# A MODIFIED BOUSSINESQ-BASED APPROACH FOR NUMERICAL INVESTIGATION OF THERMAL FLOODING PROCESS WITH EXPERIMENTAL VERIFICATION

**OBEMBE ABIOLA DAVID**

**DEPARTMENT OF PETROLEUM ENGINEERING**

**MAY 2016**

# A MODIFIED BOUSSINESQ-BASED APPROACH FOR NUMERICAL INVESTIGATION OF THERMAL FLOODING PROCESS WITH EXPERIMENTAL VERIFICATION

BY

**OBEMBE ABIOLA DAVID**

A Dissertation Presented to the

DEANSHIP OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

# DOCTOR OF PHILOSOPHY

**In**

**PETROLEUM ENGINEERING**

**MAY 2016**

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

**DEANSHIP OF GRADUATE STUDIES**

This thesis, written by **OBEMBE ABIOLA DAVID** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN PETROLEUM ENGINEERING.**

Dr. M. Enamul Hossain
(Advisor)

Dr. Abdullah S. Sultan
Department Chairman

Dr. Sidqi A. Abu-Khamsin
(Co-Advisor)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Hasan Y. Al-Yousef
(Member)

21/9/16
Date

Dr. Mustapha Kassem
(Member)

Dr. Ben-Mansour Rached
(Member)

*Dedication*

# ACKNOWLEDGMENTS

I thank the Almighty God for giving me the privilege to study at King Fahd University of Petroleum and Minerals (KFUPM). I take this opportunity to thank the many people who contributed to my success and wonderful experience here.

First of all, I would like to thank my advisor Dr. Enamul Hossain for his guidance, constant encouragement, critical feedback, and for fostering a research environment with great intellectual freedom. He has been an excellent role model and has had a profound impact on my outlook towards research. I would also like to thank my committee members. I like to acknowledge Prof. Kassem Mustapha for his invaluable guidance and feedback on my thesis research, and for accommodating me amidst his busy schedule. I am grateful to the faculty and colleagues at the Department of Petroleum Engineering Department for providing a very stimulating and knowledgeable environment, inspiration, and cooperation during the four and half years of my Ph.D.

I thank Dr. Hasan Al-Yousef for his help and guidance on various technical issues. Last but not the least, I am very grateful to my parents who have always given me unconditional love and emotional support. No words can express my gratitude for their love and confidence in me. I immensely value the blessings and help of my parents-in-law.

I wish to dedicate this work to my wife and son Mrs. Ayooluwa Obembe and Oluwa Tobiloba Obembe for their endless patience, prayers, love and encouragement that was a pillar of strength when it was most required.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| $A_{x/z}$ | Cross-sectional area of rock perpendicular to the flow of flowing fluid in x or z direction, $m^2$ |
| $API$ | American Petroleum Institute |
| $A_m$ | Matrix pore surface available per unit bulk volume, $m^{-1}$ |
| $a$ | Civan empirical constant in permeability and porosity correlation |
| $b$ | Civan empirical constant in permeability and porosity correlation |
| $B$ | Formation volume factor, $m^3/m^3$ |
| $B_{ob}$ | Oil formation volume factor at bubble point pressure, $m^3/m^3$ |
| $c$ | Civan empirical constant in permeability and porosity correlation |
| $C_p$ | Specific heat capacity, J/kg K |
| $c_w$ | Formation water compressibility of the systems, $kPa^{-1}$ |
| $c_o$ | oil compressibility coefficient, $kPa^{-1}$ |
| $c_t$ | Total compressibility of the systems, $kPa^{-1}$ |
| $c_s$ | Formation rock compressibility of the systems, $kPa^{-1}$ |
| $c_{\mu p}$ | Fractional change in viscosity per unit change of pressure, $kPa^{-1}$ |
| $c_{\mu T}$ | Fractional change in viscosity per unit change of temperature, $K^{-1}$ |
| $d_p$ | Pore throat diameter, m |
| $d$ | Core diameter, m |
| $D$ | Dispersion-diffusion coefficient, $m^2/sec$ |
| $Eq.$ | Equation. |
| $FZI$ | Flow zone indicator, m |
| $GL$ | Grunwald- Letnikov |
| $g$ | Gravitational acceleration, 9.8 $m/sec^2$ |
| $h_c$ | Film heat transfer coefficient, $W/m^2K$ |
| $K$ | Absolute variable permeability, $m^2$ |
| $L$ | Core length, m |
| $LTE$ | Local Thermal Equilibrium |
| $M$ | Average volumetric heat capacity of the fluid-saturated rock, $kJ/m^3$ K |
| $MB-$ | Memory-Based |
| $m$ | Cementation factor, dimensionless |
| $n$ | Empirical exponent |
| $NOLTE$ | No Local Thermal Equilibrium |
| $OB-$ | Oberbeck-Boussinesq |
| $p$ | Pressure of the system, kPa |
| $p_b$ | Oil bubble point pressure, kPa |
| $p_{init}$ | Initial pressure of the system, kPa |
| $q_{inj}$ | injection volume flow rate of steam, $m^3/sec$ |
| $Q^{"}$ | constant rate of heat generation per unit volume, $W/m^3K$ |
| $R_s$ | Gas oil ratio, $m^3/m^3$ |
| $r_{core}$ | Radius of core, m |

| | |
|---|---|
| $t$ | time, sec |
| $T$ | Temperature, K |
| $T_\infty$ | Ambient Temperature, K |
| $u$ | Darcy velocity, m/sec |
| $u_m$ | Fluid velocity with memory in porous media in x direction, m/sec |
| $U(t)$ | Overall heat transfer coefficient, W/m$^2$-K |
| $V$ | Volume, m$^3$ |

**Greek Symbols**

| | |
|---|---|
| $\alpha_L$ | Longitudinal thermal diffusivity of fluid-saturated porous medium, m$^2$/sec |
| $\beta$ | Expansion coefficient, K$^{-1}$ |
| $\lambda$ | Thermal conductivity, W/m-K |
| $\gamma_g$ | Gas specific gravity |
| $\gamma_o$ | Oil gravity |
| $\rho$ | density, kg/m$^3$ |
| $\Delta x$ | Size of grid block in x direction, m |
| $\Delta y$ | Size of grid block in y direction, m |
| $\Delta z$ | Size of grid block in z direction, m |
| $\phi$ | Porosity, fraction |
| $\Phi$ | Fluid potential, kPa |
| $\mu$ | fluid dynamic viscosity, kg/m-sec |
| $\eta$ | Ratio of the pseudo-permeability of the medium with memory to fluid viscosity, m$^3$ $sec^{2-\alpha}$/kg |

Subscripts

| | |
|---|---|
| $b$ | Bulk |
| $i$ | Cell counter |
| $e$ | Effective |
| $init$ | Initial condition |
| $inlet$ | Inlet condition |
| $ref$ | Reference condition |
| $s$ | Solid matrix |
| $w$ | Water |

Superscripts

| | |
|---|---|
| $n$ | Old time level |
| $n+1$ | New time level |
| $\upsilon$ | Old iteration level |
| $\upsilon+1$ | New iteration level |

# ABSTRACT

Full Name : [Abiola David Obembe]

Thesis Title : [A Modified Boussinesq-based approach to numerical investigation of thermal flooding process with experimental verification]

Major Field : [Petroleum Engineering]

Date of Degree : [May 2016]

Thermal recovery process involves the injection of heat energy into the reservoir. It relies on both momentum and heat transport mechanisms to induce oil displacement. The resulting elevation of reservoir temperature produces a nonlinear, transient temperature profile which is a key parameter in estimating the efficiency of the process. It also causes continuous alteration of the thermal characteristics of both reservoir rock and fluids that are seldom modeled in the heat and momentum transfer equations. A numerical and experimental study of Hot Fluid Injection (HFI) process in a porous medium was conducted, focusing on the underlying physical phenomena.

In this study, four numerical models describing a 1D HFI process in a cylindrical core was formulated taking into account the alterations of rock permeability and porosity, dispersion, and heat loss to the surrounding via forced convection. Two mathematical formulations each considering both the Local Thermal Equilibrium (LTE) and No Local Thermal Equilibrium (NOLTE) are proposed. The first approach employs a memory-based formalism (i.e. the use of fractional derivative concept) to relate the fluid volumetric flux to the fluid pressure/ potential. The second approach employs incorporating the Oberbeck-Boussinesq (OB) approximation to reduce the complexity of the flow problem. However, the effect of variation of rock physical property is considered which is usually not accounted for in the classic OB approximation approach. Therefore, a modified OB based mathematical model describing the HFI process using Darcy equation under LTE and NOLTE conditions was also formulated.

Through verification against the analytical solution of the simplified problem, and validation against published hot water flooding experimental data based on Berea sandstone the accuracy of the numerical simulators was established. Furthermore, numerical experiments were carried out to understand the effect of rock and fluid memory on the pore pressure, velocity, and temperature.

The results from the NOLTE numerical simulators reveal that the rock and fluid temperatures are almost identical and that the LTE formulation is sufficient for reservoir performance predictions.

# ملخص الرسالة

**الاسم الكامل:: أبيولا ديفيد أوبيمب**

**عنوان الرسالة: طريقة معدلة لنهج بوسينسق لاستقصاء عددي لعملية الغمر الحراري للمكمن مع التحقق التجريبي**

**التخصص: هندسة البترول**

**تاريخ الدرجة العلمية: شعبان 1437هـ**

تنطوي عملية الإنتعاش الحراري على ضخ الطاقة الحرارية إلى الخزان. ويعتمد ذلك على آليتي: قوة الدفع، ونقل الحرارة؛ للحث على إزاحة النفط. إن الإرتفاع الناتج في درجة حرارة الخزان ينتج منحنى ذاتي لدرجات حرارة غير خطية، عابرة والتي تعتبر معامل أساسي في تقدير كفاءة هذه العملية. كما أنه يسبب أيضا تغييرا مستمرا في الخصائص الحرارية في كل من صخور المكمن والسوائل التي نادرا ما تمثل في معادلات نقل الحرارة وقوة الدفع.
وقد أجريت دراسة رقمية وتجريبية لعملية حقن سائل ساخن (HFI) في وسط مسامي، مع التركيز على الظواهر الفيزيائية الكامنة.

في هذه الدراسة، لقد صممت أربعة نماذج رقمية والتي تصف عمليات حقن السوائل الساخنة، وإزاحتها (1D HFI) خلال عينة أسطوانية الشكل مع الأخذ بعين الإعتبار التعديلات في نفاذية الصخور ومساميتها، التشتت، وفقدان الحرارة إلى المناطق المحيطة عن طريق الحمل الحراري القسري.
لقد اقترحت صيغتين رياضيتين تعتبر كل منهما كلا الإتزان الحراري المحلي (LTE) والاتزان الحراري غير المحلي (NOLTE).
يوظف النهج الأول الشكلية القائمة على الذاكرة (أي استخدام مفهوم مبدأ الاشتقاق الجزئي) لربط التدفق الحجمي للسائل مع ضغط السائل / المحتمل.
أما النهج الثاني فإنه يوظف دمج تقريب أوبربك -بواسانسك (OB) للحد من تعقيدات مشكلة التدفق.
على أية حال، فقد أخذ في الإعتبار تأثير الاختلاف في الخواص الطبيعة للصخر، والتي عادة ما تكون غير محسوبة في النموذج الكلاسيكي لتقريب OB.
لذلك، فقد تمت تعديل تقريب OB على أساس نموذج رياضي يصف عملية حقن السوائل الساخنة HFI باستخدام معادلة دارسي (Darcy) في ظل الظروف التي وضعت لوصف صيغتي LTE وNOLTE.

لقد تأسست دقة المحاكي الرقمي من خلال التحقق من الحل التحليلي للمشكلة المبسطة، والتحقق من صحة البيانات التجريبية المنشورة لضخ الماء الساخن خلال الحجر الرملي بيريا.
وعلاوة على ذلك، فقد أجريت تجارب رقمية لفهم تأثير الصخر وذاكرة السائل على: ضغط المسام، السرعة، ودرجة الحرارة.
لقد كشفت النتائج من المحاكي الرقمي باستخدام NOLTE أن نوع الصخر ودرجة حرارة السائل تكاد تكون متطابقة، وأن صياغة LTE هي كافية لتوقعات أداء المكمن.

