM(J,JY,KZ+1) = TM(J,JY,KZ+1) - TSH
      TSH = 0.0

C      IF(YOP(J,JY,KZ+1) .GT. 0.98) YOSH = (1.0-YOP(J,JY,KZ+1))/2.
      IF(YOP(J,JY,KZ+1) .GT. 0.98)YOSH= -1.0D-04
      IF(YOP(J,JY,KZ+1) .LE. 0.98 .OR. YOP(J,JY,KZ+1).GE.1.0)YOSH=
C     IF(YOP(J,JY,KZ+1) .LT. 0.0001) YOSH = -1.0D-04
      IF(YOP(J,JY,KZ+1) .LT. 0.01) YOSH = YOP(J,JY,KZ+1)/2.0
      IF(YOP(J,JY,KZ+1) .LT. 0.0001) YOSH = 1.0D-04
      YO(J,JY,KZ+1) = YO(J,JY,KZ+1) + YOSH
      CALL FUNC(J,JY,KZ)
      A(I,K+4) = (RHS(I) + EALPG)/YOSH
      A(I+1,K+4) = (RHS(I+1) + EALSW)/YOSH
      A(I+2,K+4) = (RHS(I+2) + EALSO)/YOSH
      A(I+3,K+4) = (RHS(I+3) + EALTM)/YOSH
      A(I+4,K+4) = (RHS(I+4) + EALYO)/YOSH
      A(I+5,K+4) = (RHS(I+5) + EALXH)/YOSH
      YO(J,JY,KZ+1) = YO(J,JY,KZ+1) - YOSH
      YOSH = 0.0

      IF(XHP(J,JY,KZ+1) .LT. 0.008) XHSH = - XHP(J,JY,KZ+1)/2.0
      IF(XHP(J,JY,KZ+1) .GT. 0.008)XHSH=0.0001
C     IF(XHP(J,JY,KZ+1) .GE. 1.000) XHSH = - 0.01
      IF(XHP(J,JY,KZ+1) .GT. 0.990) XHSH = - 0.0001
      XH(J,JY,KZ+1) = XH(J,JY,KZ+1) + XHSH
      CALL FUNC(J,JY,KZ)
      A(I,K+5) = (RHS(I) + EALPG)/XHSH
      A(I+1,K+5) = (RHS(I+1) + EALSW)/XHSH
      A(I+2,K+5) = (RHS(I+2) + EALSO)/XHSH
```

```
      A(I+3,K+5) = (RHS(I+3) + EALTM)/XHSH
      A(I+4,K+5) = (RHS(I+4) + EALYO)/XHSH
      A(I+5,K+5) = (RHS(I+5) + EALXH)/XHSH
      XH(J,JY,KZ+1) = XH(J,JY,KZ+1) - XHSH
      XHSH = 0.0

18    CONTINUE
13    CONTINUE

5     CONTINUE
6     CONTINUE
4006  CONTINUE

      RETURN
      END

      SUBROUTINE FUNC(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE RIGHT HAND SIDE OF THE BALANCE
C      EQUATIONS


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      CALL PHYPRO(IX,JY,KZ)
      CALL EQUAT(IX,JY,KZ)
       EALPG = EALPG + EALYO
       EALSO = EALSO + EALXH

      RETURN
      END


      SUBROUTINE LUDCMP

C***** THIS SUBROUTINE PERFORMS THE LU DECOMPOSITION


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
```

```
       IF (MELIMF .EQ. 1) GOTO 5000
       TINY=1.0E-20
      D=1.
      DO 12 I=1,N
       AAMAX=0.
       DO 11 J=1,N
        IF (ABS(A(I,J)).GT.AAMAX) AAMAX=ABS(A(I,J))
11     CONTINUE
       IF (AAMAX.EQ.0.) THEN
        WRITE(*,*) 'Singular matrix.'
        WRITE(14,*) 'Singular matrix.'
        WRITE(*,*)
        STOP
       ENDIF
       VV(I)=1./AAMAX
12    CONTINUE
      DO 19 J=1,N
       IF (J.GT.1) THEN
        DO 14 I=1,J-1
         SUM=A(I,J)
         IF (I.GT.1)THEN
          DO 13 K=1,I-1
           SUM=SUM-A(I,K)*A(K,J)
13        CONTINUE
          A(I,J)=SUM
         ENDIF
14      CONTINUE
       ENDIF
       AAMAX=0.
       DO 16 I=J,N
        SUM=A(I,J)
        IF (J.GT.1)THEN
         DO 15 K=1,J-1
          SUM=SUM-A(I,K)*A(K,J)
15       CONTINUE
         A(I,J)=SUM
        ENDIF
        DUM=VV(I)*ABS(SUM)
        IF (DUM.GE.AAMAX) THEN
```

```
              IMAX=I
              AAMAX=DUM
            ENDIF
16         CONTINUE
           IF (J.NE.IMAX)THEN
             DO 17 K=1,N
               DUM=A(IMAX,K)
               A(IMAX,K)=A(J,K)
               A(J,K)=DUM
17           CONTINUE
             D=-D
             VV(IMAX)=VV(J)
           ENDIF
           INDX(J)=IMAX
           IF(J.NE.N)THEN
             IF(A(J,J).EQ.0.)A(J,J)=TINY
             DUM=1./A(J,J)
             DO 18 I=J+1,N
               A(I,J)=A(I,J)*DUM
18           CONTINUE
           ENDIF
19       CONTINUE
         IF(A(N,N).EQ.0.)A(N,N)=TINY
5000       CONTINUE
         RETURN
         END

         SUBROUTINE LUBKSB

C***** THIS SUBROUTINE PERFORMS THE BACK SUBTITUTION


         INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
         DO 229 I = 1 , N
           B(I) = RHS(I)
229       CONTINUE
         II=0
         DO 12 I=1,N
          LL=INDX(I)
```

```
          SUM=B(LL)
          B(LL)=B(I)
          IF (II.NE.0)THEN
            DO 11 J=II,I-1
              SUM=SUM-A(I,J)*B(J)
11          CONTINUE
          ELSE IF (SUM.NE.0.) THEN
            II=I
          ENDIF
          B(I)=SUM
12      CONTINUE
        DO 14 I=N,1,-1
          SUM=B(I)
          IF(I.LT.N)THEN
            DO 13 J=I+1,N
              SUM=SUM-A(I,J)*B(J)
13          CONTINUE
          ENDIF
          B(I)=SUM/A(I,I)
14      CONTINUE
        DO 239 I = 1 , N
          BETA(I) = B(I)
239     CONTINUE
        RETURN
        END


        SUBROUTINE EQUAT(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE MASS AND HEAT BALANCE EQUATIONS


        INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

        CALL MOLWT(IX,JY,KZ)
C           IF (ITER .EQ. 1 .AND. IFLAG5 .EQ. 0)THEN
C             CALL OMEGA(IX,JY,KZ)
C           ENDIF
        CALL DARCY(IX,JY,KZ)
```

```
      GGRAV1 = - GGRAV1
      GGRAV2 = - GGRAV2
      WGRAV1 = - WGRAV1
      WGRAV2 = - WGRAV2
      OGRAV1 = - OGRAV1
      OGRAV2 = - OGRAV2
      CALL COKE(IX,JY,KZ)
      CALL REACT(IX,JY,KZ)

C... 1- INERTS MASS BALANCE EQUATION
      CALL INERTS(IX,JY,KZ)

C... 2- WATER MASS BALANCE EQUATION
      CALL WATER(IX,JY,KZ)

C... 3- LIGHT OIL MASS BALANCE EQUATION
      CALL LIGHT(IX,JY,KZ)

C... 4- OXYGEN MASS BALANCE EQUATION
      CALL OXYGEN(IX,JY,KZ)

C... 5- HEAVY OIL MASS BALANCE
      CALL HEAVY(IX,JY,KZ)

C... 6- ENERGY BALANCE EQUATION
      CALL ENERGY(IX,JY,KZ)
      RETURN
      END


      SUBROUTINE OMEGA(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES OMEGA IF WHEATLY METHOD IS USED

      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
      IF(TIM .GT. 2.00)THEN
      IF(IX .EQ. 1) THEN
        OMGX(IX,JY,KZ) = 1.0D+00
        OMOX(IX,JY,KZ) = 1.0D+00
```

```
  OMWX(IX,JY,KZ) = 1.0D+00
ELSEIF(IX .GT. 1 .AND. IX .NE. NX)THEN
 DUMY1G = RPGP(IX,JY,KZ)*DGP(IX,JY,KZ)/VGP(IX,JY,KZ)
 DUMY2G = RPGP(IX-1,JY,KZ)*DGP(IX-1,JY,KZ)/VGP(IX-1,JY,KZ)
 DUMY3G = RPGP(IX+1,JY,KZ)*DGP(IX+1,JY,KZ)/VGP(IX+1,JY,KZ)



 DUMY1O = RPOP(IX,JY,KZ)*DOP(IX,JY,KZ)/VOP(IX,JY,KZ)
 DUMY2O = RPOP(IX-1,JY,KZ)*DOP(IX-1,JY,KZ)/VOP(IX-1,JY,KZ)
 DUMY3O = RPOP(IX+1,JY,KZ)*DOP(IX+1,JY,KZ)/VOP(IX+1,JY,KZ)

 DUMY1W = RPWP(IX,JY,KZ)*DWP(IX,JY,KZ)/VWP(IX,JY,KZ)
 DUMY2W = RPWP(IX-1,JY,KZ)*DWP(IX-1,JY,KZ)/VWP(IX-1,JY,KZ)
 DUMY3W = RPWP(IX+1,JY,KZ)*DWP(IX+1,JY,KZ)/VWP(IX+1,JY,KZ)
 DUMY4 = ABS(DUMY3G - DUMY1G)
 IF (DUMY4 .LT. 1.0D-20) THEN
  OMGX(IX,JY,KZ) = 1.0D+00
 ELSE
 OMGX(IX,JY,KZ)=1.-AAX(IX,JY,KZ)*((DUMY1G-DUMY2G)/(DUMY3G-DU
 ENDIF
 DUMY5 = ABS(DUMY3O - DUMY1O)
 IF (DUMY5 .LT. 1.0D-20) THEN
  OMOX(IX,JY,KZ) = 1.0D+00
 ELSE
 OMOX(IX,JY,KZ)=1.-AAX(IX,JY,KZ)*((DUMY1O-DUMY2O)/(DUMY3O-DU
 ENDIF
 DUMY6 = ABS(DUMY3W - DUMY1W)
 IF (DUMY6 .LT. 1.0D-20) THEN
  OMWX(IX,JY,KZ) = 1.0D+00
 ELSE
 OMWX(IX,JY,KZ)=1.-AAX(IX,JY,KZ)*((DUMY1W-DUMY2W)/(DUMY3W-DU
 ENDIF

ELSEIF(IX .EQ. NX)THEN
  OMGX(IX,JY,KZ) = 1.0D+00
  OMOX(IX,JY,KZ) = 1.0D+00
  OMWX(IX,JY,KZ) = 1.0D+00
ENDIF
```

```
C     IF(OMGX(IX,JY,KZ) .GT. 1.00)THEN
C      OMGX(IX,JY,KZ) = 1.0D+00
C     ELSEIF(OMGX(IX,JY,KZ) .LT. AAX(IX,JY,KZ))THEN
C      OMGX(IX,JY,KZ) = AAX(IX,JY,KZ)
C     ENDIF
C     IF(OMOX(IX,JY,KZ) .GT. 1.00)THEN
C      OMOX(IX,JY,KZ) = 1.0D+00
C     ELSEIF(OMOX(IX,JY,KZ) .LT. AAX(IX,JY,KZ))THEN
C      OMOX(IX,JY,KZ) = AAX(IX,JY,KZ)
C     ENDIF
C     IF(OMWX(IX,JY,KZ) .GT. 1.00)THEN
C      OMWX(IX,JY,KZ) = 1.0D+00
C     ELSEIF(OMWX(IX,JY,KZ) .LT. AAX(IX,JY,KZ))THEN
C      OMWX(IX,JY,KZ) = AAX(IX,JY,KZ)
C     ENDIF

      IF(NY .NE. 1) THEN
      IF(JY .EQ. 1) THEN
        OMGY(IX,JY,KZ) = 1.0D+00
        OMOY(IX,JY,KZ) = 1.0D+00
        OMWY(IX,JY,KZ) = 1.0D+00
      ELSEIF(JY .GT. 1 .AND. JY .NE. NY)THEN
       DUMY1G = RPGP(IX,JY,KZ)*DGP(IX,JY,KZ)/VGP(IX,JY,KZ)
       DUMY2G = RPGP(IX,JY-1,KZ)*DGP(IX,JY-1,KZ)/VGP(IX,JY-1,KZ)
       DUMY3G = RPGP(IX,JY+1,KZ)*DGP(IX,JY+1,KZ)/VGP(IX,JY+1,KZ)

       DUMY1O = RPOP(IX,JY,KZ)*DOP(IX,JY,KZ)/VOP(IX,JY,KZ)
       DUMY2O = RPOP(IX,JY-1,KZ)*DOP(IX,JY-1,KZ)/VOP(IX,JY-1,KZ)
       DUMY3O = RPOP(IX,JY+1,KZ)*DOP(IX,JY+1,KZ)/VOP(IX,JY+1,KZ)

       DUMY1W = RPWP(IX,JY,KZ)*DWP(IX,JY,KZ)/VWP(IX,JY,KZ)
       DUMY2W = RPWP(IX,JY-1,KZ)*DWP(IX,JY-1,KZ)/VWP(IX,JY-1,KZ)
       DUMY3W = RPWP(IX,JY+1,KZ)*DWP(IX,JY+1,KZ)/VWP(IX,JY+1,KZ)

       OMGY(IX,JY,KZ)=1.-AAY(IX,JY,KZ)*((DUMY1G-DUMY2G)/(DUMY3G-DU
       OMOY(IX,JY,KZ)=1.-AAY(IX,JY,KZ)*((DUMY1O-DUMY2O)/(DUMY3O-DU
       OMWY(IX,JY,KZ)=1.-AAY(IX,JY,KZ)*((DUMY1W-DUMY2W)/(DUMY3W-DU

      ELSEIF(JY .EQ. NY)THEN
```

```
   OMGY(IX,JY,KZ) = 1.0D+00
   OMOY(IX,JY,KZ) = 1.0D+00
   OMWY(IX,JY,KZ) = 1.0D+00
ENDIF

IF(OMGY(IX,JY,KZ) .GT. 1.00)THEN
 OMGY(IX,JY,KZ) = 1.0D+00
ELSEIF(OMGY(IX,JY,KZ) .LT. AAY(IX,JY,KZ))THEN
 OMGY(IX,JY,KZ) = AAY(IX,JY,KZ)
ENDIF
IF(OMOY(IX,JY,KZ) .GT. 1.00)THEN
 OMOY(IX,JY,KZ) = 1.0D+00
ELSEIF(OMOY(IX,JY,KZ) .LT. AAY(IX,JY,KZ))THEN
 OMOY(IX,JY,KZ) = AAY(IX,JY,KZ)
ENDIF
IF(OMWY(IX,JY,KZ) .GT. 1.00)THEN
 OMWY(IX,JY,KZ) = 1.0D+00
ELSEIF(OMWY(IX,JY,KZ) .LT. AAY(IX,JY,KZ))THEN
 OMWY(IX,JY,KZ) = AAY(IX,JY,KZ)
ENDIF
ENDIF
IF(NZ .NE. 1) THEN
IF(KZ .EQ. 1) THEN
   OMGY(IX,JY,KZ) = 1.0D+00
   OMOY(IX,JY,KZ) = 1.0D+00
   OMWY(IX,JY,KZ) = 1.0D+00
ELSEIF(KZ .GT. 1 .AND. KZ .NE. NZ)THEN
 DUMY1G = RPGP(IX,JY,KZ)*DGP(IX,JY,KZ)/VGP(IX,JY,KZ)
 DUMY2G = RPGP(IX,JY,KZ-1)*DGP(IX,JY,KZ-1)/VGP(IX,JY,KZ-1)
 DUMY3G = RPGP(IX,JY,KZ+1)*DGP(IX,JY,KZ+1)/VGP(IX,JY,KZ+1)

 DUMY1O = RPOP(IX,JY,KZ)*DOP(IX,JY,KZ)/VOP(IX,JY,KZ)
 DUMY2O = RPOP(IX,JY,KZ-1)*DOP(IX,JY,KZ-1)/VOP(IX,JY,KZ-1)
 DUMY3O = RPOP(IX,JY,KZ+1)*DOP(IX,JY,KZ+1)/VOP(IX,JY,KZ+1)

 DUMY1W = RPWP(IX,JY,KZ)*DWP(IX,JY,KZ)/VWP(IX,JY,KZ)
 DUMY2W = RPWP(IX,JY,KZ-1)*DWP(IX,JY,KZ-1)/VWP(IX,JY,KZ-1)
 DUMY3W = RPWP(IX,JY,KZ+1)*DWP(IX,JY,KZ+1)/VWP(IX,JY,KZ+1)
```

```
       OMGZ(IX,JY,KZ)=1.-AAZ(IX,JY,KZ)*((DUMY1G-DUMY2G)/(DUMY3G-DU
       OMOZ(IX,JY,KZ)=1.-AAZ(IX,JY,KZ)*((DUMY1O-DUMY2O)/(DUMY3O-DU
       OMWZ(IX,JY,KZ)=1.-AAZ(IX,JY,KZ)*((DUMY1W-DUMY2W)/(DUMY3W-DU

      ELSEIF(KZ .EQ. NZ)THEN
        OMGY(IX,JY,KZ) = 1.0D+00
        OMOY(IX,JY,KZ) = 1.0D+00
        OMWY(IX,JY,KZ) = 1.0D+00
      ENDIF

      IF(OMGZ(IX,JY,KZ) .GT. 1.00)THEN
       OMGZ(IX,JY,KZ) = 1.0D+00
      ELSEIF(OMGZ(IX,JY,KZ) .LT. AAZ(IX,JY,KZ))THEN
       OMGZ(IX,JY,KZ) = AAZ(IX,JY,KZ)
      ENDIF
      IF(OMOZ(IX,JY,KZ) .GT. 1.00)THEN
       OMOZ(IX,JY,KZ) = 1.0D+00
      ELSEIF(OMOZ(IX,JY,KZ) .LT. AAZ(IX,JY,KZ))THEN
       OMOZ(IX,JY,KZ) = AAZ(IX,JY,KZ)
      ENDIF
      IF(OMWZ(IX,JY,KZ) .GT. 1.00)THEN
       OMWZ(IX,JY,KZ) = 1.0D+00
      ELSEIF(OMWZ(IX,JY,KZ) .LT. AAZ(IX,JY,KZ))THEN
       OMWZ(IX,JY,KZ) = AAZ(IX,JY,KZ)
      ENDIF
      ENDIF
      ELSE
        OMGX(IX,JY,KZ) = 1.0D+00
        OMOX(IX,JY,KZ) = 1.0D+00
        OMWX(IX,JY,KZ) = 1.0D+00
        OMGY(IX,JY,KZ) = 1.0D+00
        OMOY(IX,JY,KZ) = 1.0D+00
        OMWY(IX,JY,KZ) = 1.0D+00
        OMGZ(IX,JY,KZ) = 1.0D+00
        OMOZ(IX,JY,KZ) = 1.0D+00
        OMWZ(IX,JY,KZ) = 1.0D+00
      ENDIF
   C  WRITE(14,*)OMGY(IX,JY,KZ),OMOY(IX,JY,KZ),OMWY(IX,JY,KZ)
   C  WRITE(14,*)OMGX(IX,JY,KZ),OMOX(IX,JY,KZ),OMWX(IX,JY,KZ)
```

```
      RETURN
      END

      SUBROUTINE DARCY3(IX,JY,KZ)

C**** THIS ROUINE CALCULTES THE MOBILTY TERM USING WHEATELY METHOD


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      IF(IX .GT. 1 .AND. IX .NE. NX)THEN
        INX = IX + 1
        CALL PHYPRO(INX,JY,KZ)
      ENDIF
      IF(IX .EQ. 1)THEN
       GTRAN = APR*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)
       OTRAN = APR*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)
       WTRAN = APR*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)

       GDARC1 = GTRAN*(PG(IX+1,JY,KZ) - PG(IX,JY,KZ))/(DX*DX)
       GDARC2 = 0.00
       ODARC1 = OTRAN*(PO(IX+1,JY,KZ) - PO(IX,JY,KZ))/(DX*DX)
       ODARC2 = 0.00
       WDARC1 = WTRAN*(PW(IX+1,JY,KZ) - PW(IX,JY,KZ))/(DX*DX)
       WDARC2 = 0.00
      ELSEIF(IX .GT. 1 .AND. IX .NE. NX)THEN
       GTRAN1 = APR*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)
       GTRAN2 = APR*RPG(IX-1,JY,KZ)*DG(IX-1,JY,KZ)/VG(IX-1,JY,KZ)
       GTRAN3 = APR*RPG(IX+1,JY,KZ)*DG(IX+1,JY,KZ)/VG(IX+1,JY,KZ)

       GTRAN = OMGX(IX,JY,KZ)*GTRAN1 + (1.0 - OMGX(IX,JY,KZ))*GTRAN
       GTRANP = OMGX(IX,JY,KZ)*GTRAN2 + (1.0 - OMGX(IX,JY,KZ))*GTRA

       OTRAN1 = APR*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)
       OTRAN2 = APR*RPO(IX-1,JY,KZ)*DO(IX-1,JY,KZ)/VO(IX-1,JY,KZ)
       OTRAN3 = APR*RPO(IX+1,JY,KZ)*DO(IX+1,JY,KZ)/VO(IX+1,JY,KZ)

       OTRAN = OMOX(IX,JY,KZ)*OTRAN1 + (1.0 - OMOX(IX,JY,KZ))*OTRAN
```

```
 OTRANP = OMOX(IX,JY,KZ)*OTRAN2 + (1.0 - OMOX(IX,JY,KZ))*OTRA

 WTRAN1 = APR*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)
 WTRAN2 = APR*RPW(IX-1,JY,KZ)*DW(IX-1,JY,KZ)/VW(IX-1,JY,KZ)
 WTRAN3 = APR*RPW(IX+1,JY,KZ)*DW(IX+1,JY,KZ)/VW(IX+1,JY,KZ)

 WTRAN = OMWX(IX,JY,KZ)*WTRAN1 + (1.0 - OMWX(IX,JY,KZ))*WTRAN
 WTRANP = OMWX(IX,JY,KZ)*WTRAN2 + (1.0 - OMWX(IX,JY,KZ))*WTRA

 GDARC1 = GTRAN*(PG(IX+1,JY,KZ) - PG(IX,JY,KZ))/(DX*DX)
 GDARC2 = GTRANP*(PG(IX-1,JY,KZ) - PG(IX,JY,KZ))/(DX*DX)
 ODARC1 = OTRAN*(PO(IX+1,JY,KZ) - PO(IX,JY,KZ))/(DX*DX)
 ODARC2 = OTRANP*(PO(IX-1,JY,KZ) - PO(IX,JY,KZ))/(DX*DX)
 WDARC1 = WTRAN*(PW(IX+1,JY,KZ) - PW(IX,JY,KZ))/(DX*DX)
 WDARC2 = WTRANP*(PW(IX-1,JY,KZ) - PW(IX,JY,KZ))/(DX*DX)

ELSEIF(IX .EQ. NX) THEN
 GTRAN = APR*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)
 OTRAN = APR*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)
 WTRAN = APR*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)
 GTRANP = APR*RPG(IX-1,JY,KZ)*DG(IX-1,JY,KZ)/VG(IX-1,JY,KZ)
 OTRANP = APR*RPO(IX-1,JY,KZ)*DO(IX-1,JY,KZ)/VO(IX-1,JY,KZ)
 WTRANP = APR*RPW(IX-1,JY,KZ)*DW(IX-1,JY,KZ)/VW(IX-1,JY,KZ)

 GDARC1 = 0.00D+00
 GDARC2 = GTRANP*(PG(IX-1,JY,KZ) - PG(IX,JY,KZ))/(DX*DX)
 ODARC1 = 0.00D+00
 ODARC2 = OTRANP*(PO(IX-1,JY,KZ) - PO(IX,JY,KZ))/(DX*DX)
 WDARC1 = 0.00D+00
 WDARC2 = WTRANP*(PW(IX-1,JY,KZ) - PW(IX,JY,KZ))/(DX*DX)
ENDIF

IF(NY .NE. 1)THEN
IF(JY .GT. 1 .AND. JY .NE. NY)THEN
  JNY = JY + 1
  CALL PHYPRO(IX,JNY,KZ)
ENDIF

IF(JY .EQ. 1)THEN
```

```
 GTRANY = APR*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)
 OTRANY = APR*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)
 WTRANY = APR*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)

 GDAR1Y = GTRANY*(PG(IX,JY+1,KZ) - PG(IX,JY,KZ))/(DY*DY)
 GDAR2Y = 0.00
 ODAR1Y = OTRANY*(PO(IX,JY+1,KZ) - PO(IX,JY,KZ))/(DY*DY)
 ODAR2Y = 0.00
 WDAR1Y = WTRANY*(PW(IX,JY+1,KZ) - PW(IX,JY,KZ))/(DY*DY)
 WDAR2Y = 0.00
ELSEIF(JY .GT. 1 .AND. JY .NE. NY)THEN
 GTRA1Y = APR*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)
 GTRA2Y = APR*RPG(IX,JY-1,KZ)*DG(IX,JY-1,KZ)/VG(IX,JY-1,KZ)
 GTRA3Y = APR*RPG(IX,JY+1,KZ)*DG(IX,JY+1,KZ)/VG(IX,JY+1,KZ)

 GTRANY = OMGY(IX,JY,KZ)*GTRA1Y + (1.0 - OMGY(IX,JY,KZ))*GTRA
 GTRAPY = OMGY(IX,JY,KZ)*GTRA2Y + (1.0 - OMGY(IX,JY,KZ))*GTRA

 OTRA1Y = APR*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)
 OTRA2Y = APR*RPO(IX,JY-1,KZ)*DO(IX,JY-1,KZ)/VO(IX,JY-1,KZ)
 OTRA3Y = APR*RPO(IX,JY+1,KZ)*DO(IX,JY+1,KZ)/VO(IX,JY+1,KZ)

 OTRANY = OMOY(IX,JY,KZ)*OTRA1Y + (1.0 - OMOY(IX,JY,KZ))*OTRA
 OTRAPY = OMOY(IX,JY,KZ)*OTRA2Y + (1.0 - OMOY(IX,JY,KZ))*OTRA

 WTRA1Y = APR*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)
 WTRA2Y = APR*RPW(IX,JY-1,KZ)*DW(IX,JY-1,KZ)/VW(IX,JY-1,KZ)
 WTRA3Y = APR*RPW(IX,JY+1,KZ)*DW(IX,JY+1,KZ)/VW(IX,JY+1,KZ)

 WTRANY = OMWY(IX,JY,KZ)*WTRA1Y + (1.0 - OMWY(IX,JY,KZ))*WTRA
 WTRAPY = OMWY(IX,JY,KZ)*WTRA2Y + (1.0 - OMWY(IX,JY,KZ))*WTRA

 GDAR1Y = GTRANY*(PG(IX,JY+1,KZ) - PG(IX,JY,KZ))/(DY*DY)
 GDAR2Y = GTRAPY*(PG(IX,JY-1,KZ) - PG(IX,JY,KZ))/(DY*DY)
 ODAR1Y = OTRANY*(PO(IX,JY+1,KZ) - PO(IX,JY,KZ))/(DY*DY)
 ODAR2Y = OTRAPY*(PO(IX,JY-1,KZ) - PO(IX,JY,KZ))/(DY*DY)
 WDAR1Y = WTRANY*(PW(IX,JY+1,KZ) - PW(IX,JY,KZ))/(DY*DY)
 WDARC2 = WTRAPY*(PW(IX,JY-1,KZ) - PW(IX,JY,KZ))/(DY*DY)
ELSEIF(JY .EQ. NY) THEN
```

```
 GTRANY = APR*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)
 OTRANY = APR*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)
 WTRANY = APR*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)
 GTRAPY = APR*RPG(IX,JY-1,KZ)*DG(IX,JY-1,KZ)/VG(IX,JY-1,KZ)
 OTRAPY = APR*RPO(IX,JY-1,KZ)*DO(IX,JY-1,KZ)/VO(IX,JY-1,KZ)
 WTRAPY = APR*RPW(IX,JY-1,KZ)*DW(IX,JY-1,KZ)/VW(IX,JY-1,KZ)

 GDAR1Y = 0.00D+00
 GDAR2Y = GTRAPY*(PG(IX,JY-1,KZ) - PG(IX,JY,KZ))/(DY*DY)
 ODAR1Y = 0.00D+00
 ODAR2Y = OTRAPY*(PO(IX,JY-1,KZ) - PO(IX,JY,KZ))/(DY*DY)
 WDAR1Y = 0.00D+00
 WDAR2Y = WTRAPY*(PW(IX,JY-1,KZ) - PW(IX,JY,KZ))/(DY*DY)
ENDIF
ELSE
 GDAR1Y = 0.00D+00
 GDAR2Y = 0.00D+00
 ODAR1Y = 0.00D+00
 ODAR2Y = 0.00D+00
 WDAR1Y = 0.00D+00
 WDAR2Y = 0.00D+00
ENDIF

IF(NZ .NE. 1)THEN
IF(KZ .GT. 1 .AND. KZ .NE. NZ)THEN
  KNZ = KZ + 1
  CALL PHYPRO(IX,JY,KNZ)
ENDIF

IF(KZ .EQ. 1)THEN
 GTRANZ = APR*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)
 OTRANZ = APR*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)
 WTRANZ = APR*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)

 GDAR1Y = GTRANZ*(PG(IX,JY,KZ+1) - PG(IX,JY,KZ))/(DZ*DZ)
 GDAR2Z = 0.00
 ODAR1Z = OTRANZ*(PO(IX,JY,KZ+1) - PO(IX,JY,KZ))/(DZ*DZ)
 ODAR2Z = 0.00
 WDAR1Z = WTRANZ*(PW(IX,JY,KZ+1) - PW(IX,JY,KZ))/(DZ*DZ)
```

```
 WDAR2Z = 0.00
ELSEIF(KZ .GT. 1 .AND. KZ .NE. NZ)THEN
 GTRA1Z = APR*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)
 GTRA2Z = APR*RPG(IX,JY,KZ-1)*DG(IX,JY,KZ-1)/VG(IX,JY,KZ-1)
 GTRA3Z = APR*RPG(IX,JY,KZ+1)*DG(IX,JY,KZ+1)/VG(IX,JY,KZ+1)

 GTRANZ = OMGZ(IX,JY,KZ)*GTRA1Z + (1.0 - OMGY(IX,JY,KZ))*GTRA
 GTRAPZ = OMGZ(IX,JY,KZ)*GTRA2Z + (1.0 - OMGY(IX,JY,KZ))*GTRA

 OTRA1Z = APR*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)
 OTRA2Z = APR*RPO(IX,JY,KZ-1)*DO(IX,JY,KZ-1)/VO(IX,JY,KZ-1)
 OTRA3Z = APR*RPO(IX,JY,KZ+1)*DO(IX,JY,KZ+1)/VO(IX,JY,KZ+1)

 OTRANZ = OMOZ(IX,JY,KZ)*OTRA1Z + (1.0 - OMOZ(IX,JY,KZ))*OTRA
 OTRAPZ = OMOZ(IX,JY,KZ)*OTRA2Z + (1.0 - OMOZ(IX,JY,KZ))*OTRA

 WTRA1Z = APR*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)
 WTRA2Z = APR*RPW(IX,JY,KZ-1)*DW(IX,JY,KZ-1)/VW(IX,JY,KZ-1)
 WTRA3Z = APR*RPW(IX,JY,KZ+1)*DW(IX,JY,KZ+1)/VW(IX,JY,KZ+1)

 WTRANZ = OMWZ(IX,JY,KZ)*WTRA1Z + (1.0 - OMWZ(IX,JY,KZ))*WTRA
 WTRAPZ = OMWZ(IX,JY,KZ)*WTRA2Z + (1.0 - OMWZ(IX,JY,KZ))*WTRA

 GDAR1Z = GTRANZ*(PG(IX,JY,KZ+1) - PG(IX,JY,KZ))/(DZ*DZ)
 GDAR2Z = GTRAPZ*(PG(IX,JY,KZ-1) - PG(IX,JY,KZ))/(DZ*DZ)
 ODAR1Z = OTRANZ*(PO(IX,JY,KZ+1) - PO(IX,JY,KZ))/(DZ*DZ)
 ODAR2Z = OTRAPZ*(PO(IX,JY,KZ-1) - PO(IX,JY,KZ))/(DZ*DZ)
 WDAR1Z = WTRANZ*(PW(IX,JY,KZ+1) - PW(IX,JY,KZ))/(DZ*DZ)
 WDAR2Z = WTRAPZ*(PW(IX,JY,KZ-1) - PW(IX,JY,KZ))/(DZ*DZ)
ELSEIF(KZ .EQ. NZ) THEN
 GTRANZ = APR*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)
 OTRANZ = APR*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)
 WTRANZ = APR*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)
 GTRAPZ = APR*RPG(IX,JY,KZ-1)*DG(IX,JY,KZ-1)/VG(IX,JY,KZ-1)
 OTRAPZ = APR*RPO(IX,JY,KZ-1)*DO(IX,JY,KZ-1)/VO(IX,JY,KZ-1)
 WTRAPZ = APR*RPW(IX,JY,KZ-1)*DW(IX,JY,KZ-1)/VW(IX,JY,KZ-1)

 GDAR1Z = 0.00D+00
 GDAR2Z = GTRAPZ*(PG(IX,JY,KZ-1) - PG(IX,JY,KZ))/(DZ*DZ)
```

```
       ODAR1Z = 0.00D+00
       ODAR2Z = OTRAPZ*(PO(IX,JY,KZ-1) - PO(IX,JY,KZ))/(DZ*DZ)
       WDAR1Z = 0.00D+00
       WDAR2Z = WTRAPZ*(PW(IX,JY,KZ-1) - PW(IX,JY,KZ))/(DZ*DZ)
      ENDIF
      ELSE
       GDAR1Z = 0.00D+00
       GDAR2Z = 0.00D+00
       ODAR1Z = 0.00D+00
       ODAR2Z = 0.00D+00
       WDAR1Z = 0.00D+00
       WDAR2Z = 0.00D+00
      ENDIF
      RETURN
      END

      SUBROUTINE INERTS(IX,JY,KZ)

C***** THIS ROUTINE CALCULTES THE INERTS GAS MASS BALANCE


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      IF (IX .EQ. 1) THEN
       DARCY = GDARC1*YN(IX,JY,KZ)
      ELSEIF (IX .GT. 1 .AND. IX .NE. NX) THEN
       DARCY = (GDARC1*YN(IX,JY,KZ) + GDARC2*YN(IX-1,JY,KZ))
      ELSEIF (IX .EQ. NX) THEN
       DARCY = GDARC2*YN(IX-1,JY,KZ)
      ENDIF
      IF (NY .NE. 1) THEN
      IF (JY .EQ. 1) THEN
       DARCYY = GDAR1Y*YN(IX,JY,KZ)
       GRAVYY = 0.00
      ELSEIF (JY .GT. 1 .AND. JY .NE. NY) THEN
       DARCYY = (GDAR1Y*YN(IX,JY,KZ) + GDAR2Y*YN(IX,JY-1,KZ))
       GRAVYY = GGRAV1*YN(IX,JY,KZ) - GGRAV2*YN(IX,JY-1,KZ)
      ELSEIF (JY .EQ. NY) THEN
       DARCYY = GDAR2Y*YN(IX,JY-1,KZ)
```

```
C      GRAVYY = GGRAV2*YN(IX,JY-1,KZ)
        GRAVYY = GGRAV1*YN(IX,JY,KZ) - GGRAV2*YN(IX,JY-1,KZ)
       ENDIF
       ELSE
       DARCYY = 0.00
        GRAVYY = 0.0
       ENDIF

       IF (NZ .NE. 1) THEN
       IF (KZ .EQ. 1) THEN
        DARCYZ = GDAR1Z*YN(IX,JY,KZ)
       ELSEIF (KZ .GT. 1 .AND. KZ .NE. NZ) THEN
        DARCYZ = (GDAR1Z*YN(IX,JY,KZ) + GDAR2Z*YN(IX,JY,KZ-1))
       ELSEIF (KZ .EQ. NZ) THEN
        DARCYZ = GDAR2Z*YN(IX,JY,KZ-1)
       ENDIF
       ELSE
       DARCYZ = 0.00
       ENDIF
        ACUM = (DG(IX,JY,KZ)*SG(IX,JY,KZ)*YN(IX,JY,KZ) -
     +        DGP(IX,JY,KZ)*SGP(IX,JY,KZ)*YNP(IX,JY,KZ))*POROS/DT
        REAC = S2*RA(IX,JY,KZ) + S5*RB(IX,JY,KZ) + S8*RC(IX,JY,KZ)
     +     + S11*RD(IX,JY,KZ)
        OTNR(IX,JY,KZ) = YN(IX,JY,KZ)*GTRAN*OTCN(IX,JY,KZ)
     +              *(PG(IX,JY,KZ) - WLP)

        EALPG = QN(IX,JY,KZ) + REAC - (ACUM - DARCY - DARCYY
     +        - DARCYZ + OTNR(IX,JY,KZ)) - GRAVYY

        RETURN
        END

        SUBROUTINE WATER(IX,JY,KZ)

C***** THIS ROUTINE CALCULTES THE WATER MASS BALANCE


        INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
```

```
      IF (IX .EQ. 1) THEN
       DARCY = GDARC1*YW(IX,JY,KZ) + WDARC1
      ELSEIF (IX .GT. 1 .AND. IX .NE. NX) THEN
       DARCY = GDARC1*YW(IX,JY,KZ) + GDARC2*YW(IX-1,JY,KZ) + WDARC1
     +      + WDARC2
      ELSEIF (IX .EQ. NX) THEN
       DARCY = GDARC2*YW(IX-1,JY,KZ) + WDARC2
      ENDIF
      IF (NY .NE. 1) THEN
      IF (JY .EQ. 1) THEN
       DARCYY = GDAR1Y*YW(IX,JY,KZ) + WDAR1Y
       GRAVYY = 0.0
      ELSEIF (JY .GT. 1 .AND. JY .NE. NY) THEN
       DARCYY = GDAR1Y*YW(IX,JY,KZ) + GDAR2Y*YW(IX,JY-1,KZ) + WDAR1
     +       +  WDAR2Y
       GRAVYY = GGRAV1*YW(IX,JY,KZ) - GGRAV2*YW(IX,JY-1,KZ) + WGRAV
     +       - WGRAV2
      ELSEIF (JY .EQ. NY) THEN
       DARCYY = GDAR2Y*YW(IX,JY-1,KZ) + WDAR2Y
       GRAVYY = GGRAV1*YW(IX,JY,KZ) - GGRAV2*YW(IX,JY-1,KZ) + WGRAV
     +       - WGRAV2
C      GRAVYY = GGRAV2*YW(IX,JY-1,KZ) + WGRAV2
      ENDIF
      ELSE
      DARCYY = 0.00
       GRAVYY = 0.0
      ENDIF

      IF (NZ .NE. 1) THEN
      IF (KZ .EQ. 1) THEN
       DARCYZ = GDAR1Z*YW(IX,JY,KZ) + WDAR1Z
      ELSEIF (KZ .GT. 1 .AND. KZ .NE. NZ) THEN
       DARCYZ = GDAR1Z*YW(IX,JY,KZ) + GDAR2Z*YW(IX,JY,KZ-1) + WDAR1
     +       +  WDAR2Z
      ELSEIF (KZ .EQ. NZ) THEN
       DARCYZ = GDAR2Z*YW(IX,JY,KZ-1) + WDAR2Z
      ENDIF
      ELSE
      DARCYZ = 0.00
```

```
      ENDIF
       ACUM = ((DG(IX,JY,KZ)*SG(IX,JY,KZ)*YW(IX,JY,KZ) -
     +      DGP(IX,JY,KZ)*SGP(IX,JY,KZ)*YWP(IX,JY,KZ))
     +    +  (DW(IX,JY,KZ)*SW(IX,JY,KZ)-DWP(IX,JY,KZ)*SWP(IX,JY,K
     +    *POROS/DT
       REAC = S3*RA(IX,JY,KZ) + S6*RB(IX,JY,KZ) + S12*RD(IX,JY,KZ)
       OTWTV(IX,JY,KZ) = GTRAN*YW(IX,JY,KZ)*OTCN(IX,JY,KZ)
     +           *(PG(IX,JY,KZ) - WLP)
       OTWTL(IX,JY,KZ) = WTRAN*OTCN(IX,JY,KZ)*(PW(IX,JY,KZ) - WLP)
       OTWT = OTWTV(IX,JY,KZ) + OTWTL(IX,JY,KZ)