# CHAPTER 1

# INTRODUCTION

Currently, the Kingdom of Saudi Arabia has an approximated production capacity of 12.5 million barrels of oil per day, these huge reserves make the Kingdom a primary petroleum supplier both now and in the foreseeable future especially when forecasts for 2027 put the country's oil reserves and production rate at 262.3 billion barrels and 13.97 million barrels/day, respectively [1]. Nevertheless, as existing fields mature, meeting the increasing demand would require advanced technologies that boost the recovery factor from such fields.

EOR methods involve the injection of substances and/or energy into the oil reservoir to unlock trapped oil, improve sweep and enhance production rates. Among all the thermal EOR processes, steam injection is the most successful and widely used process that is applicable to a variety of reservoirs [2]. However, the success of a steam flood is entirely dependent on understanding the mechanism of heat transfer within the reservoir and the complex interactions between all temperature-sensitive rock and fluid parameters which govern the evolution of the temperature profile during the process [3–5]. A more accurate prediction of the temperature profile within a reservoir undergoing a thermal flooding process is a key factor in process design, production forecasting, and better reservoir management.

The classical constitutive equation describing fluid flow in porous media is the Darcy's equation which applies under conditions of isothermal laminar flow. However, when a heat energy is introduced, thermal alterations of rock and fluid properties induce non-Darcy flow effects which cannot be captured accurately by the Darcy equation. A less rigorous approach was introduced by Chan and Banerjee [6] and also Kaviany [7] that employed incorporating the OB approximation to couple the momentum equation with the energy equation. However, these approach has its pitfalls in that the approximation is only valid for flow problems where the variation of density is not significant.

It is, therefore, necessary to introduce new fluid flow models that incorporate such rock and fluid property alterations and still be amenable to solution. In the previous works by Hossain and co-authors [3–5], sets of mathematical model were developed in terms of a group of dimensionless numbers that correlate the varying rock and fluid properties using the "memory" concept. The "memory" concept is the incorporation of continuous time

functions in the model equations to account for alteration of rock and fluid properties. This approach has proven very effective in tackling the complex nature of the problem.

While the memory approach (MB Model) constitutes a step in the right direction, they remain to be proven by actual data. Since field data is seldom available in a form that affords direct model verification, and since experimental data are easier to generate under prescribed conditions, this research will endeavor to verify the memory models with available experimental data in the literature.

In addition, a simplified mathematical model based on the OB approximation is developed (subsequently referred to by OB Model). In this study, both the memory approach and the OB approach is investigated through numerical experiments considering both LTE and NOLTE. Furthermore, the less rigorous OB model was compared with the MB model to establish the region of validity of the former. Since the OB model is the simpler of the two, such comparison will also delineate the range of its applicability.

In conclusion, the memory-based approach presented herein is an alternative approach towards the mathematical modeling of thermal displacement processes in the subsurface.

# CHAPTER 2

# LITERATURE REVIEW

In this chapter, the literature review is divided into three sections which comprise of the following topics: Fractional derivatives and their applications in reservoir engineering, modelling of thermal displacement processes in porous media, and the OB approximation and its application in fluid flow problems in porous media.

## 2.1 Fractional derivatives and their applications in reservoir engineering

In this section, we present a detailed overview of the reported applications of fractional derivatives in the reservoir engineering literature. This section is comprised of four sub-sections each presenting an extensive overview of the reservoir engineering problem, and how the application of fractional derivatives leads to more accurate and reliable flow models.

### 2.1.1 Transient testing in fractured porous media

A large percentage of reservoir rocks in the world fall under the classification of naturally fractured or naturally fissured rocks with wide-spread applications in groundwater, hydro-thermal reservoirs, and petro-thermal reservoirs. By definition, a fractured rock is a porous medium that is intersected by a network of interconnected fractures, or flow channels [8]. Fractured rocks are usually recognized to be anisotropic and heterogeneous systems, and can be characterized by models that allow for the rock petro-physical properties (i.e. porosity and permeability) to vary rapidly and discontinuously over the reservoir domain. The literature reveals different approaches for describing transport through such systems. These include the double-porosity realization, dual porosity/permeability models, and the triple-porosity realization [9–23]. The double porosity realization for instance, considers the fractures (secondary continuum) and the matrix block (primary continuum) as two different but overlapping continua, interacting through the matrix-fracture interface [20]. The porosity and permeability are usually considered to be much greater in the fractures than in the matrix blocks. Therefore, the

macroscopic description of fluid flow in naturally fractured systems requires different orders of magnitude representative elementary volumes (REVs) [24].The challenge with modelling flow in these systems lies in the spatial heterogeneity of flow properties. Heterogeneous porous media comprise of various transport units each with different types and order of magnitude rates. A detailed overview of transport units and transport in heterogeneous porous media was presented by Civan (2010).

Pressure transient test offer the most direct technique to estimate permeability at the inter-well length scale [25]. The classic pressure transient models are formulated with the assumption of laterally non-heterogeneous reservoirs. Unfortunately, observations from core, log, and outcrop data prove otherwise. Thus the conventional approaches may not work in all situations. To improve these limitations, fractal models and fractional diffusion models have been proposed. The two mentioned approaches include the conventional homogenous reservoir solution as a special case. Discussion on the fractal models is beyond the scope of this review. Interested readers can refer to the excellent review article by Sahimi and Yortsos [26] and text by Hardy and Beier [25]. All fractional calculus formulations employ a non-local flux constitutive law to relate the volumetric flux to the pressure gradient. The non-local flux relationship implies that the observed/predicted flux is influenced by other factors other than the pressure gradient at the desired location at any instant in time.

Numerous studies exist in the literature towards the justification of such modifications [27–32]. Applications of the non-local flux constitutive equations can be found in the literature devoted to; fluid flow in porous media of fractal geometry, in naturally fractured unconventional shale reservoir, and nano-porous and porous materials [26,33–55]. In such systems, the classic diffusion approach based on random Brownian motion of the particles is inappropriate. The normal diffusion model(s) consider the mean square displacement of the diffusing particle to be a linear function of time [56]. More so, the normal diffusion models have been shown to be related to random walks, where only the previous location controls the subsequent particle location. In a nutshell, the classic random walk problems have a mean waiting time that is finite [38] .

A schematic of a naturally fractured and disordered porous media illustrated in **Figure 2.1**, the literature reveals employing an anomalous diffusion model may be more appropriate to describe fluid flow [36,57,58]. Equation 2.1 presents a more comprehensive relationship between the mean square variance and time [56,59]:

$$\sigma_r^2 \sim D t^{\gamma}, \quad \text{where} \begin{cases} \gamma = 1 \;\; Normal\; Diffusion \\ \gamma \neq 1 \;\; Anomalous\; Diffuison \\ \gamma > 1 \;\; Super\; Diffusion \\ \gamma < 1 \;\; Sub\; Diffusion \end{cases} \tag{2.1}$$

Where $\sigma_r^2$ is the variance and $\gamma$ is the fractional order of differentiation.

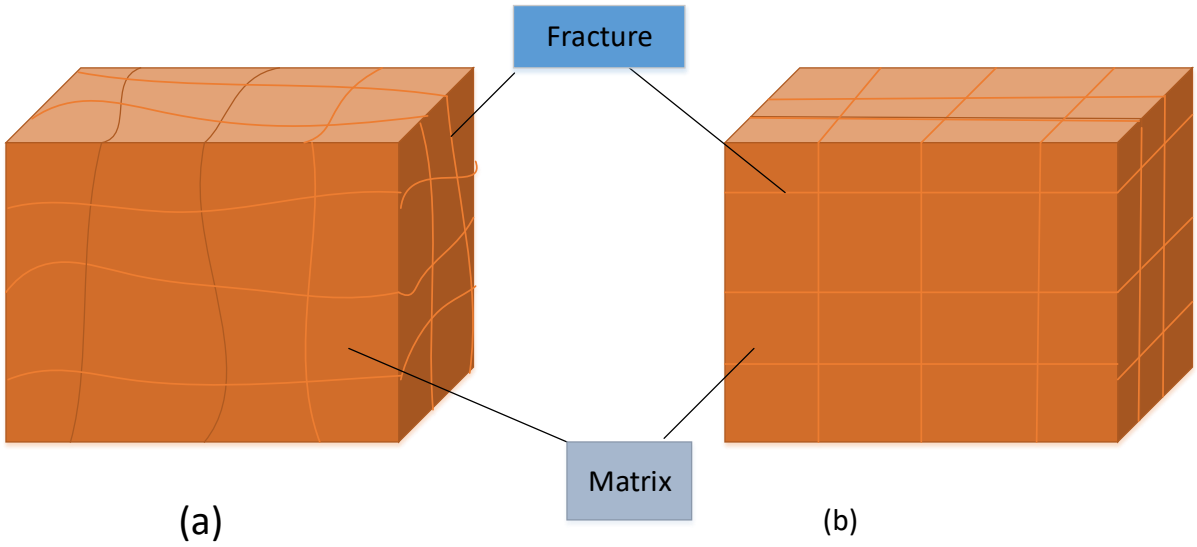


**Figure 2.1 Actual (a) and idealized (b) naturally fractured dual-porosity reservoir model (as described in [60]).**

Scenarios for which $\gamma = 1$ according to Eqn. (2.1) are termed as diffusive and the corresponding classical diffusivity models are appropriate. Implicitly, solutions to such equations are obtained based on the knowledge of the current state of the system. On the other hand, scenarios for which,$\gamma \neq 1$ require more information other than the current state of the system (i.e. the history of the process). Although, the dual-porosity idealizations [10–14,61–68], and triple porosity idealizations [69–72] have been employed in the literature with varying degree of success. A point what mentioning is that even such formulations are inadequate, in that there is a lack of a clear scale separation in unconventional reservoirs. In fact it was recently acknowledged that the dual- porosity idealization is a first order approximation for describing the transport network in shale reservoirs [73,74]. Fractional derivative concept have also been employed for pressure transient testing in scenarios where the geology is described as complex [75]. Thomas et al. [75] noted a pressure drop signature of the form

$$\Delta p \sim \Delta t^{\alpha} \tag{2.2}$$

Where $\alpha$ is a constant. Equation 2.2 implies a power law relationship as opposed to the classical signature which follows an exponential decay trend [76]. A power law behavior describes a slower rate of decay, with the contribution of the history of the process playing a huge role. Equations 2.3 and 2.4 predict the pressure distribution due to an instantaneous source for the normal diffusion equation, and for the CTRW model [38].

$$\hat{p}(\omega,t)\sim p(\omega,0)exp(-\eta\omega^2 t) \tag{2.3}$$

$$\hat{p}(\omega,t)\sim p(\omega,0)\ E_\gamma\left[-\left(\frac{t}{\eta\omega}\right)^\gamma\right] \tag{2.4}$$

Where $\omega$ is the Fourier mode, and $E_\gamma$ is the Mittag-Leffler function defined as:

$$E_\gamma(z)=\sum_{n=0}^{\infty}\frac{z^n}{\Gamma(1+\gamma n)} \tag{2.5}$$

Raghavan [38] concluded that due to the typical nature of the rock fabric in naturally occurring geologic media, transport occurs across disordered structures, rough interfaces, obstacles, cracks, traps, and crevices. He noted that employing fractional derivatives offer a natural way to capture such effects. Furthermore, he presented a flux law of the form

$$\vec{u}=-\lambda_\gamma\frac{d}{dt}\int_0^t dt'\ \frac{1}{(t-t')^{1-\gamma}}\nabla \mathrm{p}(\vec{x},t') \tag{2.6}$$

Where,$\lambda_\gamma = K_\gamma/\mu$, and $\gamma < 1$. For the special case where $\gamma = 1$, Eqn. (2.6) reduces to Darcy equation, implying $\lambda_{\gamma=1} = K/\mu$

A generalized constitutive non-local and temporal flux law describing anomalous diffusion in disordered fractured media is described by [77]:

$$\vec{u}=-\frac{K_{a,b}}{\mu}\frac{\partial^a}{\partial t^a}\left[\chi\frac{\partial^b p}{\partial x^b}-(1-\chi)\frac{\partial^b p}{\partial(-x)^b}\right] \tag{2.7}$$

where $0 < a, b \leq 1$, and $\chi(0 \leq \chi \leq 1)$ is the skewness parameter which controls the bias of the dispersion [78], and $K_{a,b}$ is a phenomenological coefficient with dimension [ $\mathrm{L}^2\mathrm{T}^a\mathrm{L}^{b-1}$].

The operators; $\frac{\partial^a}{\partial t^a}$, $\frac{\partial^b p}{\partial x^b}$ and $\frac{\partial^b p}{\partial(-x)^b}$ stand for the time fractional derivatives, left and right space fractional derivatives respectively. Equation (2.7) allows for diverging jump length variances ("long jumps") and unequal forward and backward jump probabilities at the particle scale. Till date, to the author's knowledge, the generalized constitutive flux law described by Eqn. (2.7) has not yet been employed however simplified variants of Eqn. (2.7) have been applied with varying degree of success in the literature. Accordingly, as pointed out by Holy and Ozkan [43], such a generalized flux constitutive equation allows for incorporation of the complexities associated with heterogeneous systems and other multi-scale flow mechanism. The main advantage of this approach is that it does not require the matrix and fracture petro-physical properties as one would require in the dual-porosity, or triple porosity models.

For sake of clarity, time fractional derivative order,$a$ accounts for presence of flow hindrances to flow, thus describing the degree of sub-diffusion. The space fractional derivative order,$b$ accounts for flow buffers i.e. super-diffusion. Although, Sprouse [79] pointed out that with a time fractional derivative order, $b > 1$, a super-diffusive transport can be simulated.

Therefore, incorporating Eqn. (2.7) into the continuity equation results in a generalized fractional diffusion equation for 1D linear flow of a slightly compressible fluid as follows:

$$\frac{\partial}{\partial x}\left\{\frac{K_{a,b}}{\mu}\frac{\partial^a}{\partial t^a}\left[\chi\frac{\partial^b p(x,t)}{\partial x^b}-(1-\chi)\frac{\partial^b p(x,t)}{\partial(-x)^b}\right]\right\}=\phi c_l\frac{\partial p(x,t)}{\partial t} \tag{2.8}$$

Where the different arguments for *a*, and *b* employed in the constitutive flux equations i.e. Eqn. (2.7) in the literature devoted to fractal porous media are presented in **Table 2.1**.

**Table 2.1 Non-local constitutive flux laws in the literature**

| Researchers Year | Time fractional order of differentiation | Space fractional order of differentiation | Skewness ($\chi$) | Interpretation of time fractional operator | Interpretation of space | Dimension of phenomenological coefficient |
|---|---|---|---|---|---|---|
| LeMhaut and Crepy (1983) | $\gamma-1$ where $\gamma=\frac{1}{d_f}$ | 1 | 1 | Not clear | Not applicable | $L^2T^{\gamma-1}$ |
| Le Mhaut (1984) | $\gamma-1$ where $\gamma=\frac{1}{d_f}$ | 1 | 1 | Not clear | Not applicable | $L^2T^{\gamma-1}$ |
| Nigmatullin (1984, 1986) | $\gamma-1$ where $\gamma=\frac{1}{d_f}$ | 1 | 1 | Not clear | Not applicable | $L^2T^{\gamma-1}$ |
| Compte and Jou (1996) | $\gamma-1$ where $\gamma=\frac{1}{d_f}$ | 1 | 1 | Not clear | Not applicable | $L^2T^{\gamma-1}$ |
| Caputo (1998, 1999) | $1-\gamma$ | 1 | 1 | Caputo sense | Not applicable | $L^2T^{1-\gamma}$ |
| Raghavan (2011) | $1-\gamma$ | 1 | 1 | Caputo sense | Not applicable | $L^2T^{1-\gamma}$ |

| | | | | | | |
|---|---|---|---|---|---|---|
| Raghavan and Chen (2013) | $1-\gamma$ | 1 | 1 | Caputo sense | Not applicable | $L^2T^{1-\gamma}$ |
| Ozcan *et al.* (2014) | $1-\gamma$ | 1 | 1 | Caputo sense | Not applicable | $L^2T^{1-\gamma}$ |
| Chen and Raghavan (2015) | $1-\gamma$ | 1 | $\beta$ | Caputo sense | Caputo sense | $L^2T^{1-\gamma}L^{\beta-1}$ |
| Raghavan and Chen (2016) | $1-\gamma$ | 1 | 1 | Caputo sense | Not applicable | $L^2T^{1-\gamma}$ |
| Holy and Ozkan (2016) | $1-\gamma$ | 1 | $\beta$ | Caputo sense | Caputo sense | $L^2T^{1-\gamma}L^{\beta-1}$ |
| Albinali and Ozkan (2016) | $1-\gamma$ | 1 | 1 | Caputo sense | Not applicable | $L^2T^{1-\gamma}$ |
| Awotunde *et al.* (2016) | $1-\gamma$ | 1 | 1 | Caputo sense | Not applicable | $L^2T^{1-\gamma}$ |

In conclusion, the fractional diffusion equations obtained by employing various forms of constitutive flux laws address the phenomena of long range dependence and/or trapping events much better than models governing classical diffusion [48].

### 2.1.2 Memory formalism in laboratory evolution experiments

The basic law describing the diffusion process of fluids in a porous medium is the well-known Darcy's law [80]. Many researchers have proposed different extensions to the classic Darcy's law by accounting for slip, inertia, non-Darcy flow etc. [81–87]. It has been reported in the literature [46,88,89] that under certain conditions the rock permeability may vary locally as a result of different factors. For instance, fluids can react chemically with the porous medium, solid particles embedded with the reservoir fluids may deposit or get attached along the pore throats, rock compaction, mineralization, precipitation, grain re-arrangement, fines migration, clay swelling and temperature variations within the porous media due to injection of hotter fluids. In such cases, Darcy equation may not be appropriate to describe fluid flow process because the permeability evolution is not known priori. Caputo and Plastino [90] pointed out that in the all above cases the variation in permeability depends on the amount of fluid passed through the matrix, which they termed a system memory. Therefore, authors introduced into the constitutive equation(s) derivatives of fractional order which weighs the past of the function. The main advantage of the memory formalism in describing flux of the fluid is that it allows the use of more

than two parameters instead of one parameter $K$ in the classic Darcy equation. The memory parameters employed are then estimate using experimental data. A flowchart illustrating the memory approach employed in some laboratory evolution experiments is presented in **Figure 2.2**.

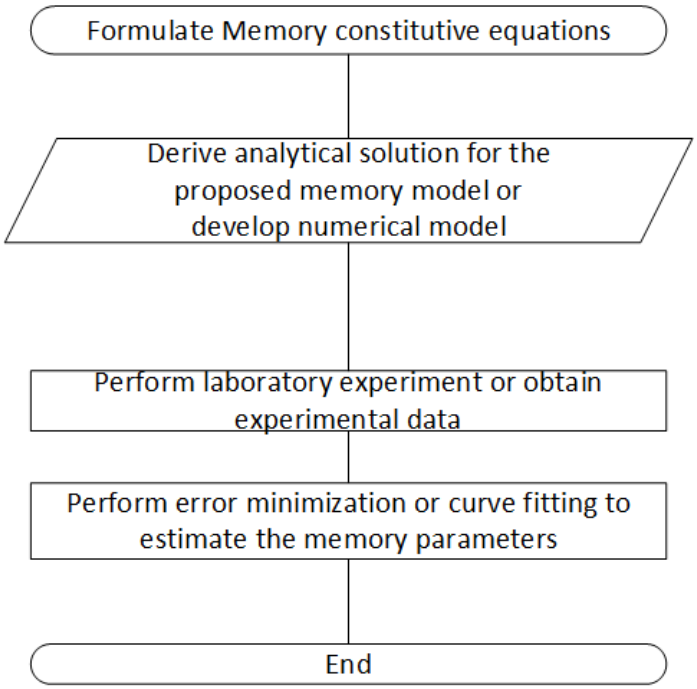


**Figure 2.2 Flowchart describing memory formalism for laboratory evolution experiments**

It is what stressing to the readers that the equations and parameters resulting from the memory formalism are phenomenological and must be calibrated against experimental data. The generalized memory formalism may be described by the following system of equations [90]:

$$\left(a + b\,\partial^{m_2}/_{\partial t^{m_2}}\right)p = \left(\alpha + \beta\,\partial^{m_2}/_{\partial t^{m_2}}\right)\{m(x,t) - m_0\} \tag{2.9}$$

$$\left(\gamma + \varepsilon\,\partial^{n_1}/_{\partial t^{n_1}}\right)q = -\left(c + d\,\partial^{n_2}/_{\partial t^{n_2}}\right)\nabla p \tag{2.10}$$

$$\nabla . q + \frac{\partial m}{\partial t} = 0 \tag{2.11}$$

Where

$$0 \le n_1 < 1,\ \ 0 \le n_2 < 1\ and\ 0 \le m_1, 0 \le m_2 < 1$$

From a practical point of view (i.e. to reduce the number of parameters) it is advisable to allow: $n_1 = n_2 = m_1 = m_2 = n$. The models in the literature on memory formalism in laboratory experiments are variants of Eqns. (2.9) to (2.11) and are presented herein.

Iaffaldano *et al.*[91] carried out experimental investigations to understand the permeability reduction observed during the diffusion of water in sand layers. Based on their results they concluded that the reduction in permeability observed was a result of grain rearrangement and compaction. Based on the data observed from five experiments with sand and water the average values of the memory parameters obtained are $n \approx 0.5$, and a pseudo-permeability ($\dot{k}$) of $10^{-8}$ $m^2 s^n$. It is worth stating that estimate of pseudo-permeability obtained is out of range of typical permeability of sand obtained from Darcy equation which is around $10^{-13}$ to $10^{-9}$ $m^2$ [84,92] which is not unusual due to the different dimensions of both variables.

Caputo [46,93,94] proposed a modified form of Darcy's law by introducing the time fractional derivative to account for the local permeability changes in the porous media. Subsequently he suggested a method to determine the two parameters defining his memory diffusion model.

$$q_i = -\eta \frac{\partial^\alpha}{\partial t^\alpha}\left(\frac{\partial p}{\partial x_i}\right) \tag{2.12}$$

Caputo and Plastino [95] proposed a modified constitutive relation in order to better describe the diffusion process of fluids in porous media. They proposed a space fractional derivative of pressure be introduced to Darcy's law. In this model the memory effect is introduced through the space fractional derivative, and its purpose is to capture the effect of the medium previously affected by the fluid. It is worth stating that the time-memory is suitable for accounting for local phenomena, and the space memory captures the variations in space.

$$q = \alpha_1 \frac{\partial^{1+\gamma}}{\partial x^{1+\gamma}} p + \alpha_2 \frac{\partial}{\partial x} p \tag{2.13}$$

Similar studies in the literature [90,96], were associated with modifications to the classic Darcy equation and one or more constitutive equation. Table 2.2 summarizes the above memory models presented over the years in the literature.

**Table 2.2 Memory models presented in the literature**

| Author & Year | Description | Equations |
|---|---|---|
| Di Giuseppe *et al.* [89] (2009) | Presented two modified constitutive equations by incorporating the time fractional operator acting on both the both the pressure gradient–flux and the pressure–density variations to describe the local variation of permeability in a porous medium. | $\left(a_1 + a_2 \frac{\partial^{\gamma_1}}{\partial t^{\gamma_1}}\right) q = -\left(a_3 + a_4 \frac{\partial^{\gamma_2}}{\partial t^{\gamma_2}}\right) \nabla p$<br><br>$\left(a_5 + a_6 \frac{\partial^{\gamma_3}}{\partial t^{\gamma_3}}\right) p = -\left(a_7 + a_8 \frac{\partial^{\gamma_4}}{\partial t^{\gamma_4}}\right)$ m |
| Iaffaldano et al., (2006) | Proposed a new diffusion model applicable to fluid transport porous media. | 1. Linear, isotropic, homogeneous porous medium.<br>2. Incompressible fluid.<br>3. The porosity of the media is not considered.<br>4. Transient flowing conditions. |
| Caputo and Plastino (2004) | Proposed a modification to Darcy's law and an additional constitutive relation.<br><br>Incorporated the Caputo fractional derivative to account for memory of the fluid and rock. | 1. Neglects the elastic reaction of the matrix since the equation of diffusion is uncoupled from the equation of elasticity.<br>2. Isotropic and homogeneous porous media.<br>3. Neglected inertia effects<br>4. The permeability is assumed to vary with time. |
| Caputo and Plastino (2003) | Proposed a modification to Darcy law by introducing a space fractional derivative to accurately describe | 1. Assumed the sand layers to be very thick.<br>2. Linear, isotropic, homogeneous porous medium.<br>3. Incompressible and viscous fluid.<br>4. The porosity of the media is not considered.<br>5. Permeability is assumed to vary with time. |

| | | |
|---|---|---|
| | the diffusion process in a porous media. | |
| Caputo (1999) | Proposed a modification to Darcy law through the Caputo fractional derivative to account for permeability reduction with time. | 1. Memory formalism to simulate permeability reduction with time in geothermal areas.<br>2. Linear, isotropic, homogeneous porous medium.<br>3. Incompressible and viscous fluid.<br>4. The porosity of the media is not considered. |

## 2.2 Numerical modelling of thermal displacement processes in porous media

The use of numerical reservoir simulators for steam flood or HFI performance prediction has been reported in the literature with applications dating to over 20 years. With rapid increasing numerical, simulation and computational capabilities, almost all important reservoir phenomena can be modeled adequately. Non-isothermal numerical models are similar to the conventional black-oil simulators with the additional modeling of the energy balance. That is, thermal effects are considered.