       EALSW = QW(IX,JY,KZ) + REAC - (ACUM - DARCY - DARCYY
     +      - DARCYZ + OTWT) - GRAVYY
      RETURN
      END

      SUBROUTINE LIGHT(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE LIGHT OIL MASS BALANCE


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      IF (IX .EQ. 1) THEN
       DARCY = GDARC1*YL(IX,JY,KZ) + ODARC1*XL(IX,JY,KZ)
      ELSEIF (IX .GT. 1 .AND. IX .NE. NX) THEN
       DARCY = GDARC1*YL(IX,JY,KZ) + GDARC2*YL(IX-1,JY,KZ)
     +       + ODARC1*XL(IX,JY,KZ) + ODARC2*XL(IX-1,JY,KZ)
      ELSEIF (IX .EQ. NX) THEN
       DARCY = GDARC2*YL(IX-1,JY,KZ) + ODARC2*XL(IX-1,JY,KZ)
      ENDIF
      IF (NY .NE. 1) THEN
      IF (JY .EQ. 1) THEN
       DARCYY = GDAR1Y*YL(IX,JY,KZ) + ODAR1Y*XL(IX,JY,KZ)
       GRAVYY = 0.0
      ELSEIF (JY .GT. 1 .AND. JY .NE. NY) THEN
       DARCYY = GDAR1Y*YL(IX,JY,KZ) + GDAR2Y*YL(IX,JY-1,KZ)
     +       + ODAR1Y*XL(IX,JY,KZ) + ODAR2Y*XL(IX,JY-1,KZ)
       GRAVYY = GGRAV1*YL(IX,JY,KZ) + OGRAV1*XL(IX,JY,KZ)
```

```
     +      - GGRAV2*YL(IX,JY-1,KZ) - OGRAV2*XL(IX,JY-1,KZ)
      ELSEIF (JY .EQ. NY) THEN
       DARCYY = GDAR2Y*YL(IX,JY-1,KZ) + ODAR2Y*XL(IX,JY-1,KZ)
       GRAVYY = GGRAV1*YL(IX,JY,KZ) + OGRAV1*XL(IX,JY,KZ)
     +      - GGRAV2*YL(IX,JY-1,KZ) - OGRAV2*XL(IX,JY-1,KZ)
C      GRAVYY = GGRAV2*YL(IX,JY-1,KZ) + OGRAV2*XL(IX,JY-1,KZ)
      ENDIF
      ELSE
       DARCYY = 0.00
       GRAVYY = 0.0
      ENDIF

      IF (NZ .NE. 1) THEN
      IF (KZ .EQ. 1) THEN
       DARCYZ = GDAR1Z*YL(IX,JY,KZ) + ODAR1Z*XL(IX,JY,KZ)
      ELSEIF (KZ .GT. 1 .AND. KZ .NE. NZ) THEN
       DARCYZ = GDAR1Z*YL(IX,JY,KZ) + GDAR2Z*YL(IX,JY,KZ-1)
     +        + ODAR1Z*XL(IX,JY,KZ) + ODAR2Z*XL(IX,JY,KZ-1)
      ELSEIF (KZ .EQ. NZ) THEN
       DARCYZ = GDAR2Z*YL(IX,JY,KZ-1) + ODAR2Z*XL(IX,JY,KZ-1)
      ENDIF
      ELSE
       DARCYZ = 0.00
      ENDIF
       ACUM = ((DG(IX,JY,KZ)*SG(IX,JY,KZ)*YL(IX,JY,KZ) -
     +        DGP(IX,JY,KZ)*SGP(IX,JY,KZ)*YLP(IX,JY,KZ)) +
     +       (DO(IX,JY,KZ)*SO(IX,JY,KZ)*XL(IX,JY,KZ) -
     +        DOP(IX,JY,KZ)*SOP(IX,JY,KZ)*XLP(IX,JY,KZ)))*POROS/DT
       REAC = S7*RC(IX,JY,KZ) - RA(IX,JY,KZ)
       OTLTV(IX,JY,KZ) = GTRAN*YL(IX,JY,KZ)*OTCN(IX,JY,KZ)
     +            *(PG(IX,JY,KZ) - WLP)
       OTLTL(IX,JY,KZ) = OTRAN*XL(IX,JY,KZ)*OTCN(IX,JY,KZ)
     +            *(PO(IX,JY,KZ) - WLP)
       OTLT = OTLTV(IX,JY,KZ) + OTLTL(IX,JY,KZ)

       EALSO = QL(IX,JY,KZ) + REAC - (ACUM - DARCY - DARCYY
     +      - DARCYZ + OTLT) - GRAVYY

      RETURN
```

```
      END

      SUBROUTINE ENERGY(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE HEAT BALANCE


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      IF (IX .EQ. 1) THEN
       CONV = GDARC1*HG(IX,JY,KZ) + ODARC1*HO(IX,JY,KZ)
     +     + WDARC1*HW(IX,JY,KZ)
       COND = THCO*(TM(IX+1,JY,KZ) - TM(IX,JY,KZ))/DX**2
      ELSEIF (IX .GT. 1 .AND. IX .NE. NX) THEN
       CONV = GDARC1*HG(IX,JY,KZ) + GDARC2*HG(IX-1,JY,KZ) +
     +          ODARC1*HO(IX,JY,KZ) + ODARC2*HO(IX-1,JY,KZ) +
     +          WDARC1*HW(IX,JY,KZ) + WDARC2*HW(IX-1,JY,KZ)
       COND = THCO*((TM(IX+1,JY,KZ)-TM(IX,JY,KZ)) +
     +       (TM(IX-1,JY,KZ)-TM(IX,JY,KZ)))/DX**2
      ELSEIF (IX .EQ. NX) THEN
       CONV = GDARC2*HG(IX-1,JY,KZ) + ODARC2*HO(IX-1,JY,KZ) +
     +          WDARC2*HW(IX-1,JY,KZ)
       COND = THCO*(TM(IX-1,JY,KZ) - TM(IX,JY,KZ))/DX**2
      ENDIF
      IF (NY .NE. 1) THEN
      IF (JY .EQ. 1) THEN
       CONVY = GDAR1Y*HG(IX,JY,KZ) + ODAR1Y*HO(IX,JY,KZ)
     +      + WDAR1Y*HW(IX,JY,KZ)
       GRAVY = 0.0
       CONDY = THCO*(TM(IX,JY+1,KZ) - TM(IX,JY,KZ))/DY**2
      ELSEIF (JY .GT. 1 .AND. JY .NE. NY) THEN
       CONVY = GDAR1Y*HG(IX,JY,KZ) + GDAR2Y*HG(IX,JY-1,KZ) +
     +          ODAR1Y*HO(IX,JY,KZ) + ODAR2Y*HO(IX,JY-1,KZ) +
     +          WDAR1Y*HW(IX,JY,KZ) + WDAR2Y*HW(IX,JY-1,KZ)
       GRAVY = GGRAV1*HG(IX,JY,KZ) - GGRAV2*HG(IX,JY-1,KZ) +
     +        OGRAV1*HO(IX,JY,KZ) - OGRAV2*HO(IX,JY-1,KZ) +
     +        WGRAV1*HW(IX,JY,KZ) - WGRAV2*HW(IX,JY-1,KZ)
       CONDY=THCO*((TM(IX,JY+1,KZ)-TM(IX,JY,KZ))+(TM(IX,JY-1,KZ)-
     +       TM(IX,JY,KZ)))/DY**2
```

```
      ELSEIF (JY .EQ. NY) THEN
       CONVY = GDAR2Y*HG(IX,JY-1,KZ) + ODAR2Y*HO(IX,JY-1,KZ) +
     +          WDAR2Y*HW(IX,JY-1,KZ)
       GRAVY = GGRAV1*HG(IX,JY,KZ) - GGRAV2*HG(IX,JY-1,KZ) +
     +        OGRAV1*HO(IX,JY,KZ) - OGRAV2*HO(IX,JY-1,KZ) +
     +        WGRAV1*HW(IX,JY,KZ) - WGRAV2*HW(IX,JY-1,KZ)
C      GRAVY = GGRAV2*HG(IX,JY-1,KZ) + OGRAV2*HO(IX,JY-1,KZ) +
C    +          WGRAV2*HW(IX,JY-1,KZ)
       CONDY = THCO*(TM(IX,JY-1,KZ) - TM(IX,JY,KZ))/DY**2
      ENDIF
      ELSE
        CONDY = 0.00
        CONVY = 0.00
        GRAVY = 0.00
      ENDIF
      IF (NZ .NE. 1) THEN
      IF (KZ .EQ. 1) THEN
       CONVZ = GDAR1Z*HG(IX,JY,KZ) + ODAR1Z*HO(IX,JY,KZ)
     +      + WDAR1Z*HW(IX,JY,KZ)
       CONDZ = THCO*(TM(IX,JY,KZ+1) - TM(IX,JY,KZ))/DZ**2
      ELSEIF (KZ .GT. 1 .AND. KZ .NE. NZ) THEN
       CONVZ = GDAR1Z*HG(IX,JY,KZ) + GDAR2Z*HG(IX,JY,KZ-1) +
     +          ODAR1Z*HO(IX,JY,KZ) + ODAR2Z*HO(IX,JY,KZ-1) +
     +          WDAR1Z*HW(IX,JY,KZ) + WDAR2Z*HW(IX,JY,KZ-1)
       CONDZ=THCO*((TM(IX,JY,KZ+1)-TM(IX,JY,KZ))+(TM(IX,JY,KZ-1)-
     +      TM(IX,JY,KZ)))/DZ**2
      ELSEIF (KZ .EQ. NZ) THEN
       CONVZ = GDAR2Z*HG(IX,JY,KZ-1) + ODAR2Z*HO(IX,JY,KZ-1) +
     +          WDAR2Z*HW(IX,JY,KZ-1)
       CONDZ = THCO*(TM(IX,JY,KZ-1) - TM(IX,JY,KZ))/DZ**2
      ENDIF
      ELSE
        CONDZ = 0.00
        CONVZ = 0.00
      ENDIF
       ACUM = ((1-POROS)*(ER(IX,JY,KZ) - ERP(IX,JY,KZ)) +
     +        (HC(IX,JY,KZ)*CC(IX,JY,KZ) - HCP(IX,JY,KZ)*CCP(IX,JY,
     +       + POROS*((DG(IX,JY,KZ)*EG(IX,JY,KZ)*SG(IX,JY,KZ) +
     +         DO(IX,JY,KZ)*EO(IX,JY,KZ)*SO(IX,JY,KZ)+
```

```
     +        DW(IX,JY,KZ)*EW(IX,JY,KZ)*SW(IX,JY,KZ)) -
     +        (DGP(IX,JY,KZ)*EGP(IX,JY,KZ)*SGP(IX,JY,KZ) +
     +        DOP(IX,JY,KZ)*EOP(IX,JY,KZ)*SOP(IX,JY,KZ) +
     +        DWP(IX,JY,KZ)*EWP(IX,JY,KZ)*SWP(IX,JY,KZ))))/DT

      HREAC(IX,JY,KZ) = HA*RA(IX,JY,KZ) + HB*RB(IX,JY,KZ)
     +              + HRC*RC(IX,JY,KZ) + HD*RD(IX,JY,KZ)
      OTGS = OTNR(IX,JY,KZ) + OTOX(IX,JY,KZ) + OTLTV(IX,JY,KZ)
     +    + OTHVV(IX,JY,KZ) + OTWTV(IX,JY,KZ)
      OTOL = OTLTL(IX,JY,KZ) + OTHVL(IX,JY,KZ)
      HOUT = OTGS*HG(IX,JY,KZ) + OTOL*HO(IX,JY,KZ)
     +     + OTWTL(IX,JY,KZ)*HW(IX,JY,KZ)
C       CALL HTLOSS(IX,JY,KZ)
      HLOSS(IX,JY,KZ) = 0.00
      EALTM =HIN(IX,JY,KZ) + HREAC(IX,JY,KZ) - (ACUM + HOUT
     +      - CONV - COND) + CONVZ + CONDZ
     +      - HLOSS(IX,JY,KZ)/DT + CONVY + CONDY - GRAVY
     RETURN
     END

     SUBROUTINE OXYGEN(IX,JY,KZ)

C***** THIS ROUTINE CALCULTES THE OXYGEN MASS BALANCE


     INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

     IF (IX .EQ. 1) THEN
      DARCY = GDARC1*YO(IX,JY,KZ)
     ELSEIF (IX .GT. 1 .AND. IX .NE. NX) THEN
      DARCY = (GDARC1*YO(IX,JY,KZ) + GDARC2*YO(IX-1,JY,KZ))
     ELSEIF (IX .EQ. NX) THEN
      DARCY = GDARC2*YO(IX-1,JY,KZ)
     ENDIF
     IF (NY .NE. 1) THEN
     IF (JY .EQ. 1) THEN
      DARCYY = GDAR1Y*YO(IX,JY,KZ)
      GRAVYY = 0.0
     ELSEIF (JY .GT. 1 .AND. JY .NE. NY) THEN
```

```
       DARCYY = (GDAR1Y*YO(IX,JY,KZ) + GDAR2Y*YO(IX,JY-1,KZ))
       GRAVYY = GGRAV1*YO(IX,JY,KZ) - GGRAV2*YO(IX,JY-1,KZ)
      ELSEIF (JY .EQ. NY) THEN
       DARCYY = GDAR2Y*YO(IX,JY-1,KZ)
C      GRAVYY = GGRAV2*YO(IX,JY-1,KZ)
       GRAVYY = GGRAV1*YO(IX,JY,KZ) - GGRAV2*YO(IX,JY-1,KZ)
      ENDIF
      ELSE
        DARCYY = 0.00
       GRAVYY = 0.0
      ENDIF
      IF (NZ .NE. 1) THEN
      IF (KZ .EQ. 1) THEN
       DARCYZ = GDAR1Z*YO(IX,JY,KZ)
      ELSEIF (KZ .GT. 1 .AND. KZ .NE. NZ) THEN
       DARCYZ = (GDAR1Z*YO(IX,JY,KZ) + GDAR2Z*YO(IX,JY,KZ-1))
      ELSEIF (KZ .EQ. NZ) THEN
       DARCYZ = GDAR2Z*YO(IX,JY,KZ-1)
      ENDIF
      ELSE
        DARCYZ = 0.00
      ENDIF

       ACUM = (DG(IX,JY,KZ)*SG(IX,JY,KZ)*YO(IX,JY,KZ)
     +     - DGP(IX,JY,KZ)*SGP(IX,JY,KZ)*YOP(IX,JY,KZ))*POROS/DT
       REAC = - (S1*RA(IX,JY,KZ) + S4*RB(IX,JY,KZ) + S10*RD(IX,JY,K
       OTOX(IX,JY,KZ) = YO(IX,JY,KZ)*GTRAN*OTCN(IX,JY,KZ)
     +            *(PG(IX,JY,KZ) - WLP)
       EALYO = QO(IX,JY,KZ) + REAC - (ACUM - DARCY - DARCYY
     +      + OTOX(IX,JY,KZ)) + DARCYZ - GRAVYY
      RETURN
      END

      SUBROUTINE HEAVY(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE HEAVY OIL MASS BALANCE


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
```

```
      IF (IX .EQ. 1) THEN
       DARCY = GDARC1*YH(IX,JY,KZ) + ODARC1*XH(IX,JY,KZ)
      ELSEIF (IX .GT. 1 .AND. IX .NE. NX) THEN
       DARCY = GDARC1*YH(IX,JY,KZ) + GDARC2*YH(IX-1,JY,KZ)
     +         + ODARC1*XH(IX,JY,KZ) + ODARC2*XH(IX-1,JY,KZ)
      ELSEIF (IX .EQ. NX) THEN
       DARCY = GDARC2*YH(IX-1,JY,KZ) + ODARC2*XH(IX-1,JY,KZ)
      ENDIF
      IF (NY .NE. 1)THEN
      IF (JY .EQ. 1) THEN
       DARCYY = GDAR1Y*YH(IX,JY,KZ) + ODAR1Y*XH(IX,JY,KZ)
       GRAVYY = 0.0
      ELSEIF (JY .GT. 1 .AND. JY .NE. NY) THEN
       DARCYY = GDAR1Y*YH(IX,JY,KZ) + GDAR2Y*YH(IX,JY-1,KZ)
     +         + ODAR1Y*XH(IX,JY,KZ) + ODAR2Y*XH(IX,JY-1,KZ)
       GRAVYY = GGRAV1*YH(IX,JY,KZ) + OGRAV1*XH(IX,JY,KZ)
     +       - GGRAV2*YH(IX,JY-1,KZ) - OGRAV2*XH(IX,JY-1,KZ)
      ELSEIF (JY .EQ. NY) THEN
       DARCYY = GDAR2Y*YH(IX,JY-1,KZ) + ODAR2Y*XH(IX,JY-1,KZ)
C      GRAVYY = GGRAV2*YH(IX,JY-1,KZ) + OGRAV2*XH(IX,JY-1,KZ)
       GRAVYY = GGRAV1*YH(IX,JY,KZ) + OGRAV1*XH(IX,JY,KZ)
     +       - GGRAV2*YH(IX,JY-1,KZ) - OGRAV2*XH(IX,JY-1,KZ)
      ENDIF
      ELSE
        DARCYY = 0.00
       GRAVYY = 0.0
      ENDIF
      IF (NZ .NE. 1)THEN
      IF (KZ .EQ. 1) THEN
       DARCYZ = GDAR1Z*YH(IX,JY,KZ) + ODAR1Z*XH(IX,JY,KZ)
      ELSEIF (KZ .GT. 1 .AND. KZ .NE. NZ) THEN
       DARCYZ = GDAR1Z*YH(IX,JY,KZ) + GDAR2Z*YH(IX,JY,KZ-1)
     +         + ODAR1Z*XH(IX,JY,KZ) + ODAR2Z*XH(IX,JY,KZ-1)
      ELSEIF (KZ .EQ. NZ) THEN
       DARCYZ = GDAR2Z*YH(IX,JY,KZ-1) + ODAR2Z*XH(IX,JY,KZ-1)
      ENDIF
      ELSE
        DARCYZ = 0.00
      ENDIF
```

```
       ACUM = ((DG(IX,JY,KZ)*SG(IX,JY,KZ)*YH(IX,JY,KZ)
     +     -  DGP(IX,JY,KZ)*SGP(IX,JY,KZ)*YHP(IX,JY,KZ))
     +     + (DO(IX,JY,KZ)*SO(IX,JY,KZ)*XH(IX,JY,KZ)
     +     -  DOP(IX,JY,KZ)*SOP(IX,JY,KZ)*XHP(IX,JY,KZ)))*POROS/D
       REAC = - (RB(IX,JY,KZ) + RC(IX,JY,KZ))
       OTHVV(IX,JY,KZ) = GTRAN*YH(IX,JY,KZ)*OTCN(IX,JY,KZ)
     +                *(PG(IX,JY,KZ) - WLP)
       OTHVL(IX,JY,KZ) = OTRAN*XH(IX,JY,KZ)*OTCN(IX,JY,KZ)
     +                *(PO(IX,JY,KZ) - WLP)
       OTHV = OTHVV(IX,JY,KZ) + OTHVL(IX,JY,KZ)
       EALXH = QH(IX,JY,KZ) + REAC - (ACUM - DARCY - DARCYY + OTHV)
     +       - GRAVYY + DARCYZ
      RETURN
      END

      SUBROUTINE REACT(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE REACTION RATES


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      CALL REAC1(IX,JY,KZ)
      CALL REAC2(IX,JY,KZ)
      CALL REAC3(IX,JY,KZ)
      CALL REAC4(IX,JY,KZ)

      RETURN
      END

      SUBROUTINE REAC1(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

C... THE REACTION IS:
C
C...  LTOIL + S1O2 -------> S2COx + S3H2O
C
        DUMY1 = ARA*EXP(-EA/(R*TM(IX,JY,KZ)))
        RA(IX,JY,KZ) = DUMY1*YO(IX,JY,KZ)*PG(IX,JY,KZ)
```

```
     +               *POROS*SO(IX,JY,KZ)*DO(IX,JY,KZ)*XL(IX,JY,KZ)
      RETURN
      END

      SUBROUTINE REAC2(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

C... THE REACTION IS:
C
C... HVOIL + S4O2 -------> S5COx + S6H20
C
        DUMY1 = ARB*EXP(-EB/(R*TM(IX,JY,KZ)))
        RB(IX,JY,KZ) =  DUMY1*YO(IX,JY,KZ)*PG(IX,JY,KZ)
     +               *POROS*SO(IX,JY,KZ)*DO(IX,JY,KZ)*XH(IX,JY,KZ)

      RETURN
      END

      SUBROUTINE REAC3(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

C... THE REACTION IS
C
C... HVOIL ------->S7LTOIL + S8COKE + S9INERTS
C
       DUM = ARC*EXP(-EC/(R*TM(IX,JY,KZ)))
       RC(IX,JY,KZ) =
     +          DUM*POROS*SO(IX,JY,KZ)*DO(IX,JY,KZ)*XH(IX,JY,KZ)*
     +          (1.0-(CC(IX,JY,KZ)/CKMAX)**5.)