Numerical models have the advantage of encompassing all important physics in terms of accurate modeling of the temperature transients in a reservoir. However, the implementation of numerical Model requires proper understanding of the issues that are relevant and important. **Table 2.3** describes the amount of data required for the development of any numerical thermal model.

**Table 2.3 Typical data required by thermal reservoir simulators**

| Group | Property | Requirements |
|---|---|---|
| Reservoir | Principal values of the anisotropic absolute permeability and thermal conductivity, assigned to the directions x, y and z. | Three values of permeability and conductivity respectively for each block |
| | Porosity and heat capacity of reservoir rocks. | Two values respectively for each block |
| | Relative permeability for each phase | One relation for each phase at each grid block; each relation is a function of saturations and temperature |
| | Capillary pressure | Two relations as functions of saturations; several pairs allowed |
| | Reservoir geometry | Specify coordinate system to be used and locations of wells and boundaries |
| | Rock matrix compressibility | One value for each block |
| Overburden and Under-burden formations | Thermal conductivity and heat capacity | At least one of each for both Caprock and base rock |
| | Rock density | |
| Initialization values | Saturations, pressure, temperature, and composition | One value for each variable at each grid block |
| Fluid property | Density and viscosity of each phase; compressibility of the fluids | One relation for each phase; each relation should depend on temperature, pressure and possibly composition |
| | Component properties and K values (for compositional simulation) | Should be a function of pressures and temperature |
| | Latent heat of vaporization and saturation pressure | Latent heat of vaporization and pressure/temperature relation at saturation for each component that undergoes a phase change |
| | Enthalpy and internal energy of each phase | A relation for each quantity for each phase as a function of temperature, pressure, and possibly composition |
| Well and boundary conditions | Specify well type, and inner boundary conditions, Rates, pressures, and temperatures | Maximum and minimum values, constraints and penalties |

The earliest numerical model were developed for varying applications encompassing a large spectrum some of which are; one dimensional/two-phase flow and heat transfer neglecting the effect of heat losses [97], estimation of the recharge rate and the time of evolution for a fault charged hydrothermal system [98], economic analysis for comparing costs associated with different thermal recovery schemes [99], compositional steam flooding numerical Model [100,101], an equation of state thermal simulator [102], investigating multidimensional heat transfer problems associated with hot water or steam injection into an oil reservoir [103–106], heat flow in fractured carbonate reservoir [107–111], natural convection [112,113], understanding the effect of temperature dependent rock properties on three phase fluid flow during a steam flood [106,114–116], oil recovery correlation applicable for typical heavy-oil reservoirs [117,118], studies devoted to investigating the effect of steam distillation and solution gas during steam flooding [119] and application of steam injection for removal of non-aqueous phase liquids from subsurface [120,121]. However, a major drawback in the above studies was the simplistic assumptions incorporated into their numerical model. Take for instance; the injected fluid was considered to be non-condensable, temperature independent rock properties, convection only in one direction etc.

Hossain et *al.* [122,123] developed a one-dimensional numerical model to investigate the effects of the reservoir fluid and injection steam velocities on the temperature distribution in a one-dimensional reservoir. For the first case, they assumed that the reservoir rock and fluids had different temperatures. They observed little or no difference between the fluid and rock temperatures. Secondly, they considered when the reservoir rock and fluid temperature were equal. The authors solved the governing energy balance equation using an explicit finite difference scheme. The convective term and diffusive term were discretized using central differencing. Results showed that fluid velocity, initial steam injection rate, and time have strong effects on the temperature profile. However, in both cases the fluid velocity was assumed to be a linear function of time and was a function of rock and fluid properties.

Cicek [124] considered the steam displacement of oil in a naturally fractured reservoir by developing a three-dimensional, three phase, compositional, dual-porosity/dual-permeability model. The effects of capillary pressure, gravity were all incorporated into the numerical simulator. Cicek [125], again presented a detailed study on the effects of the reservoir and operational properties on the performance of steam displacement considering an inverted nine-spot pattern in a naturally fractured reservoir. In both studies, a fully implicit numerical scheme was developed. Subsequently, the Newton-Raphson method was employed to linearize the resulting sets of the equation.

Wu et *al.* [126] developed a model for predicting the breakthrough time for steam during steam injection into heavy oil reservoirs based on production performance data. However, the authors concluded that due to some features of the model, the model is best

applied during the early time period of steam drive applications and numerical simulations during the latter stages. Recent investigations have shown that temperature variations can lead to continuous alterations in rock and fluid properties [5,127,128]. This continuous alteration of fluid and pore space can be captured or modeled by fluid memory model especially in geothermal areas [46].

Again, Hossain et *al.* [129] developed a finite difference numerical model to investigate the permeability, porosity, and pore volume changes that occur during steam flooding process in a reservoir. The following assumptions were during their analysis; instantaneous thermal equilibrium between rock and fluid, the OB approximation was applicable. Their results showed that the reduction in permeability, increase in porosity and increase in pore volume during the steam injection process. They concluded that higher cumulative oil recovery would be predicted when the alterations of rock properties are included in recovery calculations. However, the authors assumed a constant fluid velocity for the energy balance.

Recently, new mathematical models have been proposed to describe the temperature evolution in a reservoir during steam injection process [3,4]. They included the effects of fluid memory through a modified Darcy law. The model was derived assuming a one-dimensional linear reservoir for both the case of instantaneous thermal equilibrium and unequal fluid and rock temperatures. Their study produced new dimensionless numbers that are specific to and influence the performance of a thermal process in an oil reservoir.

Civan [130] proposed an empirical model to describe the permeability impairment in porous rocks incorporating the contributions from fines deposition and non-isothermal conditions such as steam flooding or hot water injection. He developed a one-dimensional, finite difference, numerical scheme to predict the temperature distribution in a reservoir during non-isothermal conditions assuming thermal equilibrium between the flowing fluid system and the porous matrix. From the numerical results, it became evident that temperature variation had a significant effect on permeability impairment, with a higher degree of permeability impairment observed during non-isothermal conditions than isothermal conditions. The proposed model could easily be extended to two- or three-dimensional cases to account for the dispersion in various directions.

Yoshida et *al*.[131,132] developed a mathematical model capable of predicting the flow and temperature profile for a system comprising of horizontal wells intersected by transverse fractures. They assumed only single phase gas flow conditions. Sensitivity studies were conducted to understand the influence of fracture conductivity, and the fracture half-length on the temperature behavior of the system. They observed that the wellbore temperature was strongly affected by the fracture half-length and the fracture

conductivity. The proposed model is very useful for evaluating created fracture parameters with real-time post fracture temperature measurements.

Mozaffari et *al.*[133] had developed a three-dimensional, three phase simulator to investigate the steam injection process in heavy oil reservoir using a finite difference scheme. Although the proposed simulator was rigorous in that it accounted for three-phase relative permeability, capillary pressure, pressure and temperature dependent fluid properties, and interphase mass transfer between water and steam. Yet, the effect on temperature on rock properties (porosity, absolute permeability, and relative permeability) was not included, the oil was assumed to be non-volatile, and the hydrocarbon gas was considered insoluble in the liquid phases. The authors pointed out that steam injection could result in an overall recovery improvement of almost 60% from nothing for a fixed period of time.

Very recently, Irawan and Bathaee [134] developed a three phase mathematical model for the prediction of flow and temperature distribution for water alternating gas (WAG) process in a heterogeneous porous media. They included the effects of gravity, turbulence, relative permeability and capillary pressure. The flow model was developed in the cylindrical coordinate, with the flow in the tangential direction neglected. The governing equations were solved using implicit finite difference scheme. However, the model did not consider temperature-dependent relative permeability and changing rock properties.

**Tables 2.4 to 2.6** presents a summary of the comparison of the treatment of some rock and fluid properties, distribution of components in fluid phases, and features respectively in randomly selected non-isothermal simulators in literature.

**Table 2.4 Comparison of treatment of rock and fluid properties in some non-isothermal simulators presented in literature**

| Author | Rel. perm | Oil/ water viscosity | Gas viscosity | Oil/water density | K-values (*Kv*) | Porosity | Permeability |
|---|---|---|---|---|---|---|---|
| Spillete (1967) | $k_r(S)$ | $\mu_o(T)$ | $\mu_g(P,T)$ | $\rho_o(T)$ | NA[1] | Const. | Const. |
| Shutler (1969) | $k_r(S)$ | $\mu_o(P,T)$ | $\mu_g(P,T)$ | $\rho_o(P,T)$ | NA[3] | Const. | Const. |
| Shutler (1970) | $k_r(S)$ | $\mu_o(P,T)$ | $\mu_g(P,T)$ | $\rho_o(P,T)$ | NA[3] | Const. | Const. |
| Abdalla and Coats (1971) | $k_r(S)$ | $\mu_o(T)$ | $\mu_g(T)$ | $\rho_o(T)$ | NA[3] | $\phi(P)$ | Const. |
| Vinsome (1974) | $k_r(S)$ | $\mu_o(T)$ | $\mu_g(T)$ | $\rho_o(P,T,C)$ | NA[3] | Const. | Const. |
| Coats (1974) | $k_r(S,T)$ | $\mu_o(T)$ | $\mu_g(T)$ | $\rho_o(P,T,C)$ | NA[3] | $\phi(P)$ | Const. |
| Weinstein et al. (1977) | $k_r(S)$ | $\mu_o(T,C)$ | $\mu_g(T)$ | $\rho_o(P,T,C)$ | NA[3] | Const. | Const. |
| Coats (1978) | $k_r(S,T)$ | $\mu_o(T,C)$ | NC[2] | $\rho_o(P,T,C)$ | $Kv(P,T)$ | $\phi(P)$ | Const. |
| Abu-Kassem (1981) | $k_r(S,T)$ | $\mu_o(T,C)$ | $\mu_g(T,P)$ | $\rho_o(T)$ | $Kv(P,T)$ | Const. | Const. |
| Rubin and Buchanan (1985) | $k_r(S,T)$ | $\mu_o(T)$ | $\mu_g(T)$ | $\rho_o(T)$ | $Kv(P,T,C)$ | $\phi(C_c,\rho_c)$ | Const. |
| Ishimoto et al. (1987) | $k_r(S,T)$ | $\mu_o(T,C,P)$ | $\mu_g(T,P,C)$ | $\rho_o(T,P,C)$ | $Kv(P,T,C)$ | Const. | Const. |
| Sarathi (1990) | $k_r(S,T)$ | $\mu_o(T,C)$ | $\mu_g(T,P,C)$ | $\rho_o(T,P,C)$ | $Kv(P,T)$ | $\phi(P)$ | Const. |
| Jensen et al. (1992) | $k_r(S,T)$ | $\mu_o(T)$ | $\mu_g(T,C)$ | $\rho_O(T,C)$ | $Kv(P,T,C)$ | $\phi(P)$ | Const. |
| H. Class et al. (2002) | $k_r(S)$ | $\mu_o(T)$ | $\mu_g(T,C)$ | $\rho_O(T)$ | NA[3] | Const. | Const. |
| Cicek (2005a) | $k_r(S)$ | $\mu_o(T)$ | $\mu_g(T)$ | $\rho(T)$ | Yes | Const. | Const. |
| Hossain et al. (2008) | NA[3] | Const. | NA[3] | Const. | NA[3] | $\phi(T)$ | $K(T)$ |
| Hossain et al. (2009) | NA[3] | Const. | NA[3] | Const. | NA[3] | Const. | Const. |
| Agarwal (2009) | $k_r(S)$ | $\mu_o(T,P)$ | $\mu_g(P,T)$ | $\rho(P,T)$ | $Kv(P,T)$ | Const. | Const. |
| Rousset (2010) | NA[3] | $\mu_o(T)$ | NA[3] | $\rho(P,T)$ | NA[3] | Const. | Const. |