      RETURN
      END

      SUBROUTINE REAC4(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

C... THE REACTION IS:
C
C... COKE + S10O2 ------> S11COx + S12H2O
```

```
C
        DUMY1 = ARD*EXP(-ED/(R*TM(IX,JY,KZ)))
        RD(IX,JY,KZ) = DUMY1*YO(IX,JY,KZ)*PG(IX,JY,KZ)*CC(IX,JY,KZ

      RETURN
      END

      SUBROUTINE COKE(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE COKE CONCENTRATION


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

        RC1 =
     +        ARC*EXP(-EC/(R*TM(IX,JY,KZ)))*POROS*SO(IX,JY,KZ)*
     +        DO(IX,JY,KZ)*XH(IX,JY,KZ)*S8*DT
        RD1 = ARD*EXP(-ED/(R*TM(IX,JY,KZ)))*PG(IX,JY,KZ)
     +      *YO(IX,JY,KZ)*DT
        CKP = CCP(IX,JY,KZ)
        CK1 = CKP
       DO 20 I = 1 , 100
          F = -CK1 + RC1*(1 - (CK1/CKMAX)**5) - RD1*CK1 + CKP
          DF = -1.0 - 5*RC1*(1./CKMAX**5)*CK1**4 - RD1
          CK2 = CK1 - F/DF
        IF(ABS(CK2-CK1) .LT. 1.0D-03) GOTO 40
          CK1 = CK2
20      CONTINUE
       WRITE(14,*)'NO CONVEGENNCE ON CK (COKE)','IX=',IX
40       CONTINUE
        IF( CK2 .LT. 0.000) THEN
         CK2 = 0.000
        ENDIF
        CC(IX,JY,KZ) = CK2
        IF(CC(IX,JY,KZ) .GT. CKMAX) CC(IX,JY,KZ)=CKMAX
      RETURN
      END

      SUBROUTINE PHYPRO(IX,JY,KZ)
```

```
C***** THIS ROUTINE CALCULATES THE PHYSICAL PROPERTIES


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
      CALL EQUILB(IX,JY,KZ)
      CALL SECOND(IX,JY,KZ)
      CALL DENSTY(IX,JY,KZ)
      CALL ENTHAL(IX,JY,KZ)
      CALL VISCOS(IX,JY,KZ)
      CALL PERM(IX,JY,KZ)

      RETURN
      END

      SUBROUTINE EQUILB(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

C... LIGHT OIL EQUILIBRIUM CONSTANT
         CLT(IX,JY,KZ) = 1./PG(IX,JY,KZ)*EXP(AKLT + BKLT/
     +                 (TM(IX,JY,KZ) - CKLT))

C... HEAVY OIL EQUILIBRIUM CONSTANT
       CHV(IX,JY,KZ)=1./PG(IX,JY,KZ)*EXP(AKHV + BKHV/(TM(IX,JY,KZ)
     +          )*(SO(IX,JY,KZ)/(SO(IX,JY,KZ)+1.0D-05))

C... WATER EQUILIBRIUM CONSTANT
       CWT(IX,JY,KZ)=1./PG(IX,JY,KZ)*((TM(IX,JY,KZ) - BKWT)/AKWT)*
       CWT(IX,JY,KZ)=CWT(IX,JY,KZ)*(SW(IX,JY,KZ)/(SW(IX,JY,KZ)+1.0

      RETURN
      END

      SUBROUTINE SECOND(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

C... WATER MOLE FRACTION
         YW(IX,JY,KZ) = CWT(IX,JY,KZ)
```

```
C... LIGHT OIL LIQUID MOLE FRACTION
       XL(IX,JY,KZ) = 1.00 - XH(IX,JY,KZ)
C       IF (XH(IX,JY,KZ) .GT. 1.000) THEN
C         XH(IX,JY,KZ) = 1.00000
C         XL(IX,JY,KZ) = 0.00000
C       ENDIF

C... LIGHT OIL VAPOUR MOLE FRACTION
       YL(IX,JY,KZ) = CLT(IX,JY,KZ)*XL(IX,JY,KZ)

C... HEAVY OIL VAPOUR MOLE FRACTION
       YH(IX,JY,KZ) = CHV(IX,JY,KZ)*XH(IX,JY,KZ)

C... OXYGEN MOLE FRACTION
       YN(IX,JY,KZ) = 1.00 - (YO(IX,JY,KZ) + YW(IX,JY,KZ) +
     +             YL(IX,JY,KZ) + YH(IX,JY,KZ))

C... GAS SATURATION
       SG(IX,JY,KZ) = 1.00 - (SO(IX,JY,KZ) + SW(IX,JY,KZ))
      IF(SG(IX,JY,KZ) .LT. 0.00)THEN
        SG(IX,JY,KZ) = 0.000
      ENDIF

C... OIL PRESSURE
       PO(IX,JY,KZ) = PG(IX,JY,KZ)

C... WATER PRESSURE

       PW(IX,JY,KZ) = PO(IX,JY,KZ)

      RETURN
      END

      SUBROUTINE DENSTY(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      DW(IX,JY,KZ) = CDWT*(1 + ADWT*(PW(IX,JY,KZ) - PREF) -
     +          BDWT*(TM(IX,JY,KZ) - TREF))
      DG(IX,JY,KZ) = PG(IX,JY,KZ)/(R*TM(IX,JY,KZ))
```

```
 DLT = (1.0/7287.0)*(1.0 - 0.32D-04*(PO(IX,JY,KZ) - PREF))*
+     (1.0 + 0.14D-02*(TM(IX,JY,KZ) - TREF))
 DHV = (1.0/4013.0)*(1.0 - 0.145D-05*(PO(IX,JY,KZ) - PREF))*
+     (1.0 + 0.70D-03*(TM(IX,JY,KZ) - TREF))
 VOL = (XL(IX,JY,KZ)*DLT + XH(IX,JY,KZ)*DHV)
 DO(IX,JY,KZ) = 1.0/VOL

 RETURN
 END

 SUBROUTINE ENTHAL(IX,JY,KZ)
 INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

 AGAS = YW(IX,JY,KZ)*AWT + YH(IX,JY,KZ)*AHV + YL(IX,JY,KZ)*ALT
+      YN(IX,JY,KZ)*ANR + YO(IX,JY,KZ)*AOX
 HG(IX,JY,KZ) = AGAS*(TM(IX,JY,KZ) - TREF)
 IF (TCWT .GT. TM(IX,JY,KZ)) THEN
  HVWT = AHWT*(TCWT - TM(IX,JY,KZ))**0.38
 ELSE
  HVWT = 0.00
 ENDIF
 HWG = AWT*(TM(IX,JY,KZ) - TREF)
 HW(IX,JY,KZ) = HWG - HVWT
 HLT = ALT*(TM(IX,JY,KZ) - TREF)
 HHV = AHV*(TM(IX,JY,KZ) - TREF)
 HOL = XL(IX,JY,KZ)*HLT + XH(IX,JY,KZ)*HHV
 TCO = XL(IX,JY,KZ)*TCLT + XH(IX,JY,KZ)*TCHV
 IF (TCO .GT. TM(IX,JY,KZ)) THEN
  HVOL = AHOL*(TCO - TM(IX,JY,KZ))**0.38
 ELSE
  HVOL = 0.00
 ENDIF
 HO(IX,JY,KZ) = HOL - HVOL
 HC(IX,JY,KZ) = ACK*(TM(IX,JY,KZ) - TREF)
 EG(IX,JY,KZ) = HG(IX,JY,KZ) - PG(IX,JY,KZ)/DG(IX,JY,KZ)
 EO(IX,JY,KZ) = HO(IX,JY,KZ) - PO(IX,JY,KZ)/DO(IX,JY,KZ)
 EW(IX,JY,KZ) = HW(IX,JY,KZ) - PW(IX,JY,KZ)/DW(IX,JY,KZ)
 ER(IX,JY,KZ) = ARK*(TM(IX,JY,KZ) - TREF)
 RETURN
```

```
      END

      SUBROUTINE VISCOS(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

      T = TM(IX,JY,KZ) - 273.
      VW(IX,JY,KZ) = AVWT/(BVWT + CVWT*T + DVWT*T**2)
      VLT = AVLT*EXP(BVLT/TM(IX,JY,KZ))
      VHV = AVHV*EXP(BVHV/TM(IX,JY,KZ))
      VO(IX,JY,KZ) = (VLT**XL(IX,JY,KZ))*(VHV**XH(IX,JY,KZ))
C      VG(IX,JY,KZ) = AVGS*(BVGS + CVGS*T)

      VG(IX,JY,KZ) = YO(IX,JY,KZ)*AVOXG*TM(IX,JY,KZ)**BVOXG +
     +           YN(IX,JY,KZ)*AVNRG*TM(IX,JY,KZ)**BVNRG
     +         + YL(IX,JY,KZ)*AVLTG*TM(IX,JY,KZ)**BVLTG +
     +           YH(IX,JY,KZ)*AVHVG*TM(IX,JY,KZ)**BVHVG
     +         + YW(IX,JY,KZ)*AVWTG*TM(IX,JY,KZ)**BVWTG

      RETURN
      END


      SUBROUTINE PERM1(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

C... GAS RELATIVE PERMEABILITY
         IF(SW(IX,JY,KZ) .GT. SWC) THEN
C     RPW(IX,JY,KZ) = CRWRO*((SW(IX,JY,KZ)-SWC)/(1.0-SORW - SWC))*
          RPW(IX,JY,KZ) = CRWRO*((SW(IX,JY,KZ) - SWC)/(1.0 - SWC))**
         ELSE
          RPW(IX,JY,KZ) = 0.0
         ENDIF
         IF(SG(IX,JY,KZ) .GT. SGC) THEN
C          RPG(IX,JY,KZ) = CRGRO*((SG(IX,JY,KZ) - SGC)/
C     +                      (1 - SWC - SORG - SGC))**ZG

          IF(SG(IX,JY,KZ) .GT. 1.0) SG(IX,JY,KZ) = 1.00
           DUM1 = (1.0 - SORG*(1 - SG(IX,JY,KZ)**12.0))**4.0
        DUM2 = 2.0*(1.0 - SORG*(1.0-SG(IX,JY,KZ)**12.0)) - SG(IX,JY
```

```
        RPG(IX,JY,KZ) = (SG(IX,JY,KZ)**3.0/DUM1)*DUM2
       ELSE
        RPG(IX,JY,KZ) = 0.0
       ENDIF
      DUM1 = 1.0 - SG(IX,JY,KZ) - SW(IX,JY,KZ)
      IF(DUM1 .LT. 0.00) THEN
        DUM1 = 0.000
       ELSE
        DUM1 = DUM1**3.0
       ENDIF
       DUM2 = (1.0 - SWC*(1.0 - (1.0 - SW(IX,JY,KZ))**5.0))**4.0
       DUM3 = 1.0+SW(IX,JY,KZ)-SG(IX,JY,KZ)-2.0*SWC*
    +         (1.0-(1.0-SW(IX,JY,KZ))**5.0)
       RPO(IX,JY,KZ) = (DUM1/DUM2)*DUM3
       RETURN
       END

     SUBROUTINE PERM(IX,JY,KZ)
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'

C... GAS RELATIVE PERMEABILITY
       IF(SW(IX,JY,KZ) .GT. SWC) THEN
      RPW(IX,JY,KZ) = 0.25*((SW(IX,JY,KZ)-SWC)/(1.0-SORW-SWC))**
       ELSE
        RPW(IX,JY,KZ) = 0.0
       ENDIF
       IF(SG(IX,JY,KZ) .GT. SGC) THEN
     RPG(IX,JY,KZ)= 0.7*((SG(IX,JY,KZ)-0.05)/(1-SWC-0.09-0.05))**
       ELSE
        RPG(IX,JY,KZ) = 0.0
       ENDIF
       IF (SO(IX,JY,KZ) .LT. SORW) THEN
       RPO(IX,JY,KZ) = 0.0
       ELSE
       DUMY1 = 1 - SWC - 0.09 - 0.05
       DUMY2 = 1 - SWC - SORW
        CROW = 1.0*(1.0 - ((SW(IX,JY,KZ) - SWC)/DUMY2))**3
        CROG = 1.0*(1.0 - ((SG(IX,JY,KZ) - 0.05)/DUMY1))**3
        IF (CROW .GT. 1.00)CROW = 1.00
```

```
      IF (CROW .LT. 0.00)CROW = 0.00
      IF (CROG .GT. 1.00)CROG = 1.00
      IF (CROG .LT. 0.00)CROG = 0.00
      RPO(IX,JY,KZ) = 1.0*((CROW/1.0 +RPW(IX,JY,KZ))*
+       (CROG/1.0 + RPG(IX,JY,KZ))
+                - RPW(IX,JY,KZ) - RPG(IX,JY,KZ))
     ENDIF
  RETURN
  END

  SUBROUTINE INPUT
  INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
  CHARACTER*50 FILEA,COMM*72

   FILEA = 'C:\PROJECT\INPUT\INPUT3D.DAT'
   OPEN (UNIT=13,FILE=FILEA,STATUS='OLD')

  READ(13,'(A72)')COMM
  READ(13,*)DTST,DTMAX,MAXSTP
  READ(13,'(A72)')COMM
  READ(13,*)URMX
  READ(13,'(A72)')COMM
  READ(13,*)PRCN
  READ(13,'(A72)')COMM
  READ(13,*)DNGP,DNOL,DNWT,DNOX,DNHL,DNTE
  READ(13,'(A72)')COMM
  READ(13,*)MAXITR,IFREQ
  READ(13,'(A72)')COMM
  READ(13,*)(TOLX(I),I=1,NV)
  READ(13,'(A72)')COMM
  READ(13,*)(PG(IX,1,1),IX=1,NX)
  READ(13,'(A72)')COMM
  READ(13,*)(TM(IX,1,1),IX=1,NX)
  READ(13,'(A72)')COMM
  READ(13,*)(YO(IX,1,1),IX=1,NX)
  READ(13,'(A72)')COMM
  READ(13,*)(XH(IX,1,1),IX=1,NX)
  READ(13,'(A72)')COMM
  READ(13,*)(SO(IX,1,1),IX=1,NX)
```

```
      READ(13,'(A72)')COMM
      READ(13,*)(SW(IX,1,1),IX=1,NX)
      READ(13,'(A72)')COMM
      READ(13,*)(CC(IX,1,1),IX=1,NX)
      READ(13,'(A72)')COMM
      READ(13,*)QW1,QH1,QL1,QN1,QO1
      READ(13,'(A72)')COMM
      READ(13,*)TII
      RETURN
      END

      BLOCK DATA BLKDAT
      PARAMETER (N=72,NX=4,NV=6,NY=3,NZ=1,NN=24,NNZ=72)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON / DELTA / DT,DX,DTMAX,DTST,OLDDT,DY,DZ
      COMMON / IFLAG / IFLAG1,IFLAG3,MAXSTP,IFLAG5
      COMMON / COUNT / ICOUNT,IFREQ,MELIMF
      COMMON / TIME  / TIM,TIME(10000)
      COMMON / PRES  / PG(NX,NY,NZ),PO(NX,NY,NZ),PW(NX,NY,NZ)
      COMMON / TEMP  / TM(NX,NY,NZ)
      COMMON / DENS  / DG(NX,NY,NZ),DO(NX,NY,NZ),DW(NX,NY,NZ)
      COMMON / VISC  / VG(NX,NY,NZ),VO(NX,NY,NZ),VW(NX,NY,NZ)
      COMMON / VAPOR / YW(NX,NY,NZ),YN(NX,NY,NZ),YO(NX,NY,NZ),
     +                 YL(NX,NY,NZ),YH(NX,NY,NZ)
      COMMON / LIQD  / XH(NX,NY,NZ),XL(NX,NY,NZ)
      COMMON / HEAT  / HG(NX,NY,NZ),HO(NX,NY,NZ),HW(NX,NY,NZ),
     +                 HC(NX,NY,NZ)
      COMMON / ENTER / EG(NX,NY,NZ),EO(NX,NY,NZ),EW(NX,NY,NZ),
     +                 ER(NX,NY,NZ)
      COMMON / PRM2  / RPG(NX,NY,NZ),RPW(NX,NY,NZ),RPO(NX,NY,NZ)
      COMMON / PREV1 / PGP(NX,NY,NZ),TMP(NX,NY,NZ),SGP(NX,NY,NZ),
     +                 SWP(NX,NY,NZ),SOP(NX,NY,NZ)
      COMMON / PREV2 / YWP(NX,NY,NZ),YNP(NX,NY,NZ),YOP(NX,NY,NZ),
     +                 YLP(NX,NY,NZ),YHP(NX,NY,NZ)
      COMMON / PREV3 / XHP(NX,NY,NZ),XLP(NX,NY,NZ),DGP(NX,NY,NZ),
     +                 DOP(NX,NY,NZ),DWP(NX,NY,NZ),CCP(NX,NY,NZ)
      COMMON / PREV4 / EGP(NX,NY,NZ),EOP(NX,NY,NZ),EWP(NX,NY,NZ),
     +                 ERP(NX,NY,NZ),HCP(NX,NY,NZ)
      COMMON / VARIAB/ X(N),A(N,N),RHS(N),BETA(N)
```

```
      COMMON / EQUT  / EALPG,EALSW,EALSO,EALTM,EALYO,EALXH
      COMMON / DARC1 / GDARC1,GDARC2,GTRAN,ODARC1,ODARC2,OTRAN
      COMMON / DARC2 / WDARC1,WDARC2,WTRAN,PRCN
      COMMON / DARC3 / GDAR1Y,GDAR2Y,GTRANY,ODAR1Y,ODAR2Y,OTRANY
      COMMON / DARC4 / WDAR1Y,WDAR2Y,WTRANY
      COMMON / DARC5 / GDAR1Z,GDAR2Z,GTRANZ,ODAR1Z,ODAR2Z,OTRANZ
      COMMON / DARC6 / WDAR1Z,WDAR2Z,WTRANZ
      COMMON / RERAT / RA(NX,NY,NZ),RB(NX,NY,NZ),RC(NX,NY,NZ),
     +                 RD(NX,NY,NZ)
      COMMON / OUTO  / OTCN(NX,NY,NZ)
      COMMON / OUTALL/ OTNR(NX,NY,NZ),OTWTV(NX,NY,NZ),OTWTL(NX,NY,
     +                 OTLTV(NX,NY,NZ),OTLTL(NX,NY,NZ),OTOX(NX,NY,NZ)
     +                 OTHVL(NX,NY,NZ),OTHVV(NX,NY,NZ)
      COMMON / COK   / CC(NX,NY,NZ)
      COMMON / EQULB / CLT(NX,NY,NZ),CHV(NX,NY,NZ),CWT(NX,NY,NZ)
      COMMON / SAT   / SG(NX,NY,NZ),SO(NX,NY,NZ),SW(NX,NY,NZ)
      COMMON / STEP  / ITER,MAXITR,URMX
      COMMON / TLOS  / UMAX(NV),TOLX(NV)
      COMMON / INLET / HIN(NX,NY,NZ),QW(NX,NY,NZ),QH(NX,NY,NZ),
     +     QL(NX,NY,NZ),QN(NX,NY,NZ),QO(NX,NY,NZ),TI(NX,NY,NZ),TII
      COMMON / INLET1/ QW1,QH1,QL1,QN1,QO1
      COMMON / DEF1  / PGD(NX,NY,NZ),SOD(NX,NY,NZ),SWD(NX,NY,NZ),
     +                 YOD(NX,NY,NZ),XHD(NX,NY,NZ),TMD(NX,NY,NZ)
      COMMON / DEF2  / DNGP,DNOL,DNWT,DNOX,DNHL,DNTE
      COMMON / DELT  / DEL(NV)
      COMMON / SOLV1 / TMAT(N,N),TRI(N,N),THS(N),RVT(N),DDEL(N),IP
      COMMON / SOLV2 / RRR(8100)
      COMMON / EQLB/ AKLT,BKLT,CKLT,AKHV,BKHV,CKHV,AKWT,BKWT,CKWT
      COMMON / ENTH/ AWT,AHV,ALT,ANR,AOX,ACK,ARK
      COMMON / ENHV/ AHWT,AHOL
      COMMON / DENS1/ ADWT,BDWT,CDWT,ADOL,BDOL,CDLT,CDHV
      COMMON / VISC1/ AVWT,BVWT,CVWT,DVWT,AVLT,BVLT,AVHV,BVHV,AVGS
     +              CVGS
      COMMON / VISC2/ AVOXG,AVNRG,AVLTG,AVHVG,AVWTG,BVOXG,BVNRG,BV
     +                BVHVG,BVWTG
      COMMON / REF / TREF,PREF
      COMMON / TC   / TCWT,TCLT,TCHV
      COMMON / CONS/ APR,APRY,APRZ,POROS,THCO,THCCAP,WLP
      COMMON / PRM / CRWRO,CRGRO,CROCW,SWC,SORW,SORG,SGC,ZW,ZG,ZOG
```