[1] Not applicable
[2]Not clear

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| App (2010) | NA[3] | $\mu_o(T,P)$ | NA[3] | $\rho(P,T)$ | NA[3] | $\phi(P,T)$ | Const. |
| Hossain et al. (2012) | NA[3] | $\mu_o(T)$ | NA[3] | Const. | NA[3] | $\phi(T)$ | $K(T)$ |
| Civan (2010) | NA[3] | $\mu_w(T)$ | NA[3] | Const. | NA[3] | $\phi(T)$ | $K(T)$ |
| Mozaffari et al. (2013) | $k_r(S)$ | $\mu_o(T)$ | $\mu_g(T)$ | $\rho(P,T)$ | NA[3] | $\phi(P)$ | Const. |
| Yoshida et al. (2014) | $k_r(S)$ | $\mu_o(T)$ | NA[3] | $\rho(P,T)$ | NA[3] | Const. | Const. |
| Irawan and Bathaee (2015) | $k_r(S)$ | NC[4] | NC[4] | $\rho(P,T)$ | NA[3] | $\phi(P)$ | Const. |

Table 2.5 Distribution of components in fluid phases in non-isothermal numerical simulators available in literature

| No | Component | Phases | | |
|---|---|---|---|---|
| | | Aqueous | Oleic | Vapor |
| 1 | Water | X | - | X |
| 2 | Light oil | - | X | X |
| 3 | Intermediate oil | - | X | X |
| 4 | Heavy oil | - | X | X |

Table 2.6 Major features in some available steam flood numerical simulators presented in literature

| Researcher | Steam distillation effect | Dimension of reservoir geometry | No of phases | Gravity override effect | No of components in phases | | Memory Effect | Capillary pressure effect |
|---|---|---|---|---|---|---|---|---|
| | | | | | Oil | Gas | | |
| Spillete (1967) | No | 2 | 2 | Yes | 1 | 0 | No | Yes |
| Shutler (1969) | No | 1 | 3 | Yes | 1 | 2 | No | Yes |
| Shutler (1970) | No | 2 | 3 | Yes | 1 | 2 | No | Yes |
| Abdalla and Coats (1971) | No | 2 | 3 | No | 1 | 1 | No | Yes |
| Vinsome (1974) | No | 3 | 3 | Yes | 1 | 2 | No | Yes |
| Coats (1974) | Yes | 3 | 3 | Yes | 3 | 3 | No | Yes |
| Coats (1976) | Yes | 3 | 3 | Yes | 2 | 2 | No | Yes |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Weinstein et al. (1977) | No | 1 | 3 | No | 2 | 2 | No | No |
| Coats (1978) | Yes | 3 | 3 | Yes | 2 | 2 | No | Yes |
| Abu-Kassem (1981) | Yes | 2 | 3 | Yes | 3 | 4 | No | Yes |
| Rubin and Buchanan (1985) | Yes | 2 | 4 | Yes | 2 | 4 | No | Yes |
| Ishimoto et al. (1987) | Yes | 1 | 3 | Yes | 3 | 3 | No | Yes |
| Sarathi (1991) | Yes | 2 | 3 | Yes | 3 | 4 | No | Yes |
| Jensen et al. (1992) | No | 2 | 3 | Yes | 1 | 1 | No | Yes |
| Cicek (2005a) | No | 3 | 3 | Yes | NA[3] | NA[5] | No | Yes |
| Hossain et al. (2008) | No | 1 | 1 | No | 1 | NA[5] | No | No |
| Hossain et al. (2009) | No | 1 | 3 | No | 1 | No | No | No |
| Civan (2010) | NA[5] | 1 | 3 | No | NA[5] | No | No | No |
| App (2010) | No | 1 | 2 | No | 1 | No | No | No |
| | | | | | | | | |
| Mozaffari et al. (2013) | No | 3 | 3 | Yes | 1 | 1 | No | Yes |
| Abu-Khasmin and Hossain (2015) | No | 1 | 3 | No | 1 | 1 | Yes | No |
| Lashgari et al. (2015) | No | 3 | 3 | Yes | NA[5] | | No | Yes |
| Irawan and Bathaee (2015) | NA[5] | 2 | 3 | Yes | NA[5] | | No | Yes |

[3] Not applicable

## 2.3 OB approximation in porous media

The OB approximation was first named after Boussinesq [135] , even though Oberbeck [136] was the first scientist who made use of it. However, current literature refers to it as the OB approximation. The OB approximation provides a form of simplification for investigating density-dependent flows and natural convection studies. In its complete form (strict form), the approximation includes the following four assumptions:
i) The variation of density due to either temperature and or solute concentration changes be neglected in all equations, except with the gravity term in the equation of motion, ii) constant material and fluid properties are assumed constant, iii) the effect of viscous dissipation is assumed negligible, and iv) a linear equation of state is considered throughout.
Boussinesq flows have been observed to occur naturally, such as oceanic circulations and atmospheric fronts. The OB approximation is extremely accurate for such flows and makes the mathematics and physics much simpler and transparent.

There are numerous versions of the OB approximation (strict, relaxed) reported in the literature. For instance, the Gartling and Hickox [137] "Relaxed"(extended) OB approximation where other fluid properties except the density were allowed to vary. Nevertheless, it must be mentioned that most scientists agree that the OB assumption in any of its form is a valid simplification for studies with negligible density variations. However, a major drawback with the OB approximation is that, in the limit of density differences approaching to zero, the conservation equations do not reduce to those used in the OB approximation [138]. Mc Doughall et *al*. [139] emphasized that the OB approximation makes three major assumptions, rather than two, as previously thought. In addition, they showed the huge errors could result from the use of a divergence free velocity to advect a tracer.

Generally, free convection arises due to the density reduction that occurs when a fluid is heated. This, in turn, causes the fluid to rise. Therefore, it is expected that any mathematical description of flow must take this feature into account. If the temperature difference is taken to be small ($\Delta T$), it is expected that the density changes in the system should be of like magnitude. If the density is expanded in a Taylor's series about an average temperature ($\bar{T}$)

$$\rho = \rho\big]_{T=\vec{T}} + \frac{d\rho}{dT}\Big]_{T=\vec{T}} \left(T - \vec{T}\right) + \cdots . higher\ order\ terms \tag{2.14}$$

$$\rho(T) = \vec{\rho} - \vec{\rho}\vec{\beta}\left(T - \vec{T}\right) \tag{2.15}$$

The terms $\vec{\rho}$ and $\vec{\beta}$ are density and volumetric expansion coefficients respectively, evaluated at the average temperature. The above terms are defined as:

$$\beta = \frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_p = \frac{1}{\left({}^1/_\rho\right)}\left(\frac{\partial\left({}^1/_\rho\right)}{\partial T}\right)_p = -\frac{1}{\rho} \tag{2.16}$$

The buoyant force is introduced by inserting the equation of state into any of the equations of motion i.e. Incorporating Eq. (2.15) (equation of state) leads to OB approximation. Over the next few pages, we will review several works devoted to of the OB approximation model in porous media.

### 2.3.1 Numerical Validation of the OB approximation in porous media

Interestingly, numerous numerical studies devoted to validating the OB approximation for natural convection process in a porous media are available. The most famous is the study by Peirotti *et al.* [140], the authors considered a fluid-saturated, porous cavity with the vertical walls maintained at two different temperatures and horizontal walls completely insulated. They considering when the saturating fluid was both water and air over a wide range of Rayleigh numbers and aspect ratios. The most important conclusions from their study was they observed that Nusselt number obtained with the OB approximation was most times different from the Nusselt number without the OB approximation.

Marpu and Satyamurty [141] studied the validity of the OB approximation based on a numerical model describing the free convective heat transfer in a fluid saturated porous annulus. They observed significant differences in the Nusselt number when the density variation was considered in all the governing equations for a gas-filled porous annuli. Hence, they concluded that the OB approximation may not be valid when modelling free convection heat transfer in gas-filled porous media particularly at low (cryogenic) temperature levels.

Johannsen [142] investigated the validity of the OB approximation numerically considering the Elder problem as a case study. The bifurcation diagram with respect to the Rayleigh number was investigated on a hierarchy of uniformly refined grids. He concluded that despite the similarity in solutions, the corresponding bifurcation diagrams were shown to be topologically not equivalent.

Based on the uncertainty accompanying the OB approximation [138,143] derived the appropriate limits for which the gravity term in the momentum equation should be retained, while neglecting the other density effects in the heat and solute equations in

porous media. Of importance, was the step by step derivation of OB equations describing the flow of a fluids in a porous medium based on Darcy's equation [144]. In addition, they obtained similar type OB equations based on the Forchheimer and Brinkman equations. There are also numerous studies which observed that the OB approximation was inappropriate, especially when the body force, fluid volumetric expansion coefficient and the temperature differences greatly exceed certain limits. Interested readers can refer to [145–151].

### 2.3.2 Applications of OB approximation in porous media flow problems

Rudraiah [152] considered numerically the effect of Darcy and viscous resistance terms assuming a fully developed natural convection fluid flow problem incorporating the OB approximation employing the Brinkman equation [153]. They assumed the two vertical plates were maintained at the same temperature with heat being generated in situ by both viscous and Darcy dissipations. They noted that when the contributions of Darcy and viscous dissipations terms in the energy equation are negligible, the energy and momentum equations can be decoupled. However, when both viscous and Darcy resistances terms are significant, the momentum and energy equations are coupled leading to a nonlinear equation.

Again, Rudraiah [154] considered the natural convection process in a vertical porous stratum based by incorporating the OB approximation into the Brinkman equation. Analytical solutions were obtained through a perturbation method with some restriction on the buoyancy parameter '*N*'. Furthermore, a numerical solution was presented using the Runge-Kuta-Gill method. An interesting result of their study was that they showed their analytical solutions were valid for *N* values less than one. For a complete review refer to original manuscript.

Prasad and Kulacki [155] developed a numerical model using finite-difference methods to model the natural convection process in a rectangular porous cavity. The effect of the aspect ratio was summarized by a family of curves, for aspect ratios ranging from 0.05 to 100. It was found that for an aspect ratio less than one, multi-celluar flow develops.

Hadjisophocleous and Sousa [156] developed a three-dimensional numerical model to investigate the free convection process incorporating the OB approximation. They assumed laminar and steady state flow of a high-Prandtl-number fluid heated in a cylindrical tank by line heat sources. For their numerical formulation, a staggered-mesh geometry, as well as skewed time-like marching procedure was implemented, and a multi-grid method was used for the discretized pressure correction equation. In addition they

proposed a variation of the SIMPLEC algorithm proposed by [157] to solve the governing equations.

Oosthuizen [158] likewise developed a numerical model based on Darcy law to investigate flow field over a horizontal plate in a saturated porous media. The two-dimensional problem was discretized by finite element method. He observed that the heat transferred from the plate was strongly dependent on the plate dimensionless depth and the strength of buoyancy forces.

Umavathi and Malashetty [159], derived an analytical solution and numerical model to understand flow and heat transfer characteristics during the natural convection process of a couple-stress fluid in a vertical porous stratum assuming a non-Newtonian fluid. The perturbation method of solution was used to obtain analytical solutions and a finite difference (Central differencing) technique was used for numerical solutions.

Khanafer and Chamkha [160] derived using a volume-averaging method the governing equations for predicting flow behavior in a fluid saturated porous medium in the presence of an internal heat generator assuming transient, laminar, and mixed convection flow. Furthermore, they incorporated the OB approximation to account for the effect of buoyancy. In addition, they developed a numerical model based on the Brinkman-extended Darcy equation of motion to understand how some non-dimensionless numbers contribute to the mixed convection flow in a square enclosure filled with a fluid-saturated porous medium.