```
      COMMON / RESV/ RESL,RESW,RESD
      COMMON / RECT/ ARA,ARB,ARC,ARD,EA,EB,EC,ED
      COMMON / HREC/ HA,HB,HRC,HD
      COMMON / SREC/ S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12
      COMMON / RGAS/ R,CKMAX
      COMMON /AM/ AMOX,AMNR,AMLT,AMHV,AMWT,AMCK
      COMMON /OUT2/ AOTNR,AOTOX,AOTLTL,AOTLTV,AOTHVL,AOTHVV,
     +         AOTWTL,AOTWTV
      COMMON /OUT3/ VOTNR,VOTOX,VOTLTL,VOTLTV,VOTHVL,VOTHVV,
     +         VOTWTL,VOTWTV
      COMMON /OUT4/ ROTLTL,ROTLTV,ROTHVL,ROTHVV,
     +         ROTWTL,ROTWTV
      COMMON /VOM / VOLUMB
      COMMON /PPPC / PCWT,PCLT,PCHV
      COMMON /OOIL / VOOIP
      COMMON /OWT /  VOWIP
      COMMON /STDN/ STDWT,STDLT,STDHV
      COMMON /HEAT1/HREAC(NX,NY,NZ),HLOSS(NX,NY,NZ),PHL(NX,NY,NZ),
     +         QHL(NX,NY,NZ),PPHL(NX,NY,NZ),
     +         PQHL(NX,NY,NZ),CAPEN(NX,NY,NZ),PCAPEN(NX,NY,NZ)
      COMMON /DIFFS/ THDIF,DIFL,PDIFL
      COMMON /HEAT2/ TOTHET,TOTLOS
      COMMON /VPPPP/ VGP(NX,NY,NZ),VOP(NX,NY,NZ),VWP(NX,NY,NZ)
      COMMON /LUBBB/ B(N),VV(N)
      COMMON /LUBB2/ INDX(N)
      COMMON /GRAVTY/ GGRAV1,WGRAV1,OGRAV1,GGRAV2,WGRAV2,OGRAV2
      COMMON /MOLWT1/ WTMOLG(NX,NY,NZ),WTMOLO(NX,NY,NZ)
      COMMON /OMEG/ AAX(NX,NY,NZ),AAY(NX,NY,NZ),OMGX(NX,NY,NZ),
     +         OMGY(NX,NY,NZ),AAZ(NX,NY,NZ),OMGZ(NX,NY,NZ),
     +         OMOX(NX,NY,NZ),OMOY(NX,NY,NZ),OMWX(NX,NY,NZ),
     +         OMOZ(NX,NY,NZ),OMWY(NX,NY,NZ),OMWZ(NX,NY,NZ),
     +         RPGP(NX,NY,NZ),RPOP(NX,NY,NZ),RPWP(NX,NY,NZ)
C      EXTERNAL BLKDAT

      DATA AMOX,AMNR,AMWT,AMLT,AMHV,AMCK /32.0,44.0,18.0,44.0,170.
     +                        13.0/
      DATA AKLT,BKLT,CKLT,AKHV,BKHV,CKHV,AKWT,BKWT,CKWT /13.71,
     +   -1872.0,25.16,14.06,-3744.0,92.85,41.818,255.37,4.464/
      DATA AWT,AHV,ALT,ANR,AOX,ACK,ARK /33.5D-03,242.0D-03,482.0D-
```

```
     +    46.0D-03,32.2D-03,17.0D-03,2.35D+03/
      DATA AHWT,AHOL /4.814,5.580/
      DATA TCWT,TCLT,TCHV /647.1,369.444,657.94/
      DATA TREF,PREF /298.0,450.0/
      DATA ADWT,BDWT,CDWT,ADOL,BDOL,CDLT,CDHV /4.5D-06,2.1D-04,5.5
     +    8.0D-06,2.0D-04,8.37D+03,4.16D+03/
      DATA AVWT,BVWT,CVWT,DVWT,AVLT,BVLT,AVHV,BVHV,AVGS,BVGS,CVGS
     +    1.0D-09,12.1,2.88,7.78D-04,2.41D-13,533.1,4.19D-15,4714
     +    1.0D-13,1.574,44.0D-02/
      DATA CRWRO,CRGRO,CROCW,SWC,SORW,SORG,SGC,ZW,ZG,ZOG,ZOW /0.25
     +    1.0,0.2,0.3,0.09,0.05,3.0,1.0,3.0,3.0/
      DATA APR,APRY,APRZ,POROS,THCO,THCCAP,WLP /4.0D-12,4.0D-12,4.
     +                  0.38,2.4D+02,2.4D+02,410.0/
      DATA RESL,RESW,RESD /50.00,35.00,19.2/
      DATA ARA,ARB,ARC,ARD,EA,EB,EC,ED /1.45D+05,1.45D+05,3.00D+05
     +    1.45D+05,77.46,77.46,66.99,54.43/
      DATA HA,HB,HRC,HD /2.21D+03,8.12D+03,4.65D+01,5.23D+02/
      DATA S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12 /5.0,3.0,4.0,18.
     +    12.0,13.0,2.0,4.67,1.33,1.25,1.00,0.5/
      DATA R,CKMAX /8.314D-03,1.2349D+03/
      DATA AVOXG,AVNRG,AVLTG,AVHVG,AVWTG,BVOXG,BVNRG,BVLTG,BVHVG,B
     +    3.8830D-15,3.7169D-15,4.3638D-16,8.6845D-17,1.9676D-16,
     +    0.721,0.702,0.943,1.102,1.116/
      DATA PCWT,PCLT,PCHV /22110.0,4248.0,1820.0/
      DATA STDWT,STDLT,STDHV/55500.0,8612.727,4274.137/
      END
```

```
      SUBROUTINE PRIME
      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
      CHARACTER*50 FILEB
      FILEB = 'C:\PROJECT\OUTPUT\PRIME3D.DAT'
      IF (ICOUNT .EQ. 0) THEN
       OPEN (UNIT=15,FILE=FILEB,STATUS='UNKNOWN')
       WRITE(15,10)
10      FORMAT('1',10X,' IN-SITU COMBUSTION SIMULATION PROGRAM')
       WRITE(15,20)
20      FORMAT(1X,10X,'       Written by Jaafar Oklany      ')
       WRITE(15,30)PRCN,URMX
30      FORMAT(1X,'Productivity Constant = ',F9.4,'  URMX = ',F9.4)
       WRITE(15,33)
33      FORMAT(1X,'CONVERGANCE TOLARENCES (PG,SW,SO,TM,YO,XH)')
       WRITE(15,35)(TOLX(I),I=1,NV)
35      FORMAT(1X,6(3X,F10.7))
       WRITE(15,40)
40      FORMAT(1X,'Injection Rate (water,ho,lo,inerts,oxy)')
        DO 2342 JY = NY , 1 , -1
       WRITE(15,50)QW(1,JY,1),QH(1,JY,1),
     +                QL(1,JY,1),QN(1,JY,1),QO(1,JY,1)
2342    CONTINUE

50      FORMAT(1X,5F9.4)
       WRITE(15,60)
60      FORMAT(1X,'Injection Temperature (K)')
       WRITE(15,70)TII
70      FORMAT(1X,5F9.4,/,1X,5F9.4)
C      WRITE(15,75)
C75      FORMAT(1X,'ORIGINAL OIL IN PLACE OOIP (MOL)')
C       WRITE(15,76)OOIP
C76      FORMAT(1X,D12.5)
       WRITE(15,80)
80      FORMAT(1X,'Initial Pressure (KPa)')
       DO 6000 KZ = 1 , NZ
        WRITE(15,1111)KZ
       DO 600 JY = NY , 1 , -1
        WRITE(15,1112)JY
       WRITE(15,90)(PG(IX,JY,KZ),IX=1,NX)
600     CONTINUE
6000    CONTINUE
90      FORMAT(1X,5(3X,F12.6),/,1X,5(3X,F12.6))
       WRITE(15,100)
100     FORMAT(1X,'Initial Temperature (K)')
       DO 6001 KZ = 1 , NZ
        WRITE(15,1111)KZ
       DO 601 JY = NY , 1 , -1
        WRITE(15,1112)JY
       WRITE(15,90)(TM(IX,JY,KZ),IX=1,NX)
601     CONTINUE
6001   CONTINUE
       WRITE(15,120)
120     FORMAT(1X,'Initial Oil Saturation')
       DO 6002 KZ = 1 , NZ
        WRITE(15,1111)KZ
       DO 602 JY = NY , 1 , -1
        WRITE(15,1112)JY
```

```
      WRITE(15,130)(SO(IX,JY,KZ),IX=1,NX)
602    CONTINUE
6002  CONTINUE
130    FORMAT(1X,5(3X,F12.8),/,1X,5(3X,F12.8))
      WRITE(15,140)
140    FORMAT(1X,'Initial Water Saturation')
      DO 6003 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 603 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,130)(SW(IX,JY,KZ),IX=1,NX)
603    CONTINUE
6003  CONTINUE
      WRITE(15,160)
160    FORMAT(1X,'Initial Oxygen Mole Fraction')
      DO 6004 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 604 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,130)(YO(IX,JY,KZ),IX=1,NX)
604    CONTINUE
6004  CONTINUE
      WRITE(15,180)
180    FORMAT(1X,'Initial Heavy Oil Mole Fraction In The Oil Phase'
      DO 6005 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 605 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,130)(XH(IX,JY,KZ),IX=1,NX)
605    CONTINUE
6005   CONTINUE
      WRITE(15,200)
200    FORMAT(1X,'Initial Coke Concentration')
      DO 6006 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 606 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,90)(CC(IX,JY,KZ),IX=1,NX)
606    CONTINUE
6006   CONTINUE
      WRITE(15,334)
334    FORMAT(1X,'OWIP  (M 3)')
      WRITE(15,343)VOWIP
343    FORMAT(1X,24X,D12.5)
      WRITE(15,353)
353    FORMAT(1X,'OOIP  (M 3).')
      WRITE(15,343)VOOIP
      GOTO 999
     ENDIF

      WRITE(15,220)ICOUNT,OLDDT,TIME(ICOUNT),ITER
220    FORMAT('1','TIME STEP NO.',I5,4X,'DT=',F9.4,4X,'TIME=',
    +  F9.4,' ITER= ',I3,/)

      WRITE(15,240)
240    FORMAT(1X,'PRESSURE (KPa)')
      DO 6007 KZ = 1 , NZ
       WRITE(15,1111)KZ
```

```
      DO 607 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,90)(PG(IX,JY,KZ),IX=1,NX)
607    CONTINUE
6007  CONTINUE
      WRITE(15,250)
250    FORMAT(1X,'TEMPERATURE (K)')
      DO 6008 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 608 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,90)(TM(IX,JY,KZ),IX=1,NX)
608    CONTINUE
6008  CONTINUE
      WRITE(15,260)
260    FORMAT(1X,'OIL SATURATION')
      DO 6009 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 609 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,130)(SO(IX,JY,KZ),IX=1,NX)
609    CONTINUE
6009  CONTINUE
      WRITE(15,270)
270    FORMAT(1X,'WATER SATURATION')
      DO 6011 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 611 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,130)(SW(IX,JY,KZ),IX=1,NX)
611    CONTINUE
6011  CONTINUE
      WRITE(15,280)
280    FORMAT(1X,'OXYGEN MOLE FRACTION')
      DO 6012 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 612 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,130)(YO(IX,JY,KZ),IX=1,NX)
612    CONTINUE
6012   CONTINUE
      WRITE(15,290)
290    FORMAT(1X,'HEAVY OIL MOLE FRACTION IN THE OIL PHASE')
      DO 6013 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 613 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,130)(XH(IX,JY,KZ),IX=1,NX)
613    CONTINUE
6013  CONTINUE
      WRITE(15,295)
295    FORMAT(1X,'LIGHT OIL MOLE FRACTION IN THE VAP PHASE')
      DO 6014 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 614 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,130)(YL(IX,JY,KZ),IX=1,NX)
614    CONTINUE
```

```
6014  CONTINUE
      WRITE(15,300)
300    FORMAT(1X,'COKE CONCENTRATION (MOL/M**3)')
      DO 6015 KZ = 1 , NZ
       WRITE(15,1111)KZ
      DO 615 JY = NY , 1 , -1
       WRITE(15,1112)JY
      WRITE(15,90)(CC(IX,JY,KZ),IX=1,NX)
615    CONTINUE
6015  CONTINUE

1111  FORMAT(1X,'Z = ',I2)
1112  FORMAT(1X,3X,'Y = ',I2)
      WRITE(15,395)
395    FORMAT('1',12X,
     +       '****************************************************
      WRITE(15,400)
400    FORMAT(1X,19X,'ACCUMULATED PRODUCTION OF PHASES (MOLE)')
      WRITE(15,396)
396    FORMAT(1X,12X,
     +       '****************************************************
      WRITE(15,405)AOTNR+AOTOX+AOTWTV+AOTLTV+AOTHVV
405    FORMAT(1X,27X,'GAS PHASE   = ',D12.5)
      WRITE(15,410)AOTWTL
410    FORMAT(1X,27X,'WATER PHASE = ',D12.5)
      WRITE(15,415)AOTLTL+AOTHVL
415    FORMAT(1X,27X,'OIL PHASE   = ',D12.5,////)

C      WRITE(15,417)
C417    FORMAT(1X,12X,
C    +       '***************************************************')
C      WRITE(15,420)
C420    FORMAT(1X,19X,'COMPONENTS PRODUCTION RATES(MOLE/DAY)')
C      WRITE(15,418)
C418    FORMAT(1X,12X,
C    +          **************************************************
C      WRITE(15,430)
430    FORMAT(1X,12X,'COMPONENT',6X,'VAPOR',10X,'LIQUID',9X,'TOTAL
C      WRITE(15,435)
435    FORMAT(1X,12X,
     +       '----------------------------------------------------
C      WRITE(15,440)OTWTV,OTWTV,OTWTL+OTWTV
440    FORMAT(1X,12X,'WATER',6X,D12.5,3X,D12.5,3X,D12.5)
C      WRITE(15,450)OTHVV,OTHVL,OTHVV+OTHVL
450    FORMAT(1X,12X,'HV OIL',5X,D12.5,3X,D12.5,3X,D12.5)
C      WRITE(15,460)OTLTV,OTLTL,OTLTV+OTLTL
460    FORMAT(1X,12X,'LT OIL',5X,D12.5,3X,D12.5,3X,D12.5)
C      WRITE(15,470)OTNR,OTNR
C470    FORMAT(1X,12X,'INERT GAS',2X,D12.5,4X,11('-'),3X,D12.5)
C      WRITE(15,480)OTOX,OTOX
C480    FORMAT(1X,12X,'OXYGEN',5X,D12.5,4X,11('-'),3X,D12.5,////)
C
      WRITE(15,485)
485    FORMAT(1X,12X,
     +      '****************************************************'
      WRITE(15,490)
490    FORMAT(1X,19X,'ACUMULATED COMPONENTS PRODUCTION (M 3)')
      WRITE(15,487)
```

```
487  FORMAT(1X,12X,
    +     '*****************************************************'

     WRITE(15,430)
     WRITE(15,435)
     WRITE(15,440)VOTWTV,VOTWTV,VOTWTL+VOTWTV
     WRITE(15,450)VOTHVV,VOTHVL,VOTHVV+VOTHVL
     WRITE(15,460)VOTLTV,VOTLTL,VOTLTV+VOTLTL
C     WRITE(15,470)VOTNR,VOTNR
C     WRITE(15,480)VOTOX,VOTOX
     DO 333 III = 1 , 4
      WRITE(15,*)
333   CONTINUE

     WRITE(15,488)
488   FORMAT(1X,10X,
    + '********************************************************')
     WRITE(15,500)
500   FORMAT(1X,12X,
    + 'OIL AND WATER PRODUCTION RELATIVE TO THE ORIGINAL (%)')
     WRITE(15,489)
489   FORMAT(1X,10X,
    + '********************************************************',/

     WRITE(15,530)
530   FORMAT(1X,12X,'COMPONENT',6X,'VAPOR',10X,'LIQUID',9X,'TOTAL
     WRITE(15,535)
535   FORMAT(1X,12X,
    +      '-----------------------------------------------------
     WRITE(15,540)ROTWTV,ROTWTL,ROTWTL+ROTWTV
540   FORMAT(1X,12X,'WATER',6X,D12.5,3X,D12.5,3X,D12.5)
     WRITE(15,550)ROTHVV,ROTHVL,ROTHVV+ROTHVL
550   FORMAT(1X,12X,'HV OIL',5X,D12.5,3X,D12.5,3X,D12.5)
     WRITE(15,560)ROTLTV,ROTLTL,ROTLTV+ROTLTL
560   FORMAT(1X,12X,'LT OIL',5X,D12.5,3X,D12.5,3X,D12.5)
     ROTOL = ROTLTV+ROTLTL+ROTHVV+ROTHVL
     WRITE(15,570)ROTLTV+ROTHVV,ROTLTL+ROTHVL,ROTOL
570   FORMAT(1X,12X,'TOT OIL',4X,D12.5,3X,D12.5,3X,D12.5)
     WRITE(15,610)
610   FORMAT(1X,10X,
    + '********************************************************',
     WRITE(15,635) TOTHET
635   FORMAT(1X,12X,'TOTAL HEAT OF REACTION (kJ):',D12.5)
     WRITE(15,620) TOTLOS
620   FORMAT(1X,12X,'TOTAL HEAT LOSS (kJ):',D12.5)
     WRITE(15,625) TOTLOS/TOTHET *100.0
625   FORMAT(1X,12X,'HEAT LOSS/HEAT OF REACTION (%):',F6.2)

999   CONTINUE
     RETURN
     END

     SUBROUTINE HTLOSS(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE HEAT LOSS RATE


     INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
```

```
       DIFL = SQRT(THDIF*TIME(ICOUNT))/2.0
        IF (JY .EQ. 1) THEN

       DUMY = (TMP(IX,JY,KZ) - TI(IX,JY,KZ))*PDIFL + PPHL(IX,JY,KZ)
     +        *(PDIFL)**2.0 + 2.0*PQHL(IX,JY,KZ)*(PDIFL)**3.0
       DUMY1 = THDIF*DT*(TM(IX,JY,KZ) - TI(IX,JY,KZ))/DIFL
       DUMY2 = (DIFL)**3.0*(TM(IX,JY,KZ) - TMP(IX,JY,KZ))/(THDIF*DT
       DUMY3 = 3.0*(DIFL)**2.0 + THDIF*DT
       PHL(IX,JY,KZ) = (DUMY1 + DUMY - DUMY2)/DUMY3

       DUMY4 = 2.0*PHL(IX,JY,KZ)*DIFL - (TM(IX,JY,KZ) - TI(IX,JY,KZ
       DUMY5 = (DIFL)**2.0 *(TM(IX,JY,KZ) - TMP(IX,JY,KZ))/(THDIF*D
       QHL(IX,JY,KZ) = (DUMY4 + DUMY5)/(2.0*(DIFL**2.0))
       CAPEN(IX,JY,KZ) = (THCCAP/THDIF)*DIFL*((TM(IX,JY,KZ) -
     + TI(IX,JY,KZ))+PHL(IX,JY,KZ)*DIFL+2.0*QHL(IX,JY,KZ)*(DIFL)**
       HLOSS(IX,JY,KZ) = CAPEN(IX,JY,KZ) - PCAPEN(IX,JY,KZ)
       HLOSS(IX,JY,KZ) = HLOSS(IX,JY,KZ)/DY
       ELSE
       HLOSS(IX,JY,KZ) = 0.0
       ENDIF
       RETURN
       END

       SUBROUTINE MOLWT(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE MOLCULAR WIEGHTS


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
       WTMOLO(IX,JY,KZ) = AMLT*XL(IX,JY,KZ) + AMHV*XH(IX,JY,KZ)
       WTMOLG(IX,JY,KZ) = AMOX*YO(IX,JY,KZ) + AMNR*YN(IX,JY,KZ) +
     + AMWT*YW(IX,JY,KZ) + AMLT*YL(IX,JY,KZ) + AMHV*YH(IX,JY,KZ)
       RETURN
       END


      SUBROUTINE DARCY(IX,JY,KZ)

C***** THIS ROUTINE CALCULATES THE MOBILITY TERM USING VAUGHN
C     METHOD


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
       IPX = IX - 1
       JPY = JY - 1
       PRG = APR*RPG(IX,JY,KZ)
       PRO = APR*RPO(IX,JY,KZ)
       PRW = APR*RPW(IX,JY,KZ)
      IF(IX .NE. NX) THEN
        INX = IX + 1
        CALL PHYPRO(INX,JY,KZ)
      ENDIF

      IF (IX .EQ. 1) THEN
       GTRAN = PRG*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
     +        (DG(IX+1,JY,KZ)/VG(IX+1,JY,KZ)))/2.0
       OTRAN = PRO*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
     +        (DO(IX+1,JY,KZ)/VO(IX+1,JY,KZ)))/2.0
```

```
       WTRAN = PRW*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
     +       (DW(IX+1,JY,KZ)/VW(IX+1,JY,KZ)))/2.0
       GDARC1 = GTRAN*(PG(IX+1,JY,KZ) - PG(IX,JY,KZ))/DX**2
       GDARC2 = 0.000
       ODARC1 = OTRAN*(PO(IX+1,JY,KZ) - PO(IX,JY,KZ))/DX**2
       ODARC2 = 0.000
       WDARC1 = WTRAN*(PW(IX+1,JY,KZ) - PW(IX,JY,KZ))/DX**2
       WDARC2 = 0.000
      ELSEIF (IX .GT. 1 .AND. IX .NE. NX) THEN
       GTRAN = PRG*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
     +       (DG(IX+1,JY,KZ)/VG(IX+1,JY,KZ)))/2.0
       OTRAN = PRO*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
     +       (DO(IX+1,JY,KZ)/VO(IX+1,JY,KZ)))/2.0
       WTRAN = PRW*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
     +       (DW(IX+1,JY,KZ)/VW(IX+1,JY,KZ)))/2.0
       PRGP = APR*RPG(IX-1,JY,KZ)
       PROP = APR*RPO(IX-1,JY,KZ)
       PRWP = APR*RPW(IX-1,JY,KZ)
       GTRANP=PRGP*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
     +      (DG(IX-1,JY,KZ)/VG(IX-1,JY,KZ)))/2.0
       OTRANP=PROP*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
     +      (DO(IX-1,JY,KZ)/VO(IX-1,JY,KZ)))/2.0
       WTRANP=PRWP*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
     +      (DW(IX-1,JY,KZ)/VW(IX-1,JY,KZ)))/2.0

       GDARC1 = GTRAN*(PG(IX+1,JY,KZ) - PG(IX,JY,KZ))/DX**2
       GDARC2 = GTRANP*(PG(IX-1,JY,KZ) - PG(IX,JY,KZ))/DX**2
       ODARC1 = OTRAN*(PO(IX+1,JY,KZ) - PO(IX,JY,KZ))/DX**2
       ODARC2 = OTRANP*(PO(IX-1,JY,KZ) - PO(IX,JY,KZ))/DX**2
       WDARC1 = WTRAN*(PW(IX+1,JY,KZ) - PW(IX,JY,KZ))/DX**2
       WDARC2 = WTRANP*(PW(IX-1,JY,KZ) - PW(IX,JY,KZ))/DX**2
      ELSEIF (IX .EQ. NX) THEN
       PRGP = APR*RPG(IX-1,JY,KZ)
       PROP = APR*RPO(IX-1,JY,KZ)
       PRWP = APR*RPW(IX-1,JY,KZ)
       GTRAN = PRG*DG(IX,JY,KZ)/VG(IX,JY,KZ)
       OTRAN = PRO*DO(IX,JY,KZ)/VO(IX,JY,KZ)
       WTRAN = PRW*DW(IX,JY,KZ)/VW(IX,JY,KZ)
       GTRANP=PRGP*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
     +      (DG(IX-1,JY,KZ)/VG(IX-1,JY,KZ)))/2.0
       OTRANP=PROP*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
     +      (DO(IX-1,JY,KZ)/VO(IX-1,JY,KZ)))/2.0
       WTRANP=PRWP*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
     +      (DW(IX-1,JY,KZ)/VW(IX-1,JY,KZ)))/2.0
       GDARC1 = 0.000
       GDARC2 = GTRANP*(PG(IX-1,JY,KZ) - PG(IX,JY,KZ))/DX**2
       ODARC1 = 0.000
       ODARC2 = OTRANP*(PO(IX-1,JY,KZ) - PO(IX,JY,KZ))/DX**2
       WDARC1 = 0.000
       WDARC2 = WTRANP*(PW(IX-1,JY,KZ) - PW(IX,JY,KZ))/DX**2
      ENDIF
      IF (NY .NE. 1) THEN
       IF (JY .NE. NY) THEN
         JNY = JY + 1
         CALL PHYPRO(IX,JNY,KZ)
         CALL MOLWT(IX,JNY,KZ)
        ENDIF
       PRG = APRY*RPG(IX,JY,KZ)
```

```
       PRO = APRY*RPO(IX,JY,KZ)
       PRW = APRY*RPW(IX,JY,KZ)
      IF (JY .EQ. 1) THEN
       GTRANY=PRG*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
     +          (DG(IX,JY+1,KZ)/VG(IX,JY+1,KZ)))/2.0
       OTRANY=PRO*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
     +          (DO(IX,JY+1,KZ)/VO(IX,JY+1,KZ)))/2.0
       WTRANY=PRW*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
     +          (DW(IX,JY+1,KZ)/VW(IX,JY+1,KZ)))/2.0
       GDAR1Y = GTRANY*(PG(IX,JY+1,KZ) - PG(IX,JY,KZ))/DY**2
       GDAR2Y = 0.000
       ODAR1Y = OTRANY*(PO(IX,JY+1,KZ) - PO(IX,JY,KZ))/DY**2
       ODAR2Y = 0.000
       WDAR1Y = WTRANY*(PW(IX,JY+1,KZ) - PW(IX,JY,KZ))/DY**2
       WDAR2Y = 0.000
       GGRAV1 = 0.00
       WGRAV1 = 0.00
       OGRAV1 = 0.00
       GGRAV2 = 0.00
       WGRAV2 = 0.00
       OGRAV2 = 0.00
      ELSEIF (JY .GT. 1 .AND. JY .NE. NY) THEN
       GTRANY=PRG*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
     +          (DG(IX,JY+1,KZ)/VG(IX,JY+1,KZ)))/2.0
       OTRANY=PRO*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
     +          (DO(IX,JY+1,KZ)/VO(IX,JY+1,KZ)))/2.0
       WTRANY=PRW*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
     +          (DW(IX,JY+1,KZ)/VW(IX,JY+1,KZ)))/2.0
       PRGP = APRY*RPG(IX,JY-1,KZ)
       PROP = APRY*RPO(IX,JY-1,KZ)
       PRWP = APRY*RPW(IX,JY-1,KZ)
       GTRAPY=PRGP*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
     +          (DG(IX,JY-1,KZ)/VG(IX,JY-1,KZ)))/2.0
       OTRAPY=PROP*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
     +          (DO(IX,JY-1,KZ)/VO(IX,JY-1,KZ)))/2.0
       WTRAPY=PRWP*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
     +          (DW(IX,JY-1,KZ)/VW(IX,JY-1,KZ)))/2.0

       GDAR1Y = GTRANY*(PG(IX,JY+1,KZ) - PG(IX,JY,KZ))/DY**2
       GDAR2Y = GTRAPY*(PG(IX,JY-1,KZ) - PG(IX,JY,KZ))/DY**2
       ODAR1Y = OTRANY*(PO(IX,JY+1,KZ) - PO(IX,JY,KZ))/DY**2
       ODAR2Y = OTRAPY*(PO(IX,JY-1,KZ) - PO(IX,JY,KZ))/DY**2
       WDAR1Y = WTRANY*(PW(IX,JY+1,KZ) - PW(IX,JY,KZ))/DY**2
       WDAR2Y = WTRAPY*(PW(IX,JY-1,KZ) - PW(IX,JY,KZ))/DY**2
       GGRAV1 = -(APRY*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)*
     +             WTMOLG(IX,JY,KZ)*DG(IX,JY,KZ))*(9.8D-06/(DY))
       GGRAV2 = -(APRY*RPG(IX,JY-1,KZ)*DG(IX,JY-1,KZ)/VG(IX,JY-1,K
     +              DG(IX,JY-1,KZ)*WTMOLG(IX,JY-1,KZ))*(9.8D-06/(DY))
       WGRAV1 = -(APRY*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)*
     +              AMWT*DW(IX,JY,KZ))*(9.8D-06/(DY))
       WGRAV2 = -(APRY*RPW(IX,JY-1,KZ)*DW(IX,JY-1,KZ)/VW(IX,JY-1,K
     +               DW(IX,JY-1,KZ)*AMWT)*(9.8D-06/(DY))
       OGRAV1 = -(APRY*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)*
     +             WTMOLO(IX,JY,KZ)*DO(IX,JY,KZ))*(9.8D-06/(DY))
       OGRAV2 = -(APRY*RPO(IX,JY-1,KZ)*DO(IX,JY-1,KZ)/VO(IX,JY-1,K
     +              DO(IX,JY-1,KZ)*WTMOLO(IX,JY-1,KZ))*(9.8D-06/(DY))
      ELSEIF (JY .EQ. NY) THEN
       PRGP = APRY*RPG(IX,JY-1,KZ)
```

```
      PROP = APRY*RPO(IX,JY-1,KZ)
      PRWP = APRY*RPW(IX,JY-1,KZ)
      GTRANY = PRG*DG(IX,JY,KZ)/VG(IX,JY,KZ)
      OTRANY = PRO*DO(IX,JY,KZ)/VO(IX,JY,KZ)
      WTRANY = PRW*DW(IX,JY,KZ)/VW(IX,JY,KZ)
      GTRAPY=PRGP*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
    +      (DG(IX,JY-1,KZ)/VG(IX,JY-1,KZ)))/2.0
      OTRAPY=PROP*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
    +      (DO(IX,JY-1,KZ)/VO(IX,JY-1,KZ)))/2.0
      WTRAPY=PRWP*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
    +      (DW(IX,JY-1,KZ)/VW(IX,JY-1,KZ)))/2.0
      GDAR1Y = 0.000
      GDAR2Y = GTRAPY*(PG(IX,JY-1,KZ) - PG(IX,JY,KZ))/DY**2
      ODAR1Y = 0.000
      ODAR2Y = OTRAPY*(PO(IX,JY-1,KZ) - PO(IX,JY,KZ))/DY**2
      WDAR1Y = 0.000
      WDAR2Y = WTRAPY*(PW(IX,JY-1,KZ) - PW(IX,JY,KZ))/DY**2
      GGRAV1 = -(APRY*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)*
    +        WTMOLG(IX,JY,KZ)*DG(IX,JY,KZ))*(9.8D-06/(DY))
      GGRAV2 = -(APRY*RPG(IX,JY-1,KZ)*DG(IX,JY-1,KZ)/VG(IX,JY-1,K
    +         DG(IX,JY-1,KZ)*WTMOLG(IX,JY-1,KZ))*(9.8D-06/(DY))
      WGRAV1 = -(APRY*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)*
    +         AMWT*DW(IX,JY,KZ))*(9.8D-06/(DY))
      WGRAV2 = -(APRY*RPW(IX,JY-1,KZ)*DW(IX,JY-1,KZ)/VW(IX,JY-1,K
    +          DW(IX,JY-1,KZ)*AMWT)*(9.8D-06/(DY))
      OGRAV1 = -(APRY*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)*
    +        WTMOLO(IX,JY,KZ)*DO(IX,JY,KZ))*(9.8D-06/(DY))
      OGRAV2 = -(APRY*RPO(IX,JY-1,KZ)*DO(IX,JY-1,KZ)/VO(IX,JY-1,K
    +         DO(IX,JY-1,KZ)*WTMOLO(IX,JY-1,KZ))*(9.8D-06/(DY))
     ENDIF
     ELSE
     GDAR1Y = 0.00
     GDAR2Y = 0.00
     WDAR1Y = 0.00
     WDAR2Y = 0.00
     ODAR1Y = 0.00
     ODAR2Y = 0.00
      GGRAV1 = 0.00
      OGRAV1 = 0.00
      WGRAV1 = 0.00
      GGRAV2 = 0.00
      OGRAV2 = 0.00
      WGRAV2 = 0.00
     ENDIF
C       GGRAV1  = 0.0
C       WGRAV1 =  0.0
C       OGRAV1 = 0.0
C       GGRAV2  = 0.0
C       WGRAV2 =  0.0
C       OGRAV2 = 0.0
     IF (NZ .NE. 1) THEN
      IF (KZ .NE. NZ) THEN
       KNZ = KZ + 1
       CALL PHYPRO(IX,JY,KNZ)
C      CALL MOLWT(IX,JY,KNZ)
       ENDIF
      PRG = APRZ*RPG(IX,JY,KZ)
      PRO = APRZ*RPO(IX,JY,KZ)
```

```
  PRW = APRZ*RPW(IX,JY,KZ)
 IF (KZ .EQ. 1) THEN
  GTRANZ=PRG*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
+        (DG(IX,JY,KZ+1)/VG(IX,JY,KZ+1)))/2.0
  OTRANZ=PRO*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
+        (DO(IX,JY,KZ+1)/VO(IX,JY,KZ+1)))/2.0
  WTRANZ=PRW*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
+        (DW(IX,JY,KZ+1)/VW(IX,JY,KZ+1)))/2.0
  GDAR1Z = GTRANZ*(PG(IX,JY,KZ+1) - PG(IX,JY,KZ))/DZ**2
  GDAR2Z = 0.000
  ODAR1Z = OTRANZ*(PO(IX,JY,KZ+1) - PO(IX,JY,KZ))/DZ**2
  ODAR2Z = 0.000
  WDAR1Z = WTRANZ*(PW(IX,JY,KZ+1) - PW(IX,JY,KZ))/DZ**2
  WDAR2Z = 0.000
 ELSEIF (KZ .GT. 1 .AND. KZ .NE. NZ) THEN
  GTRANZ=PRG*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
+        (DG(IX,JY,KZ+1)/VG(IX,JY,KZ+1)))/2.0
  OTRANZ=PRO*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
+        (DO(IX,JY,KZ+1)/VO(IX,JY,KZ+1)))/2.0
  WTRANZ=PRW*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
+        (DW(IX,JY,KZ+1)/VW(IX,JY,KZ+1)))/2.0
  PRGP = APRZ*RPG(IX,JY,KZ-1)
  PROP = APRZ*RPO(IX,JY,KZ-1)
  PRWP = APRZ*RPW(IX,JY,KZ-1)
  GTRAPZ=PRGP*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
+        (DG(IX,JY,KZ-1)/VG(IX,JY,KZ-1)))/2.0
  OTRAPZ=PROP*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
+        (DO(IX,JY,KZ-1)/VO(IX,JY,KZ-1)))/2.0
  WTRAPZ=PRWP*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
+        (DW(IX,JY,KZ-1)/VW(IX,JY,KZ-1)))/2.0

  GDAR1Z = GTRANZ*(PG(IX,JY,KZ+1) - PG(IX,JY,KZ))/DZ**2
  GDAR2Z = GTRAPZ*(PG(IX,JY,KZ-1) - PG(IX,JY,KZ))/DZ**2
  ODAR1Z = OTRANZ*(PO(IX,JY,KZ+1) - PO(IX,JY,KZ))/DZ**2
  ODAR2Z = OTRAPZ*(PO(IX,JY,KZ-1) - PO(IX,JY,KZ))/DZ**2
  WDAR1Z = WTRANZ*(PW(IX,JY,KZ+1) - PW(IX,JY,KZ))/DZ**2
  WDAR2Z = WTRAPZ*(PW(IX,JY,KZ-1) - PW(IX,JY,KZ))/DZ**2
 ELSEIF (KZ .EQ. NZ) THEN
  PRGP = APRZ*RPG(IX,JY,KZ-1)
  PROP = APRZ*RPO(IX,JY,KZ-1)
  PRWP = APRZ*RPW(IX,JY,KZ-1)
  GTRANZ = PRG*DG(IX,JY,KZ)/VG(IX,JY,KZ)
  OTRANZ = PRO*DO(IX,JY,KZ)/VO(IX,JY,KZ)
  WTRANZ = PRW*DW(IX,JY,KZ)/VW(IX,JY,KZ)
  GTRAPZ=PRGP*((DG(IX,JY,KZ)/VG(IX,JY,KZ))+
+        (DG(IX,JY,KZ-1)/VG(IX,JY,KZ-1)))/2.0
  OTRAPZ=PROP*((DO(IX,JY,KZ)/VO(IX,JY,KZ))+
+        (DO(IX,JY,KZ-1)/VO(IX,JY,KZ-1)))/2.0
  WTRAPZ=PRWP*((DW(IX,JY,KZ)/VW(IX,JY,KZ))+
+        (DW(IX,JY,KZ-1)/VW(IX,JY,KZ-1)))/2.0
  GDAR1Z = 0.000
  GDAR2Z = GTRAPZ*(PG(IX,JY,KZ-1) - PG(IX,JY,KZ))/DZ**2
  ODAR1Z = 0.000
  ODAR2Z = OTRAPZ*(PO(IX,JY,KZ-1) - PO(IX,JY,KZ))/DZ**2
  WDAR1Z = 0.000
  WDAR2Z = WTRAPZ*(PW(IX,JY,KZ-1) - PW(IX,JY,KZ))/DZ**2
 ENDIF
 ELSE
```

```
      GDAR1Z = 0.00
      GDAR2Z = 0.00
      WDAR1Z = 0.00
      WDAR2Z = 0.00
      ODAR1Z = 0.00
      ODAR2Z = 0.00
      ENDIF
      RETURN
      END


      SUBROUTINE DARCY44(IX,JY,KZ)

C***** THIS ROUTINE CALCULTES THE MOBILITY TERMS USING ONE POINT
C     UP STREAM WEIGHTING


      INCLUDE 'C:\PROJECT\INCLUDE\COMMON3D.INC'
       IPX = IX - 1
       JPY = JY - 1
       PRG = APR*RPG(IX,JY,KZ)
       PRO = APR*RPO(IX,JY,KZ)
       PRW = APR*RPW(IX,JY,KZ)

      IF (IX .EQ. 1) THEN
       GTRAN = PRG*(DG(IX,JY,KZ))/(VG(IX,JY,KZ))
       OTRAN = PRO*(DO(IX,JY,KZ))/(VO(IX,JY,KZ))
       WTRAN = PRW*(DW(IX,JY,KZ))/(VW(IX,JY,KZ))
       GDARC1 = GTRAN*(PG(IX+1,JY,KZ) - PG(IX,JY,KZ))/DX**2
       GDARC2 = 0.000
       ODARC1 = OTRAN*(PO(IX+1,JY,KZ) - PO(IX,JY,KZ))/DX**2
       ODARC2 = 0.000
       WDARC1 = WTRAN*(PW(IX+1,JY,KZ) - PW(IX,JY,KZ))/DX**2
       WDARC2 = 0.000

      ELSEIF (IX .GT. 1 .AND. IX .NE. NX) THEN
       GTRAN = PRG*(DG(IX,JY,KZ))/(VG(IX,JY,KZ))
       OTRAN = PRO*(DO(IX,JY,KZ))/(VO(IX,JY,KZ))
       WTRAN = PRW*(DW(IX,JY,KZ))/(VW(IX,JY,KZ))
       PRGP = APR*RPG(IX-1,JY,KZ)
       PROP = APR*RPO(IX-1,JY,KZ)
       PRWP = APR*RPW(IX-1,JY,KZ)
       GTRANP = PRGP*(DG(IX-1,JY,KZ))/(VG(IX-1,JY,KZ))
       OTRANP = PROP*(DO(IX-1,JY,KZ))/(VO(IX-1,JY,KZ))
       WTRANP = PRWP*(DW(IX-1,JY,KZ))/(VW(IX-1,JY,KZ))