Furthermore, Hossain and Wilson [161] investigated numerically the natural convection process in a fluid saturated porous medium immersed in a rectangular enclosure surrounded by non-isothermal walls. They assumed transient, laminar fluid flow conditions as well as the presence of internal heat generator. The OB approximation was used to handle the buoyancy effects. The authors investigated the effects of heat generation and the porosity of the medium on the streamlines, isotherms as well as on the rate of heat transfer from the walls of the enclosure.

Gaikwad and Kamble [162] developed a numerical model to predict the flow behavior of a non-Newtonian fluid in a porous medium based on a modified form of Darcy's law. They were interested in understanding the influence of Soret parameter. In their analysis, they assumed that the OB approximation to be valid, and all other fluid properties were assumed constant.

Other numerous investigations aimed at understanding natural convection in a porous medium are available in literature for example; some researchers investigated problems for which the porous medium was adjacent to heated bodies in the form of a flat plate [163,164], a vertical cylinder [165,166], a cone [167–169], along a wavy surface [170,171]. In many of these articles, the flow equation is governed by Darcy's law and the boundary

layer approximation is applicable. Therefore, most of the solutions are valid at high Rayleigh numbers and low Reynolds numbers. Similarly, other conservation studies related to natural convection for non-Newtonian fluids based on the OB approximation can be found in references [172–177].

# CHAPTER 3

# STATEMENT OF PROBLEM

Hossain et *al.* [3,4] pointed out that a majority of available models for fluid flow in porous media are unable to handle the continuous alteration of rock and fluid properties with time during a thermal operation accurately. In this regard, a new mathematical model based on the memory- formalism were formulated by Hossain et *al.* [3,4]. While these models do encompass all aspects of the phenomenon, only a few have been solved successfully. Even for those solved numerically, their validation has not been established experimentally. In this research, we tackle this challenge and provide an approximate solution to the extended memory-based partial differential equations. Due to lack of neither precise experimental data nor suitable field data, the MB models have not been verified. Therefore, an experimental investigation would be necessary to establish the validity of MB model during a thermal flooding process.

In addition, as mentioned earlier, the classical models for thermal EOR processes are based on the assumption that fluid flow is governed by Darcy's law. Such assumption is known to produce results that could vary considerably from actual flow behavior, especially under thermal unsteady state conditions. The OB approximation was used to tackle this deficiency by incorporating its concept in momentum and energy balance equations to produce a flow model which is applicable in thermal EOR processes. The OB model assumes the fluid density is uniform except for the body force term in the momentum equation along the direction of gravity. As reported in the literature review, the OB approximation provides improved convergence for many natural convection flows than by using fluid density as a function of temperature. The advantage of the OB model is that the constant density assumption reduces nonlinearity in the governing differential equation. However, the approximation is suitable to flow problems where the density variations are small. Simulations for different model parameters as well as initial and boundary conditions will be run to carry out further parametric studies.

This research proposes to modify the aforementioned OB approximation to suit thermal flooding operations and solve it using an existing or if necessary a newly formulated computationally efficient scheme/algorithm. Numerical would then establish the range of applicability of the proposed model by comparing the predicted temperature with the temperature predicted with the fully compressible flow formulation.

This proposed study will establish the MB models as essential for more accurate prediction of flood development, which would ultimately lead to improved thermal process design and better forecasting of reservoir performance.

## 3.1 Knowledge Gap

While existing reservoir simulators can model a thermal operation, the accuracy of such simulators cannot be assessed with certainty due to the simplifying assumptions built in the flow model(s). Hossain et *al.* [3,4] pointed out that unless such models tackle the complex rheological and thermal alterations in the reservoir's rock and fluid as a result of the non-isothermal conditions during thermal floods, significant errors in recovery factors could go undetected.

As mentioned earlier while MB formulations have a long history in fluid flow problems, their actual validity has not been proved yet in the area. This study will attempt to fulfill this gap by solving numerically the developed memory-based equation and the modified-OB model and validating the results against analytically or with experimental data. The experimental data has to be generated in a physical laboratory model that simulates a thermal flood in a water reservoir. The experiments will have to be designed to provide detailed and accurate temperature measurements along the length of the porous medium with time.

## 3.2 Objectives

The main objectives of this research are the following:

1. To develop a mathematical model by modifying the existing OB-based model to suit the thermal flooding process while modifying the Hossain et *al.* memory model to suit the HFI process in a cylindrical core under LTE and NOLTE conditions
2. To solve the model equations developed in objective 1 numerically using an appropriate numerical scheme.
3. To verify the MB model and the modified OB models analytically or with experimental data.

## 3.3 Research Methodology

In this research, two different approaches were employed namely: numerical modeling and analytical/experimental verification. In the numerical modeling section, two mathematical models were developed to predict the temperature profile during a hot fluid injection process into a saturated core. A summary of both approaches is discussed in the remainder of this chapter.

### 3.3.1 Mathematical Formulation

The memory formalism proposed by Hossain et *al.* [3,4,178] was modified to describe a hot water core flood. In addition, a less rigorous model, based on the OB approximation for the momentum/energy balance equations was developed by coupling the mass, momentum and energy equations. In all analysis, a porous medium with the uniform cross-sectional area is considered with a constant rate of heat generation per unit volume maintained.

### 3.3.2 Numerical Modelling

Both model equations were solved numerically through an iterative numerical scheme that could handle the nonlinear nature of the problem. All computations were performed in MATLAB language. The OB models were compared with the MB models to establish a benchmark and determine the region of validity of the former.

### 3.3.3 Simulator Verification/Validation

The verification phase of the research is important to establish the accuracy and reliability of the developed numerical simulators. Unfortunately, due to the non-linear nature of the problem considered analytical solution can only be derived for the simplified case of constant physical properties and an implicit memory exponent of one. Thus, the developed numerical simulators are first verified against the analytical solution of the simplified problem. Subsequently the published experimental data by Arihara [179] was employed to validate the simulator using the Hot water Berea sandstone 2 data. This research was accomplished throughout four phases as illustrated in the flow chart below.

Details about each phase are discussed in **Table 3.1** below.

**Table 3.3 Mapping of phases and tasks to achieve objectives**

| Objective | Tasks |
|---|---|
| 1) Develop a mathematical model by modifying the existing OB-based model and extend/modify Hossain et *al.* memory model | 1. Develop new set of mass, flow, and energy equations based on the OB approximation.<br>2. Extend the memory based Hossain et *al.* model to suit a HFI process in a core plug under LTE an NOLTE conditions. |
| 2) To solve the model equation developed in objective 1 numerically using an appropriate numerical scheme. | 1. Discretize the developed OB model and Hossain et *al.* model using suitable numerical scheme.<br>2. Develop a MATLAB code for the above discretized equations.<br>3. Find an efficient numerical scheme to solve the proposed model.<br>4. Perform computations with different porous medium properties and different boundary conditions.<br>5. Visualize the results |
| 3) To verify the modified OB-based model and the Hossain et *al.* MB model equations analytically or with published experimental data. | 1. Verfication/ validation with data in the literature. |

# CHAPTER 4

# DEVELOPMENT OF MATHEMATICAL MODELS

In this chapter, two mathematical models that describe the HFI process into a cylindrical core are under the assumption of LTE and NOLTE conditions. The model Equations are based on one-dimensional, singe-phase flow and can account for the alteration of rock permeability and porosity with time. The proposed partial differential Equations were discretized into its algebraic form in order to be solved numerically using any suitable numerical scheme. It must be emphasized, that the choice of a one-dimensional model stems from the fact that injection and production ends of the end plugs employed in core flood experiments are designed such that end effects are minimized thus, only linear flow is realized using the end-stem [180]. In fact, the design of such end-stems result only in linear flow of fluid along the axis of the core.

## 4.1 Model Assumptions

The following assumptions are used in developing this model.

1. The flow is linear and one-dimensional with gravity neglected.
2. Thermal equilibrium is instantly reached within the grid concerned only for the cases of LTE.
3. Radiative heat transfer is assumed to be absent.
4. Heat loss through the insulation is accounted for through an overall heat transfer coefficient.
5. The contribution of pressure work, and the work done by viscous forces are negligible.
6. The initial pressure and temperature are assumed uniform throughout the length of the core.
7. The rock and fluid properties (porosity, permeability, density, viscosity, and thermal conductivity) are considered to vary with either pressure or temperature or both.
8. Lastly, the compressibility coefficients of water and rock are considered as constants.

## 4.2 Local Thermal Equilibrium Mathematical Models

The memory formalism was proposed to overcome the many limitations associated with the classic Darcy equation. For instance, in Darcy's equation, flow occurs by influence of the immediate surroundings. That is, the predicted volumetric flux is solely determined by the magnitude of the pressure gradient and permeability at the location concerned. On the other hand, the memory based model proposed herein improves upon this notion. Accordingly, the memory approach considers the flow to be as a result of not only the instantaneous pressure at a certain location but also on its history.

### 4.2.1 Memory Based Formulation

The governing Equations which describe the temperature distribution in the cylindrical core under LTE consists of the continuity Equation, a constitutive Equation, relating the fluid volumetric flux to pressure, an Equation of state, and the conservation of energy.

Fluid Mass Balance

Mass conservation for the water phase is expressed by the following partial differential Equation.

$$-\nabla.[\rho_w(p,T)\vec{u}_m] + \rho_w(p,T)q_w = \frac{\partial}{\partial t}[\rho_w(p,T)\phi(p,T)] \tag{4.1}$$

A generalized Darcy's equation is proposed to describe the notion of fluid and rock memory. For a one-dimensional linear horizontal system the constitutive Equation can be written as [127];

$$u_m = -\eta D_t^{1-\gamma}\left(\frac{\partial p}{\partial x}\right) \tag{4.2}$$

The fractional derivative operator, $D_t^{1-\gamma}$ introduced in Eq. (4.2) , is interpreted through the Grunwald–Letnikov (GL) formula defined by [181] as:

$$D_t^{1-\gamma} f(t) = \lim_{\Delta t \to 0} (\Delta t)^{\gamma-1} \sum_{k=0}^{t/\Delta t} \frac{(-1)^k \Gamma(2-\gamma)}{k!\Gamma(2-\gamma-k)} f(t-k\Delta t), \text{ for } 0 < \gamma \leq 1 \tag{4.3}$$

Where $\Gamma(.)$, is the standard Gamma function.

From Eq. (4.3), the value of the GL time fractional derivative requires the history of previous values from k = 0 up to k= $t/\Delta t$. In this thesis we follow the definition of the composite variable $\eta$ presented in [127] and also re-presented in Eq. (4.4).

$$\eta(p,T) = \frac{K_{app}}{\mu(P,T)} = \frac{K(P,T)}{\mu(P,T)} t^{1-\gamma} \tag{4.4}$$

Numerous other generalized Darcy's equation applicable to describe macroscopic flow of fluid in any porous media have been reported in literature using the memory formalism [39,88,90,182–184]. However, in all literature devoted to fluid flow in porous medium the time fractional derivative operator is interpreted in the Caputo sense.

Substituting Eq. (4.2) into Eq. (4.1) as a constitutive Equation to describe fluid flow in the core plug, results in:

$$\nabla.\left[\rho_w \eta D_t^{1-\gamma}\left(\frac{\partial p}{\partial x}\right)\right] + \rho_w q_w = \frac{\partial}{\partial t}(\rho_w \phi) \tag{4.5}$$

The formation volume factor $B_w$ is introduced to relate the density at reference condition to density at any pressure and temperature as expressed below.

$$B_w = \frac{\rho_{w,ref}}{\rho_w} \tag{4.6}$$

Introducing Eq. (4.6) into Eq. (4.5) results in:

$$\nabla.\left[\frac{\eta}{B_w} D_t^{1-\gamma}\left(\frac{\partial p}{\partial x}\right)\right] + q_{w,sc} = \frac{\partial}{\partial t}\left(\frac{\phi}{B_w}\right) \tag{4.7}$$

Noting that,

$$q_{w,sc} = \frac{q_w}{B_w} \tag{4.8}$$

For convince sake, the functional dependency of rock and fluid properties has been omitted so far.

Conservation of Energy

The development of the mathematical relationship describing the heat flow in a core plug is a combination of the heat balance, Fourier equation, and a generalized and Darcy's equation. Considering a representative volume of the core element shown in Figure 4.1, the heat balance equation results in [7,130,185,186];

$$-q_{loss} + \frac{\partial}{\partial x}\left(\{\lambda_e + \alpha_L \rho_w C_{pw}\}\frac{\partial T}{\partial x}\right) - \frac{\partial}{\partial x}\left(u \rho_w C_{pw} T\right) = \frac{\partial}{\partial t}[(\rho C_P)_b T] \tag{4.9}$$

Where,

$$(\rho C_P)_b = \phi \rho_w C_{Pw} + (1-\phi)\rho_R C_{pR} \tag{4.10}$$

$$\lambda_e = \lambda_w \phi + (1-\phi)\lambda_R \tag{4.11}$$

Where, $\alpha_L$, and $\lambda_e$ refers to the longitudinal dispersivity, and effective thermal conductivity. Refer to Nomenclature for definition of other symbols presented above.

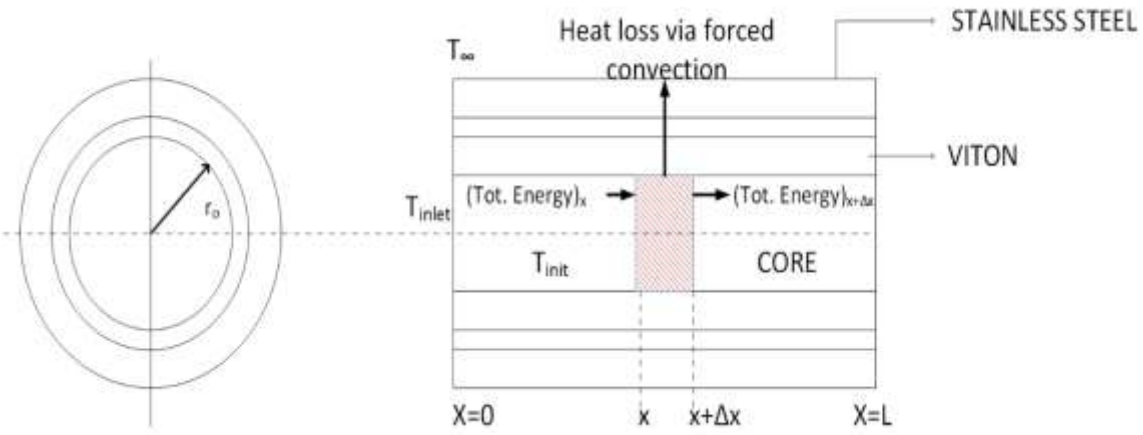


**Figure 4.1 Heat balance of a core element to derive the energy equation**

In this work, the heat transfer from the core to the surroundings was expressed by forced convection, rather than conduction. Generally, physical model such as the experimental setup are made with limited insulation thickness. Thus, a representation of the heat loss in terms of an average over-all thermal coefficient determined in the laboratory seems justified. Eq. (4.9) has been employed to approximate the heat loss from the cylindrical core holder [118,187].

$$q_{loss,i} = \frac{d\pi\Delta x U(T_i - T_\infty)}{V_b} \tag{4.12}$$

Refer to Nomenclature for definition of terms defined above.

### 4.2.2 Physical properties of injected and starting fluid for memory based models

1. Density:

The mass density of water is considered to be a function of temperature and pressure and calculated using the following correlation [188]:

$$\rho_w(T) = 236.372 - 1.29167\mathrm{T} + 3.78125 \times 10^{-3}T^2 - 5.40258 \times 10^{-6}T^3 + 3.74277 \times 10^{-9}T^4 - 1.01916 \times 10^{-12}T^5 \quad (4.13)$$

$$\rho_w(p,T) = \rho_w(T)\left[1 + c_w\left(p - p_{ref}\right)\right] \quad (4.14)$$

Where T is in ° $R$, and density in lbm/cu.ft.

2. Viscosity

Water viscosity is assumed to be dependent of pressure and temperature and is described by the following correlation [137]:

$$\mu_w(T) = 159.5T^{-1.182} \quad (4.15)$$

$$\mu_w(p,T) = \mu_w(T)\left[1 - c_{\mu p}\left(p - p_{ref}\right)\right] \quad (4.16)$$

Where T is in °F, $\mu_w$ in cp, p, and $c_{\mu p}$ are in any consistent set of units.

The functional dependences of physical properties are given in table 4.1.

**Table 4.1 Functional dependence of rock and fluid properties for MB- numerical simulator**

| Variable | Definition | Functional dependence |
|---|---|---|
| $\phi$ | Porosity | $p, T$ |
| $\lambda_w$ | Water thermal conductivity | $p, T$ |
| $\mu_w$ | Water viscosity | $p, T$ |
| $\rho_w$ | Water density | $p, T$ |
| $K$ | Permeability | $T$ |

### 4.2.3 Modified Oberbeck-Boussinesq Mathematical Formulation

In this section, we propose a modified OB approximation where the variation of rock properties (porosity and permeability) are considered. Similar to the classic form all other fluid properties are considered constant under initial pressure and temperature conditions. However, the viscosity and density of water is allowed to vary with pressure and temperature in the simulator.

The major difference between both models is the constitutive equation describing fluid flow in porous media. Therefore, following the same approach as earlier, i.e. combining the continuity equation, a constitutive equation relating the fluid volumetric flux to pressure (Darcy equation), an equation of state, and the conservation of energy for the rock matrix and the fluids respectively.

Fluid Mass Balance

Mass conservation for the water phase is expressed similar to Eq. (4.1) with the memory based constitutive equation. The term ($u_{\mathrm{m}}$) is considered as the fluid superficial velocity $(u)$ described by Darcy equation. Therefore, the proposed constitutive Equation for the modified OB model is presented in Eq. (4.13).

$$\vec{u} = -\frac{K(T)}{\mu(p,T)}\nabla p \tag{4.17}$$

Similar to the approach earlier, combining Eq. (4.13) into Eq. (4.1) as a constitutive equation to describe fluid flow in the core plug results in:

$$\nabla.\left[\frac{K(T)}{B_w(p,T)\mu(p,T)}\nabla p\right] + q_{w,sc} = \frac{\partial}{\partial t}\left(\frac{\phi(p,T)}{B_w(p,T)}\right) \tag{4.18}$$

Conservation of Energy

The temperature distribution along the core length for the modified OB model under LTE is described by Eq. (4.9) presented earlier. The key difference between the memory formulation and OB formulation lies in the constitutive equation employed in the fluid mass balance, as well as the dependency of the thermal conductivity, and specific heat capacity of the fluid phase.

### 4.2.4 Physical Properties of Injected and saturating fluid

1. Fluid density

For the Boussinesq model the following simplified correlation is employed [137]

$$\rho_w(p, T_w) = \rho_{w,ref}\left[1 + c_w\left(p - p_{ref}\right) - \beta_w\left(T_w - T_{ref}\right)\right] \tag{4.19}$$

Where,

$$c_w = -\frac{1}{V_w}\left(\frac{\partial V_w}{\partial p}\right)_T \tag{4.20}$$

$$\beta_w = \frac{1}{V_w}\left(\frac{\partial V_w}{\partial T_w}\right)_p \tag{4.21}$$

Eq. (4.21) is applicable with any consistent set of units.

2. Viscosity

For the Boussinesq model the following simplified correlation is employed [137]

$$\mu_w(p, T_w) = \mu_{w,ref}\left[1 + c_{\mu p}\left(p - p_{ref}\right) + c_{\mu T}\left(T_w - T_{ref}\right)\right] \tag{4.22}$$

Eq. (4.22) is applicable for any consistent set of units.

## 4.3 No Local Thermal Equilibrium Models

### 4.3.1 Memory Based Formulation

In this methodology the simplifying assumption of equal rock and fluid temperature i.e. LTE is invalid. The energy transport in both phases account for the convective, and diffusive transport of energy. In most studies the assumption of equal fluid and rock temperature is usually employed. However, there are certain scenarios such as highly transient problems and some steady state problems where such approximation fail [185]. This kind of situation was referred to as thermal non-equilibrium. Thus, this formulation allows for accounting of the heat transfer between solid and fluid phase.

Fluid Mass Balance

Similar to the LTE formulation, the pressure within the core is described by Eq. (4.7).

$$\nabla.\left[\frac{\eta}{B_w}D_t^{1-\gamma}\left(\frac{\partial p}{\partial x}\right)\right]+q_{w,sc}=\frac{\partial}{\partial t}\left(\frac{\phi}{B_w}\right) \tag{4.23}$$

For convince sake, the functional dependency of rock and fluid properties has been omitted so far.

Conservation of Energy

The energy equation for each phase is presented below, refer to references [189,190] for full details.

$$\frac{\partial[(1-\phi)\rho_s C_{ps}T_s]}{\partial t}=\nabla.[(1-\phi)k_s\nabla T_s]+(1-\phi)q_s^{"}+h_c A_m(T_w-T_s) \tag{4.24}$$

$$\frac{\partial(\phi\rho_w C_{pw}T_w)}{\partial t}+\nabla.\left(\rho_w C_{pf}T_w\right)=\nabla.\left(\phi\lambda_w\nabla T_f\right)+\phi q_w^{"}+h_c A_m(T_s-T_w) \tag{4.25}$$

Where $h= h_c A_m$, $h_c$ is the heat transfer coefficient, $q_{s/w}^{"}$ is the heat production per unit volume and $A_m$ is the matrix pore surface available per unit bulk volume. It must be emphasized here also that proper determination of the appropriate value of *h* is paramount to using the above Equations. It must be noted that in the above Equations the viscous dissipation and the pressure work are assumed to be negligible.

### 4.3.2 Modified Oberbeck-Boussinesq Formulation

Here, a simplified non-isothermal numerical model is developed in this section by employing the OB approximation under the assumption of NOLTE. Just like in the earlier OB model, all fluid properties are considered constants at the initial pressure and temperature conditions with exception of the viscosity and density of water which are allowed to vary with pressure and temperature. Following the same approach as earlier, temperature distribution can be described by the combination of the continuity equation, Darcy equation, an equation of state, and the energy equation for the rock matrix and the fluids respectively.

Fluid Mass Balance

The pressure within the cylindrical core is described by Eq. (4.7) presented earlier.

Conservation of Energy

The temperature distribution along the core length and the fluid phase for the modified OB model under NOLTE is similarly described by Eqs. (4.20) and (4.21) presented earlier.

Table 4.2 presents the functional dependence of physical properties of rock and fluid for the modified OB model under NOLTE.

**Table 4.2 Functional dependence of rock and fluid properties for modified OB Model**

| Variable | Definition | Functional dependence |
|---|---|---|
| $\phi$ | Porosity | $p, T$ |
| $\lambda_w$ | Water thermal conductivity | $const$ |
| $\mu_w$ | Water viscosity | $p, T$ |
| $\rho_w$ | Water density | $p, T$ |
| $C_{pw}$ | Specific heat capacity of water | $const$ |
| $K$ | Permeability | $T$ |

## 4.4 Modelling Alterations in Rock properties during HFI

The two proposed non-isothermal simulators (memory-based and OB-based) proposed in this chapter allow for alteration of rock permeability and porosity with temperature and pressure. The porosity and permeability alteration model- Eq. (4.26) to Eq. (4.28) describe the evolution of these rock properties with time due to change in temperature along the length of the core. These empirical functions are fed into the simulator model to calculate the pressure and temperature along different locations along the core. The following empirical correlations are of the form presented by Civan [24] :

$$\phi(p,T) = \phi_{ref}\left[1 + c_s\left(p - p_{ref}\right) - \beta_s\left(T - T_{ref}\right)\right] \tag{4.26}$$

$$\frac{K}{K_{ref}} = \left\{\frac{1-\left(1-\phi_{ref}\right)exp[f(T)]}{\phi_{ref}}\right\}^{\left(3-\frac{1}{m}\right)} exp\left[-\frac{4}{3}f(T)\right] \tag{4.27}$$

Where, *m* is the cementation factor and *f(T)* is a function, given as

$$f(T) = a\left(T - T_{ref}\right) + \left({}^{b}/_{2}\right)\left(T - T_{ref}\right)^2 + \left({}^{c}/_{3}\right)\left(T - T_{ref}\right)^3 \tag{4.28}$$

In the above equations, *a*, *b* and *c* are empirical constants. To neglect the effect of temperature on the porosity and permeability the empirical constants are assigned zero values respectively. For the case of NOLTE, the rock porosity and permeability are evaluated at the average temperature of the fluid and rock matrix.