       GDARC1 = GTRAN*(PG(IX+1,JY,KZ) - PG(IX,JY,KZ))/DX**2
       GDARC2 = GTRANP*(PG(IX-1,JY,KZ) - PG(IX,JY,KZ))/DX**2
       ODARC1 = OTRAN*(PO(IX+1,JY,KZ) - PO(IX,JY,KZ))/DX**2
       ODARC2 = OTRANP*(PO(IX-1,JY,KZ) - PO(IX,JY,KZ))/DX**2
       WDARC1 = WTRAN*(PW(IX+1,JY,KZ) - PW(IX,JY,KZ))/DX**2
       WDARC2 = WTRANP*(PW(IX-1,JY,KZ) - PW(IX,JY,KZ))/DX**2
      ELSEIF (IX .EQ. NX) THEN
       PRGP = APR*RPG(IX-1,JY,KZ)
       PROP = APR*RPO(IX-1,JY,KZ)
       PRWP = APR*RPW(IX-1,JY,KZ)
       GTRAN = PRG*DG(IX,JY,KZ)/VG(IX,JY,KZ)
       OTRAN = PRO*DO(IX,JY,KZ)/VO(IX,JY,KZ)
```

```
       WTRAN = PRW*DW(IX,JY,KZ)/VW(IX,JY,KZ)
       GTRANP = PRGP*(DG(IX-1,JY,KZ))/(VG(IX-1,JY,KZ))
       OTRANP = PROP*(DO(IX-1,JY,KZ))/(VO(IX-1,JY,KZ))
       WTRANP = PRWP*(DW(IX-1,JY,KZ))/(VW(IX-1,JY,KZ))
       GDARC1 = 0.000
       GDARC2 = GTRANP*(PG(IX-1,JY,KZ) - PG(IX,JY,KZ))/DX**2
       ODARC1 = 0.000
       ODARC2 = OTRANP*(PO(IX-1,JY,KZ) - PO(IX,JY,KZ))/DX**2
       WDARC1 = 0.000
       WDARC2 = WTRANP*(PW(IX-1,JY,KZ) - PW(IX,JY,KZ))/DX**2
      ENDIF
       IF (NY .NE. 1) THEN
C        IF (JY .NE. NY ) THEN
C        JNY = JY + 1
C        CALL PHYPRO(IX,JNY)
C        CALL MOLWT(IX,JNY)
C        ENDIF
       PRGY = APRY*RPG(IX,JY,KZ)
       PROY = APRY*RPO(IX,JY,KZ)
       PRWY = APRY*RPW(IX,JY,KZ)
      IF (JY .EQ. 1) THEN
       GTRANY = PRGY*(DG(IX,JY,KZ))/(VG(IX,JY,KZ))
       OTRANY = PROY*(DO(IX,JY,KZ))/(VO(IX,JY,KZ))
       WTRANY = PRWY*(DW(IX,JY,KZ))/(VW(IX,JY,KZ))
       GDAR1Y = GTRANY*(PG(IX,JY+1,KZ) - PG(IX,JY,KZ))/DY**2
       GDAR2Y = 0.000
       ODAR1Y = OTRANY*(PO(IX,JY+1,KZ) - PO(IX,JY,KZ))/DY**2
       ODAR2Y = 0.000
       WDAR1Y = WTRANY*(PW(IX,JY+1,KZ) - PW(IX,JY,KZ))/DY**2
       WDAR2Y = 0.000
       GGRAV1 = 0.00
       WGRAV1 = 0.00
       OGRAV1 = 0.00
       GGRAV2 = 0.00
       WGRAV2 = 0.00
       OGRAV2 = 0.00
      ELSEIF (JY .GT. 1 .AND. JY .NE. NY) THEN
       GTRANY = PRGY*(DG(IX,JY,KZ))/(VG(IX,JY,KZ))
       OTRANY = PROY*(DO(IX,JY,KZ))/(VO(IX,JY,KZ))
       WTRANY = PRWY*(DW(IX,JY,KZ))/(VW(IX,JY,KZ))
       PRGPY = APRY*RPG(IX,JY-1,KZ)
       PROPY = APRY*RPO(IX,JY-1,KZ)
       PRWPY = APRY*RPW(IX,JY-1,KZ)
       GTRAPY = PRGPY*(DG(IX,JY-1,KZ))/(VG(IX,JY-1,KZ))
       OTRAPY = PROPY*(DO(IX,JY-1,KZ))/(VO(IX,JY-1,KZ))
       WTRAPY = PRWPY*(DW(IX,JY-1,KZ))/(VW(IX,JY-1,KZ))

       GDAR1Y = GTRANY*(PG(IX,JY+1,KZ) - PG(IX,JY,KZ))/DY**2
       GDAR2Y = GTRAPY*(PG(IX,JY-1,KZ) - PG(IX,JY,KZ))/DY**2
       ODAR1Y = OTRANY*(PO(IX,JY+1,KZ) - PO(IX,JY,KZ))/DY**2
       ODAR2Y = OTRAPY*(PO(IX,JY-1,KZ) - PO(IX,JY,KZ))/DY**2
       WDAR1Y = WTRANY*(PW(IX,JY+1,KZ) - PW(IX,JY,KZ))/DY**2
       WDAR2Y = WTRAPY*(PW(IX,JY-1,KZ) - PW(IX,JY,KZ))/DY**2
       GGRAV1 = -(APRY*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)*
     +        WTMOLG(IX,JY,KZ)*DG(IX,JY,KZ))*(9.8D-06/(DY))
       GGRAV2 = -(APRY*RPG(IX,JY-1,KZ)*DG(IX,JY-1,KZ)/VG(IX,JY-1,K
     +         DG(IX,JY-1,KZ)*WTMOLG(IX,JY-1,KZ))*(9.8D-06/(DY))
       WGRAV1 = -(APRY*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)*
```

```
    +        AMWT*DW(IX,JY,KZ))*(9.8D-06/(DY))
      WGRAV2 = -(APRY*RPW(IX,JY-1,KZ)*DW(IX,JY-1,KZ)/VW(IX,JY-1,K
    +         DW(IX,JY-1,KZ)*AMWT)*(9.8D-06/(DY))
      OGRAV1 = -(APRY*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)*
    +       WTMOLO(IX,JY,KZ)*DO(IX,JY,KZ))*(9.8D-06/(DY))
      OGRAV2 = -(APRY*RPO(IX,JY-1,KZ)*DO(IX,JY-1,KZ)/VO(IX,JY-1,K
    +        DO(IX,JY-1,KZ)*WTMOLO(IX,JY-1,KZ))*(9.8D-06/(DY))
     ELSEIF (JY .EQ. NY) THEN
      PRGPY = APRY*RPG(IX,JY-1,KZ)
      PROPY = APRY*RPO(IX,JY-1,KZ)
      PRWPY = APRY*RPW(IX,JY-1,KZ)
      GTRANY = PRGY*DG(IX,JY,KZ)/VG(IX,JY,KZ)
      OTRANY = PROY*DO(IX,JY,KZ)/VO(IX,JY,KZ)
      WTRANY = PRWY*DW(IX,JY,KZ)/VW(IX,JY,KZ)
      GTRAPY = PRGPY*(DG(IX,JY-1,KZ))/(VG(IX,JY-1,KZ))
      OTRAPY = PROPY*(DO(IX,JY-1,KZ))/(VO(IX,JY-1,KZ))
      WTRAPY = PRWPY*(DW(IX,JY-1,KZ))/(VW(IX,JY-1,KZ))
      GDAR1Y = 0.000
      GDAR2Y = GTRAPY*(PG(IX,JY-1,KZ) - PG(IX,JY,KZ))/DY**2
      ODAR1Y = 0.000
      ODAR2Y = OTRAPY*(PO(IX,JY-1,KZ) - PO(IX,JY,KZ))/DY**2
      WDAR1Y = 0.000
      WDAR2Y = WTRAPY*(PW(IX,JY-1,KZ) - PW(IX,JY,KZ))/DY**2
      GGRAV1 = -(APRY*RPG(IX,JY,KZ)*DG(IX,JY,KZ)/VG(IX,JY,KZ)*
    +       WTMOLG(IX,JY,KZ)*DG(IX,JY,KZ))*(9.8D-06/(DY))
      GGRAV2 = -(APRY*RPG(IX,JY-1,KZ)*DG(IX,JY-1,KZ)/VG(IX,JY-1,K
    +        DG(IX,JY-1,KZ)*WTMOLG(IX,JY-1,KZ))*(9.8D-06/(DY))
      WGRAV1 = -(APRY*RPW(IX,JY,KZ)*DW(IX,JY,KZ)/VW(IX,JY,KZ)*
    +        AMWT*DW(IX,JY,KZ))*(9.8D-06/(DY))
      WGRAV2 = -(APRY*RPW(IX,JY-1,KZ)*DW(IX,JY-1,KZ)/VW(IX,JY-1,K
    +         DW(IX,JY-1,KZ)*AMWT)*(9.8D-06/(DY))
      OGRAV1 = -(APRY*RPO(IX,JY,KZ)*DO(IX,JY,KZ)/VO(IX,JY,KZ)*
    +       WTMOLO(IX,JY,KZ)*DO(IX,JY,KZ))*(9.8D-06/(DY))
      OGRAV2 = -(APRY*RPO(IX,JY-1,KZ)*DO(IX,JY-1,KZ)/VO(IX,JY-1,K
    +        DO(IX,JY-1,KZ)*WTMOLO(IX,JY-1,KZ))*(9.8D-06/(DY))

     ENDIF
      ELSE
      GDAR1Y = 0.000
      GDAR2Y = 0.000
      ODAR1Y = 0.000
      ODAR2Y = 0.000
      WDAR1Y = 0.000
      WDAR2Y = 0.000

      GGRAV1 = 0.0
      OGRAV1 = 0.00
      WGRAV1 = 0.00
      GGRAV2 = 0.0
      OGRAV2 = 0.00
      WGRAV2 = 0.00
     ENDIF
C      GGRAV1 = 0.0
C       OGRAV1 = 0.00
C       WGRAV1 = 0.00
C      GGRAV2 = 0.0
C      OGRAV2 = 0.00
C      WGRAV2 = 0.00
```

```
      IF (NZ .NE. 1) THEN
C       IF (KZ .NE. NZ) THEN
C         KNZ = KZ + 1
C         CALL PHYPRO(IX,JY,KNZ)
C        CALL MOLWT(IX,JY,KNZ)
C        ENDIF
       PRGZ = APRZ*RPG(IX,JY,KZ)
       PROZ = APRZ*RPO(IX,JY,KZ)
       PRWZ = APRZ*RPW(IX,JY,KZ)
      IF (KZ .EQ. 1) THEN
       GTRANZ=PRGZ*(DG(IX,JY,KZ)/VG(IX,JY,KZ))
       OTRANZ=PROZ*(DO(IX,JY,KZ)/VO(IX,JY,KZ))
       WTRANZ=PRWZ*(DW(IX,JY,KZ)/VW(IX,JY,KZ))
       GDAR1Z = GTRANZ*(PG(IX,JY,KZ+1) - PG(IX,JY,KZ))/DZ**2
       GDAR2Z = 0.000
       ODAR1Z = OTRANZ*(PO(IX,JY,KZ+1) - PO(IX,JY,KZ))/DZ**2
       ODAR2Z = 0.000
       WDAR1Z = WTRANZ*(PW(IX,JY,KZ+1) - PW(IX,JY,KZ))/DZ**2
       WDAR2Z = 0.000
      ELSEIF (KZ .GT. 1 .AND. KZ .NE. NZ) THEN
       GTRANZ=PRGZ*(DG(IX,JY,KZ)/VG(IX,JY,KZ))
       OTRANZ=PROZ*(DO(IX,JY,KZ)/VO(IX,JY,KZ))
       WTRANZ=PRWZ*(DW(IX,JY,KZ)/VW(IX,JY,KZ))
       PRGPZ = APRZ*RPG(IX,JY,KZ-1)
       PROPZ = APRZ*RPO(IX,JY,KZ-1)
       PRWPZ = APRZ*RPW(IX,JY,KZ-1)
       GTRAPZ=PRGPZ*(DG(IX,JY,KZ-1)/VG(IX,JY,KZ-1))
       OTRAPZ=PROPZ*(DO(IX,JY,KZ-1)/VO(IX,JY,KZ-1))
       WTRAPZ=PRWPZ*(DW(IX,JY,KZ-1)/VW(IX,JY,KZ-1))

       GDAR1Z = GTRANZ*(PG(IX,JY,KZ+1) - PG(IX,JY,KZ))/DZ**2
       GDAR2Z = GTRAPZ*(PG(IX,JY,KZ-1) - PG(IX,JY,KZ))/DZ**2
       ODAR1Z = OTRANZ*(PO(IX,JY,KZ+1) - PO(IX,JY,KZ))/DZ**2
       ODAR2Z = OTRAPZ*(PO(IX,JY,KZ-1) - PO(IX,JY,KZ))/DZ**2
       WDAR1Z = WTRANZ*(PW(IX,JY,KZ+1) - PW(IX,JY,KZ))/DZ**2
       WDAR2Z = WTRAPZ*(PW(IX,JY,KZ-1) - PW(IX,JY,KZ))/DZ**2
      ELSEIF (KZ .EQ. NZ) THEN
       PRGPZ = APRZ*RPG(IX,JY,KZ-1)
       PROPZ = APRZ*RPO(IX,JY,KZ-1)
       PRWPZ = APRZ*RPW(IX,JY,KZ-1)
       GTRANZ = PRG*DG(IX,JY,KZ)/VG(IX,JY,KZ)
       OTRANZ = PRO*DO(IX,JY,KZ)/VO(IX,JY,KZ)
       WTRANZ = PRW*DW(IX,JY,KZ)/VW(IX,JY,KZ)
       GTRAPZ=PRGPZ*(DG(IX,JY,KZ-1)/VG(IX,JY,KZ-1))
       OTRAPZ=PROPZ*(DO(IX,JY,KZ-1)/VO(IX,JY,KZ-1))
       WTRAPZ=PRWPZ*(DW(IX,JY,KZ-1)/VW(IX,JY,KZ-1))
       GDAR1Z = 0.000
       GDAR2Z = GTRAPZ*(PG(IX,JY,KZ-1) - PG(IX,JY,KZ))/DZ**2
       ODAR1Z = 0.000
       ODAR2Z = OTRAPZ*(PO(IX,JY,KZ-1) - PO(IX,JY,KZ))/DZ**2
       WDAR1Z = 0.000
       WDAR2Z = WTRAPZ*(PW(IX,JY,KZ-1) - PW(IX,JY,KZ))/DZ**2
      ENDIF
      ELSE
      GDAR1Z = 0.00
      GDAR2Z = 0.00
      WDAR1Z = 0.00
      WDAR2Z = 0.00
```

```
      ODAR1Z = 0.00
      ODAR2Z = 0.00
      ENDIF
      RETURN
      END

      PARAMETER (N=72,NX=4,NV=6,NY=3,NZ=1,NN=24,NNZ=72)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON / DELTA / DT,DX,DTMAX,DTST,OLDDT,DY,DZ
      COMMON / IFLAG / IFLAG1,IFLAG3,MAXSTP,IFLAG5
      COMMON / COUNT / ICOUNT,IFREQ,MELIMF
      COMMON / TIME  / TIM,TIME(10000)
      COMMON / PRES  / PG(NX,NY,NZ),PO(NX,NY,NZ),PW(NX,NY,NZ)
      COMMON / TEMP  / TM(NX,NY,NZ)
      COMMON / DENS  / DG(NX,NY,NZ),DO(NX,NY,NZ),DW(NX,NY,NZ)
      COMMON / VISC  / VG(NX,NY,NZ),VO(NX,NY,NZ),VW(NX,NY,NZ)
      COMMON / VAPOR / YW(NX,NY,NZ),YN(NX,NY,NZ),YO(NX,NY,NZ),
     +                 YL(NX,NY,NZ),YH(NX,NY,NZ)
      COMMON / LIQD  / XH(NX,NY,NZ),XL(NX,NY,NZ)
      COMMON / HEAT  / HG(NX,NY,NZ),HO(NX,NY,NZ),HW(NX,NY,NZ),
     +                 HC(NX,NY,NZ)
      COMMON / ENTER / EG(NX,NY,NZ),EO(NX,NY,NZ),EW(NX,NY,NZ),
     +                 ER(NX,NY,NZ)
      COMMON / PRM2  / RPG(NX,NY,NZ),RPW(NX,NY,NZ),RPO(NX,NY,NZ)
      COMMON / PREV1 / PGP(NX,NY,NZ),TMP(NX,NY,NZ),SGP(NX,NY,NZ),
     +                 SWP(NX,NY,NZ),SOP(NX,NY,NZ)
      COMMON / PREV2 / YWP(NX,NY,NZ),YNP(NX,NY,NZ),YOP(NX,NY,NZ),
     +                 YLP(NX,NY,NZ),YHP(NX,NY,NZ)
      COMMON / PREV3 / XHP(NX,NY,NZ),XLP(NX,NY,NZ),DGP(NX,NY,NZ),
     +                 DOP(NX,NY,NZ),DWP(NX,NY,NZ),CCP(NX,NY,NZ)
      COMMON / PREV4 / EGP(NX,NY,NZ),EOP(NX,NY,NZ),EWP(NX,NY,NZ),
     +                 ERP(NX,NY,NZ),HCP(NX,NY,NZ)
      COMMON / VARIAB/ X(N),A(N,N),RHS(N),BETA(N)
      COMMON / EQUT  / EALPG,EALSW,EALSO,EALTM,EALYO,EALXH
      COMMON / DARC1 / GDARC1,GDARC2,GTRAN,ODARC1,ODARC2,OTRAN
      COMMON / DARC2 / WDARC1,WDARC2,WTRAN,PRCN
      COMMON / DARC3 / GDAR1Y,GDAR2Y,GTRANY,ODAR1Y,ODAR2Y,OTRANY
      COMMON / DARC4 / WDAR1Y,WDAR2Y,WTRANY
      COMMON / DARC5 / GDAR1Z,GDAR2Z,GTRANZ,ODAR1Z,ODAR2Z,OTRANZ
      COMMON / DARC6 / WDAR1Z,WDAR2Z,WTRANZ
      COMMON / RERAT / RA(NX,NY,NZ),RB(NX,NY,NZ),RC(NX,NY,NZ),
     +                 RD(NX,NY,NZ)
      COMMON / OUTO  / OTCN(NX,NY,NZ)
      COMMON / OUTALL/ OTNR(NX,NY,NZ),OTWTV(NX,NY,NZ),OTWTL(NX,NY,
     +                 OTLTV(NX,NY,NZ),OTLTL(NX,NY,NZ),OTOX(NX,NY,NZ)
     +                 OTHVL(NX,NY,NZ),OTHVV(NX,NY,NZ)
      COMMON / COK   / CC(NX,NY,NZ)
      COMMON / EQULB / CLT(NX,NY,NZ),CHV(NX,NY,NZ),CWT(NX,NY,NZ)
      COMMON / SAT   / SG(NX,NY,NZ),SO(NX,NY,NZ),SW(NX,NY,NZ)
      COMMON / STEP  / ITER,MAXITR,URMX
      COMMON / TLOS  / UMAX(NV),TOLX(NV)
      COMMON / INLET / HIN(NX,NY,NZ),QW(NX,NY,NZ),QH(NX,NY,NZ),
     +     QL(NX,NY,NZ),QN(NX,NY,NZ),QO(NX,NY,NZ),TI(NX,NY,NZ),TII
      COMMON / INLET1/ QW1,QH1,QL1,QN1,QO1
      COMMON / DEF1  / PGD(NX,NY,NZ),SOD(NX,NY,NZ),SWD(NX,NY,NZ),
     +                 YOD(NX,NY,NZ),XHD(NX,NY,NZ),TMD(NX,NY,NZ)
      COMMON / DEF2  / DNGP,DNOL,DNWT,DNOX,DNHL,DNTE
      COMMON / DELT  / DEL(NV)
```

```
      COMMON / SOLV1 / TMAT(N,N),TRI(N,N),THS(N),RVT(N),DDEL(N),IP
      COMMON / SOLV2 / RRR(8100)
      COMMON / EQLB/ AKLT,BKLT,CKLT,AKHV,BKHV,CKHV,AKWT,BKWT,CKWT
      COMMON / ENTH/ AWT,AHV,ALT,ANR,AOX,ACK,ARK
      COMMON / ENHV/ AHWT,AHOL
      COMMON / DENS1/ ADWT,BDWT,CDWT,ADOL,BDOL,CDLT,CDHV
      COMMON / VISC1/ AVWT,BVWT,CVWT,DVWT,AVLT,BVLT,AVHV,BVHV,AVGS
     +          CVGS
      COMMON / VISC2/ AVOXG,AVNRG,AVLTG,AVHVG,AVWTG,BVOXG,BVNRG,BV
     +           BVHVG,BVWTG
      COMMON / REF / TREF,PREF
      COMMON / TC  / TCWT,TCLT,TCHV
      COMMON / CONS/ APR,APRY,APRZ,POROS,THCO,THCCAP,WLP
      COMMON / PRM / CRWRO,CRGRO,CROCW,SWC,SORW,SORG,SGC,ZW,ZG,ZOG
      COMMON / RESV/ RESL,RESW,RESD
      COMMON / RECT/ ARA,ARB,ARC,ARD,EA,EB,EC,ED
      COMMON / HREC/ HA,HB,HRC,HD
      COMMON / SREC/ S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12
      COMMON / RGAS/ R,CKMAX
      COMMON /AM/ AMOX,AMNR,AMLT,AMHV,AMWT,AMCK
      COMMON /OUT2/ AOTNR,AOTOX,AOTLTL,AOTLTV,AOTHVL,AOTHVV,
     +         AOTWTL,AOTWTV
      COMMON /OUT3/ VOTNR,VOTOX,VOTLTL,VOTLTV,VOTHVL,VOTHVV,
     +         VOTWTL,VOTWTV
      COMMON /OUT4/ ROTLTL,ROTLTV,ROTHVL,ROTHVV,
     +         ROTWTL,ROTWTV
      COMMON /VOM / VOLUMB
      COMMON /PPPC / PCWT,PCLT,PCHV
      COMMON /OOIL / VOOIP
      COMMON /OWT /  VOWIP
      COMMON /STDN/ STDWT,STDLT,STDHV
      COMMON /HEAT1/HREAC(NX,NY,NZ),HLOSS(NX,NY,NZ),PHL(NX,NY,NZ),
     +         QHL(NX,NY,NZ),PPHL(NX,NY,NZ),
     +         PQHL(NX,NY,NZ),CAPEN(NX,NY,NZ),PCAPEN(NX,NY,NZ)
      COMMON /DIFFS/ THDIF,DIFL,PDIFL
      COMMON /HEAT2/ TOTHET,TOTLOS
      COMMON /VPPPP/ VGP(NX,NY,NZ),VOP(NX,NY,NZ),VWP(NX,NY,NZ)
      COMMON /LUBBB/ B(N),VV(N)
      COMMON /LUBB2/ INDX(N)
      COMMON /GRAVTY/ GGRAV1,WGRAV1,OGRAV1,GGRAV2,WGRAV2,OGRAV2
      COMMON /MOLWT1/ WTMOLG(NX,NY,NZ),WTMOLO(NX,NY,NZ)
      COMMON /OMEG/ AAX(NX,NY,NZ),AAY(NX,NY,NZ),OMGX(NX,NY,NZ),
     +         OMGY(NX,NY,NZ),AAZ(NX,NY,NZ),OMGZ(NX,NY,NZ),
     +         OMOX(NX,NY,NZ),OMOY(NX,NY,NZ),OMWX(NX,NY,NZ),
     +         OMOZ(NX,NY,NZ),OMWY(NX,NY,NZ),OMWZ(NX,NY,NZ),
     +         RPGP(NX,NY,NZ),RPOP(NX,NY,NZ),RPWP(NX,NY,NZ)
C     EXTERNAL BLKDAT
```

Simplified Input Data

```
Starting Time Step   Maximum Time For One Step  Max. No.of Steps
0.07               2.0                    9999
URMX
0.5
Productivity Constant
1.0
Time Step Norms.(PG,SO,SW,YO,XH,TM)
400.0 0.20 0.20 0.20 0.20 40.0
Maximum Iteration Number   Output Freq.
30                        5
Convergance Tolerances
0.01  0.01 0.01 0.01 0.001 0.001
Initial Pressure (KPa)
455.0 455.0 455.0 455.0
Initial Temperature (K)
367.0 367.0 367.0 367.0
Initial Oxygen Mole Fraction In The Gas Phase
0.00001 0.00001 0.00001 0.00001
Initial Heavy Oil Mole Fraction In The Oil Phase
0.99999 0.99999 0.99999 0.99999
Initial Oil Saturation
0.50 0.50 0.50 0.50
Initial Water Saturation
0.2 0.2 0.2 0.2
Initial Coke Concentration (mol/m**3)
0.00 0.00 0.00 0.00
Injection Flowrate (m**3/day)
0.00 0.00 0.00 0.00 1.36D+05
Injection Temperature (K)
367.0
```

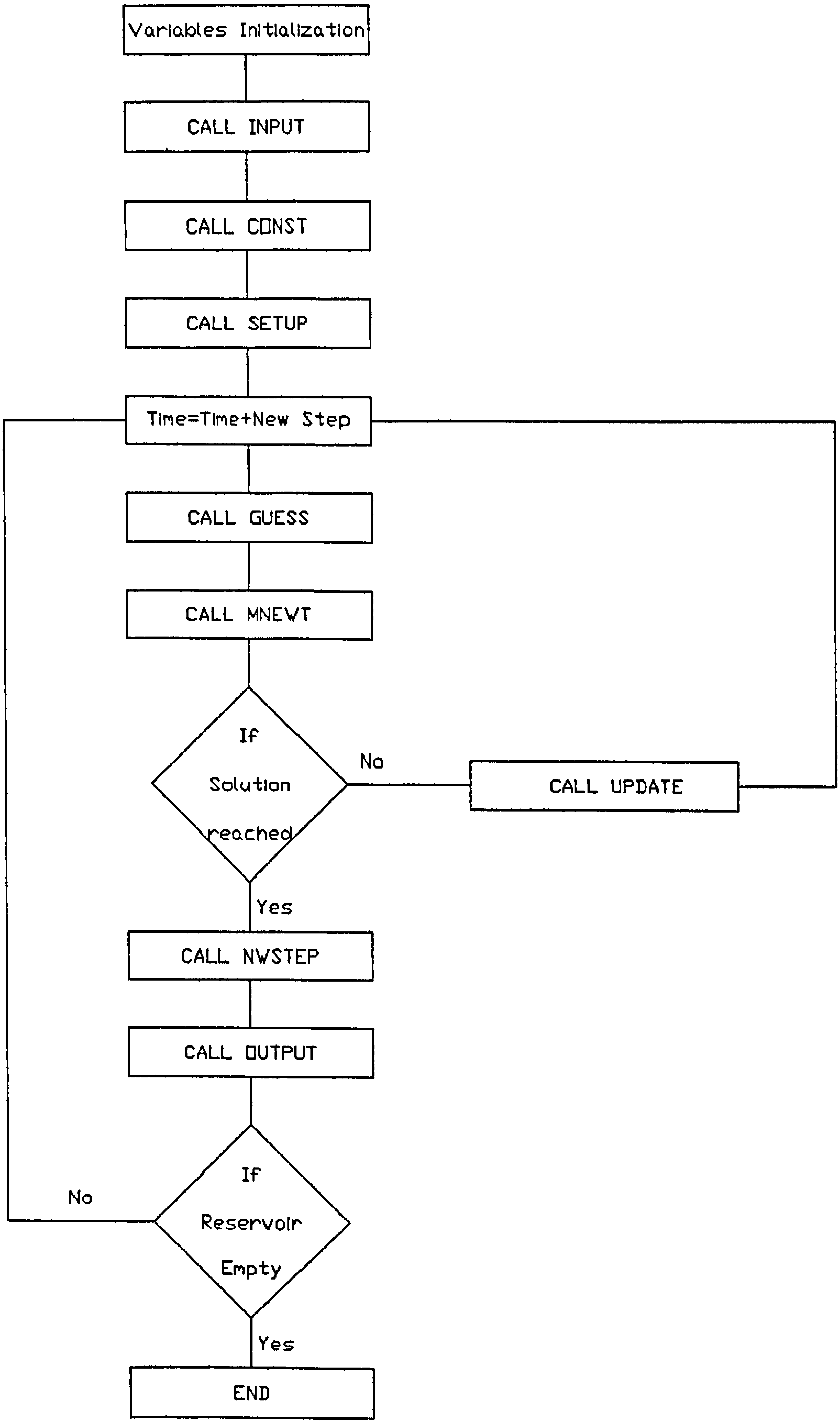


Figure C.1 Subroutine SYSTEM

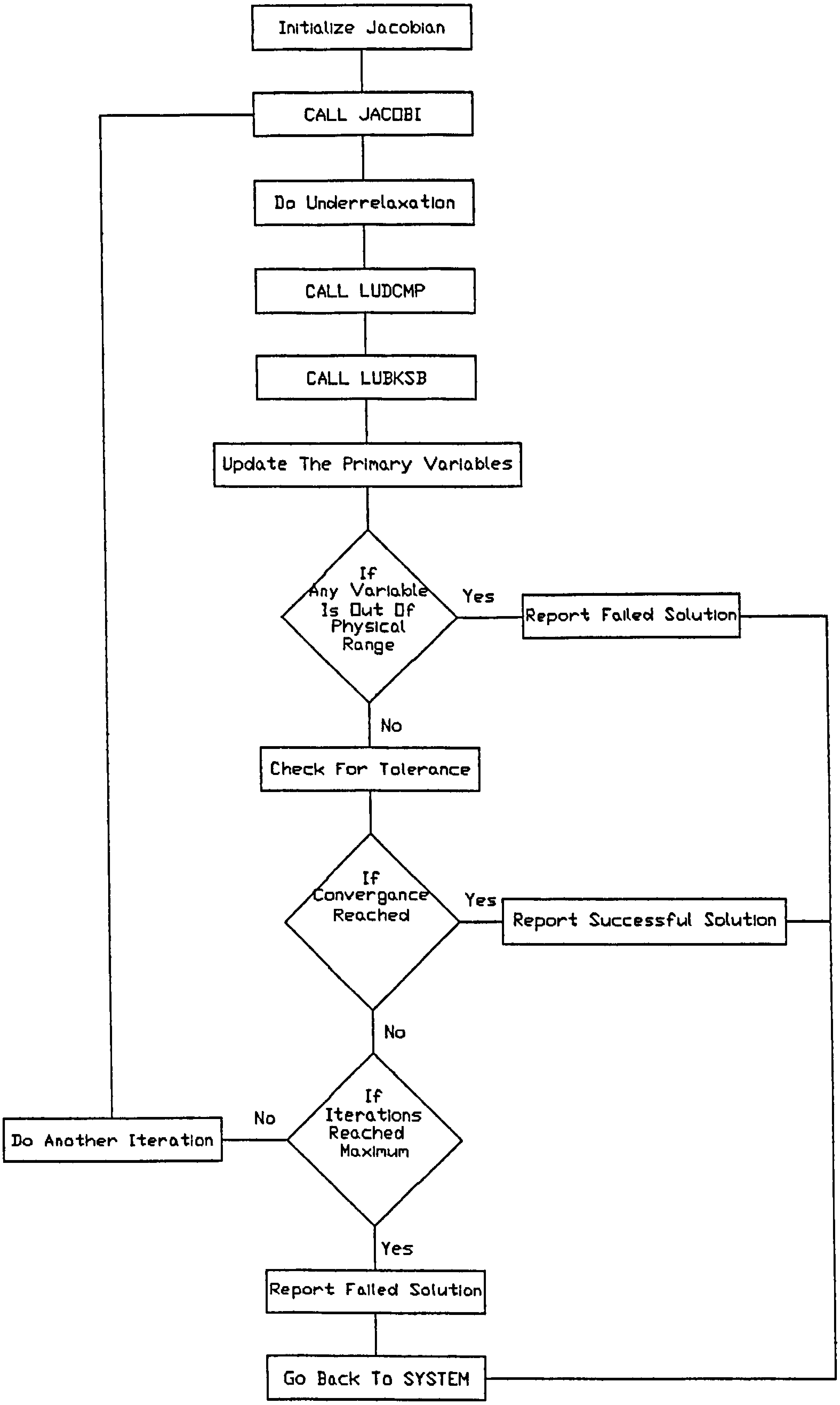


Figure C.2 Subroutine MNEWT

APPENDIX D

Input Data For The Model

General Data	
Reservoir length, m	50
Reservoir Width, m	35
Reservoir Thickness, m	6.5
Absolute Permeability, m^{-2}	$4.1*10^{-12}$
Porosity	0.38
Thermal Conductivity of Formation kJ/Day.m.K	$2.4*10^{2}$
Number of Grid Blocks	10

Initial Conditions	
Oil Saturation	0.50
Water Saturation	0.20
Gas Saturation	0.30
Reservoir Temperature, K	367
Reservoir Pressure, kPa	410
Oxygen Mole Fraction	0.00001
Light Oil Mole Fraction in Liquid Phase	0.00001
Heavy Oil Mole Fraction in Liquid Phase	0.99999

Injection/Production Condition	
Oxygen Mole Fraction	1.00
Gas Injections Rate, (gmol/day)	$1.36\ 10^5$
Gas injection Temperature, K	367
Well Pressure, kPa	410
Production Grid Number	10

Physical Properties Data

Relative Permeability

The relative permeability of each phase is calculated as;

$$K_{rw} = K_{rwro}[(S_w - S_{wc})/(1 - S_{wc} - S_{orw})]^{Z_w}$$

$$K_{rg} = K_{rgro}[(S_g - S_{gc})/(1 - S_{wc} - S_{org} - S_{gc})]^{Z_g}$$

$$K_{rog} = K_{rocw}[1 - (S_g - S_{gc})/(1 - S_{wc} - S_{org} - S_{gc})]^{Z_{og}}$$

$$K_{row} = K_{rocw}[1 - (S_w - S_{wc})/(1 - S_{orw} - S_{wc})]^{Z_{ow}}$$

$$K_{ro} = K_{rocw}[(K_{row}/K_{rocw} + K_{rw})(K_{rog}/K_{rocw} + K_{rg}) - K_{rg} - K_{rw}]$$

The following values were used for the above constants;

$K_{rwro} = 0.25$; $K_{rgro} = 0.7$; $K_{rocw} = 1.0$

$S_{wc} = 0.20$; $S_{orw} = 0.3$; $S_{org} = 0.09$ $S_{gc} = 0.05$

$Z_w = 3.0$; $Z_g = 1.0$; $Z_{og} = 3.0$ $Z_{ow} = 3.0$

Reactions Data

The model includes the following four reactions:

i) Light Oil Oxidation

$$Oil_3 + s_1O_2 \xrightarrow{r_A} s_2CO_x + s_3H_2O$$

ii) Heavy Oil Oxidation

$$Oil_4 + s_4O_2 \xrightarrow{r_B} s_5CO_x + s_6H_2O$$

iii) Heavy Oil Cracking

$$Oil_4 \xrightarrow{r_C} s_7Oil_3 + s_8Coke + s_9Gas_2$$

iv) Coke Oxidation

$$Coke + s_{10}O_2 \xrightarrow{r_D} s_{11}CO_x + s_{12}H_2O$$

The kinetic reaction rate expressions which are taken from Crookston et al and used for the above reactions are:

$$r_A = A_A\, e^{(-E_A/RT)}\, (Y_1)\, (P_o + P_{cg})(\phi\, S_o\, \rho_o\, X_3)$$

$$r_B = A_B\, e^{(-E_B/RT)}\, (Y_1)\, (P_o + P_{cg})(\phi\, S_o\, \rho_o\, X_4)$$

$$r_C = A_C\, e^{(-E_C/RT)}(\phi\, S_o\, \rho_o\, X_4)(1 - (\frac{C_C}{C_{max}})^5)$$

$$r_D = A_D\, e^{(-E_D/RT)}\, (Y_1)\, (P_o + P_{cg})(C_C)$$

j	A_j (kPa day)$^{-1}$	E_j (kJ/gmol)	H_j (kJ/gmol)
A	$1.45\ 10^5$	77.46	$2.21\ 10^3$
B	$1.45\ 10^5$	77.45	$8.12\ 10^3$
C	$3.00\ 10^5$ *	66.99	46.5
D	$1.45\ 10^5$	54.43	$5.23\ 10^2$

*units day^{-1}

$s_1 = 5.0$; $s_2 = 4$; $s_3 = 3.0$; $s_4 = 18$; $s_5 = 13$

$s_6 = 12$; $s_7 = 2.0$; $s_8 = 1.33$; $s_9 = 4.67$

$s_{10} = 1.25$; $s_{11} = 0.5$; $s_{12} = 1.0$

Water Density

$$\rho_w = \rho_{wR}\, [1 + \alpha_w(P_w - P_R) - \beta_w(T - T_R)]$$

Where $\rho_{wR} = 5.55*10^4$ gmol/m^3

$\alpha_w = 4.5*10^{-6}$ kPa^{-1}

$\beta_w = 2.1*10^{-4}$ K^{-1}

Oil Density

$$\rho_o = \rho_{oR}\, [1 + \alpha_o(P_o - P_R) - \beta_o(T - T_R)]$$

$$\rho_{oR} = \left(\frac{X_3}{\rho^3_{oR}} + \frac{X_4}{\rho^4_{oR}} \right)^{-1}$$

$\rho^{3}_{oR} = 8.37\ 10^{3}$ gmol/m^{3}

$\rho^{4}_{oR} = 4.16\ 10^{3}$ gmol/m^{3}

$\alpha_{o} = 8.0\ 10^{-6}$ kPa^{-1}

$\beta_{o} = 2.0\ 10^{-4}$ K^{-1}

Gas Viscosity

The gas phase viscosity is related to temperature and composition according to;

$$\mu_{g} = \sum_{l=1}^{5} Y_{l}\ \mu_{1l}\ (T)^{\mu_{2l}}$$

$\mu_{11} = 3.883*10^{-15}$ kPa^{-1}

$\mu_{12} = 3.7169*10^{-15}$ kPa^{-1}

$\mu_{13} = 4.3638*10^{-16}$ kPa^{-1}

$\mu_{14} = 8.6845*10^{-17}$ kPa^{-1}

$\mu_{15} = 1.9676*10^{-17}$ kPa^{-1}

$\mu_{21} = 0.721$

$\mu_{22} = 0.702$

$\mu_{23} = 0.943$

$\mu_{24} = 1.102$

$\mu_{25} = 1.116$

Water Viscosity

The water phase viscosity is related to temperature by;

$$\mu_w = 10^{-9} /(12.1 + 2.88(T - 273.15) + 7.78\text{x}10^{-4}(T - 273.15)^2)$$

Oil Viscosity

The oil phase viscosity is calculated using the following dependence of composition;

$$\mu_o = (\mu_{o3})^{X_3} (\mu_{o4})^{X_4}$$

$$\mu_{o3} = AV_3 \; e^{(BV_3/T)}$$

$$\mu_{o4} = AV_4 \; e^{(BV_4/T)}$$

$AV_3 = 2.41 \; 10^{-13}$ kPa Day

$BV_3 = 533.1$ K

$AV_4 = 4.19 \; 10^{-15}$ kPa Day

$BV_4 = 4714$ K

Equilibrium Constants

$$K_3 = \frac{1}{P_g} \; e^{\left(\frac{a_3 - b_3}{T + c_3}\right)}$$

$$K_4 = \frac{1}{P_g} \; e^{\left(\frac{a_4 - b_4}{T + c_4}\right)}$$

$$K_5 = \frac{1}{P_g} \left[\frac{T-b_5}{a_5}\right]^{c_5}$$

$a_3 = 13.71$ K $\quad b_3 = -1872.0$ K $\quad c_3 = 25.16$ K

$a_4 = 14.06$ K $\quad b_4 = -3744.0$ K $\quad c_4 = 92.85$ K

$a_5 = 41.818$ K $\quad b_5 = 255.37$ K $\quad c_5 = 4.464$

Molar Mass

$M_1 = 32$ g/gmol

$M_2 = 40.0$ g/gmol

$M_3 = 44.0$ g/gmol

$M_4 = 170.0$ g/gmol

$M_5 = 18.0$ g/gmol

$M_6 = 13$ g/gmol