# CHAPTER 5

# DEVELOPMENT OF NUMERICAL SIMULATION MODELS

The mathematical models presented in chapter 4 were solved numerically due to their nonlinearity and complexity. In this chapter, finite volume discretization is employed to obtain a set of coupled nonlinear algebraic Equations. Furthermore, the method of solution of the non-algebraic Equations are summarized.

## 5.1 Description of Problem: LTE Approach

The mathematical description of the HFI process under LTE was described in detail in chapter 4. However, to simulate the hot water flooding process we consider a homogeneous and isotropic porous medium with length L, cross-sectional area *A*, porosity $\varphi$ and permeability *K*. The core is initially saturated with cool water with an initial temperature $T_i$. Hot water/brine at constant or variable temperature $T_{inlet}$ is injected in the core at a flow rate *q*. The temperature of the injected water was maintained below the boiling point of water to ensure single phase flow in the core. The schematic of the physical model is presented in Figure 5.1.


**Figure 5.1 Schematic of one-dimensional HFI numerical simulator model**

The HFI problem considered in this research consist of two independent variables, x, and t, and two dependent variables: p, and T. Thus, the objective of the simulator model is to determine the distributions and evolution of the unknowns (dependent variables) at the new time level, ($t_{n+1}= t_n +\Delta \boldsymbol{t}$). The check for the convergence is based upon the change in pressure, and temperature between two successive iterations. All numerical run was carried out with 100 grid blocks. A sequential solution technique is employed using an Implicit Pressure Implicit Temperature scheme (IMPIT).

### 5.1.1 Boundary Conditions

In addition to the differential equations, initial and boundary conditions are required. Initially we assume a uniform pressure and temperature everywhere. The temperature gradient is zero at the external boundary, however at the inflow face we allow any combination of constant heat flux with constant pressure or mass flux.

Formulated exactly these conditions are as follows.

$$p(x,0) = p_{init} \tag{5.1}$$

$$T(x,0) = T_{init} \tag{5.2}$$

The boundary conditions are:

At the core outlet end

$$p(L,t) = p_{out} \tag{5.3}$$

$$\frac{\partial T}{\partial x}(L,t) = 0 \tag{5.4}$$

At the inlet end left, two types of boundary conditions for the pressure and temperature equations were implemented namely; specified rate, and total energy flux as presented below.

$$q(0,t) = q_{HW} \tag{5.5}$$

$$\left(\rho_w C_{pw} uT\right)_{HW} = \rho C_p uT - \left(\{\lambda_e + \alpha_L \rho_w C_{pw}\}\right)\frac{\partial T}{\partial x}, \;\; x = 0, \;\; t > 0 \tag{5.6}$$

These boundary conditions and initial condition, together with the differential equations, completely specify the problem mathematically.

### 5.1.2 Discretization of Pressure Equations

The mathematical models developed thus far must be solved numerically due to their nonlinear nature and complexity i.e. memory formalism. In this section, the finite volume representation of the pressure and energy Equations are summarized. The full details are presented in appendix A, B, and C. To begin with, the core length (with circular cross-sectional area) is discretized into *nx* grid blocks as illustrated in Figure 5.2.

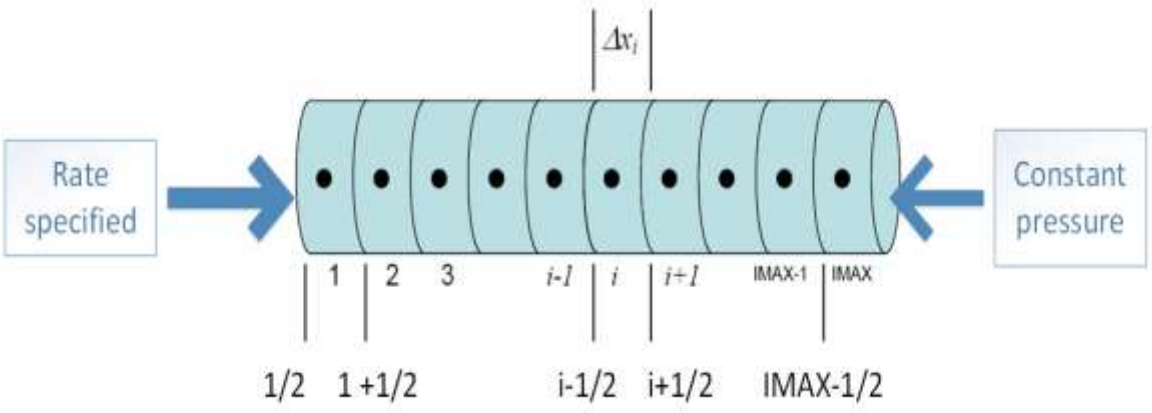


**Figure 5.2 Schematic of one dimensional discretization along core length illustrating the boundary conditions for pressure equations**

This way, each grid block represents a location for which the pressure and temperature needs to be obtained at every time step. For the discretization of the pressure equation a backward Euler scheme is employed, with the coefficients in the pressure equation calculated using the unknown pressure (one iteration step behind). Due to the non-isothermal nature of the problem, the flow related properties of the grid block at the interface between two blocks is approximated with the upwind block properties. These properties are permeability, fluid viscosity, and fluid density. The subscript 'i' represents the grid block number and the superscript 'n' represents the time step. Numerically discretized forms of the terms appearing in the equation are:

$$a_p \mathrm{p}_p = \sum_{nb} a_{nb} p_{nb} + b_p + a_p^0 \mathrm{p}_p^n \tag{5.7}$$

Where,

$$a_E = T_{x,i+\frac{1}{2}}^{n+1} \tag{5.8}$$

$$a_w = T_{x,i-\frac{1}{2}}^{n+1} \tag{5.9}$$

$$a_p^0 = 0 \tag{5.10}$$

$$b_p = \sum_{k=1}^{t/\Delta t} \psi(\gamma, k) \delta_i^{n+1-k} - \frac{V_b}{\alpha_c \Delta t} \left[ \left( \frac{\phi}{B_w} \right)_i^{n+1} - \left( \frac{\phi}{B_w} \right)_i^{n} \right] \tag{5.11}$$

$$a_p = T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} \tag{5.12}$$

Equations (5.7) to (5.12) are only applicable to the interior grid blocks. Refer to appendix A for treatment of boundary conditions, and a step by step discretization of the fully implicit MB pressure equations.

OB Model

The final form of the pressure equation for the OB model is presented in this section. Noting that for this case, the density and viscosity of the fluid can either be taken to be constant as in the strict OB approximation or allowed to vary with pressure and temperature.

Following the same approach as for the MB model, the final form of the pressure equation for the OB model in discretized form is presented below.

$$a_p \mathrm{p}_p = \sum_{nb} a_{nb} p_{nb} + b_p + a_p^0 \mathrm{p}_p^n \tag{5.13}$$

Where,

$$a_E = T_{x,i+\frac{1}{2}}^{n+1} \tag{5.14}$$

$$a_w = T_{x,i-\frac{1}{2}}^{n+1} \tag{5.15}$$

$$a_p^0 = 0 \tag{5.16}$$

$$b_p = -\frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{5.17}$$

$$a_p = T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} \tag{5.18}$$

Equations (5.13) to (5.18) are only applicable to the interior grid blocks. The boundary conditions are treated similar to that presented in appendix A. Appendix B presents a step by step discretization of the OB pressure equation.

### 5.1.3 Discretization of Energy Equations

The energy equation presented in Eq. (4.6) is discretized in this section for the IMPIT and OB models. In what follows, the energy equation would be represented in a shorthand from using the energy flux and some source or sink term as follows:

$$\frac{\partial}{\partial t}[(\rho C_P)_b T] + \frac{\partial}{\partial x}\left(u\rho_w C_{pw} T\right) - \frac{\partial}{\partial x}\left(\left\{\lambda_e + \alpha_L u\rho_w C_{pw}\right\}\frac{\partial T}{\partial x}\right) =$$

$$-\frac{2r_o\pi\Delta x U}{V_b}(T_i - T_\infty) \tag{5.19}$$

Re-writing Eq. (5.19) as follows;

$$\frac{\partial\left[(\rho C_p)_b T\right]}{\partial t} + \nabla.\bar{J} = \left(S_c + S_p T\right) \tag{5.20}$$

Implies the following,

$$\bar{J} = u\rho_w C_{pw} T - \left(\lambda_e + \alpha_L u\rho_w C_{pw}\right)\frac{\partial T}{\partial x} \tag{5.21}$$

$$S_c = \frac{2r_o\pi\Delta x U T_\infty}{V_b} \tag{5.22}$$

$$S_p = -\frac{2r_o\pi\Delta x U}{V_b} \tag{5.23}$$

IPIT Scheme / OB Model

The discretized form for the energy equation for the IMPIT scheme and the OB model is derived in detail in appendix D. The final form is presented below.

$$a_P T_i^{n+1} = a_E T_{i+1}^{n+1} + a_W T_{i-1}^{n+1} + b_i + a_P^0 T_i^n \tag{5.24}$$

Where,

$$a_P = \left[\frac{V_{b,i}\left(\rho C_{p,i}\right)^{n+1}}{\Delta t} + F_i^{n+1} + D_{i+\frac{1}{2}}^{n+1} + D_{i-\frac{1}{2}}^{n+1} - S_{p,i}V_{b,i}\right] \tag{5.25}$$

$$a_E = D_{i+\frac{1}{2}}^{n+1} \tag{5.26}$$

$$a_W = D_{i-\frac{1}{2}}^{n+1} + F_{i-1}^{n+1} \tag{5.27}$$

$$a_P^0 = \frac{V_{b,i}(\rho C_{p,i})^n}{\Delta t} \tag{5.28}$$

$$b_i = S_{c,i} V_{b,i} \tag{5.29}$$

Eqs. (5.24) to (5.29) have to be modified to incorporate boundary conditions as shown in Appendix C also.

## 5.2 Description of Problem: No Local Thermal Equilibrium

The HFI process under NOLTE described in detail in chapter 4 differs from the LTE formulation in that the temperature in the rock matrix, temperature in the fluid phase and fluid pressure are the primary variables as opposed to two unknowns (the fluid pressure and an equilibrium temperature). Using the same schematic presented in Figure 5.1, the objective of the NOLTE simulation models is to determine the distributions and evolution of the unknowns (dependent variables) at the new time level, ($t_{n+1} = t_n + \Delta \boldsymbol{t}$). The check for the convergence is based upon the change in pressure, and rock and fluid temperatures between two successive iterations. The sequential solution technique is employed using an Implicit Pressure Implicit Temperature scheme (IMPIT).

### 5.2.1 Boundary Conditions

The following initial and boundary conditions were considered.

$$p(x, 0) = p_{init} \tag{5.30}$$

$$T_w(x, 0) = T_s(x, 0) = T_{init} \tag{5.31}$$

The boundary conditions are:

At the core outlet end

$$p(L, t) = p_{out} \tag{5.32}$$

$$\frac{\partial T_w}{\partial x} = \frac{\partial T_s}{\partial x} = 0, \quad \mathrm{x = L, t > 0} \tag{5.33}$$

At the inlet end left, two type of boundary conditions for the pressure equation was considered namely; constant pressure (injection pressure), and constant rate as presented below.

$$q(0,t) = q_{HW} \tag{5.34}$$

$$\left(\rho_w u C_{pw} T\right)_{inlet} = \rho u C_{pw} T - \phi \lambda_w \frac{\partial T_w}{\partial x}, x = 0, \ \ t > 0 \tag{5.35}$$

$$\frac{\partial T_s}{\partial x} = 0, \ x = 0\,, t > 0\,, \tag{5.36}$$

These boundary conditions and initial condition, together with the differential equations, completely specify the problem mathematically.

### 5.2.2 Discretization of Pressure Equation

Due to the complexity and coupled nature of the presented equations, only numerical solution can be obtained. The discretized form of the pressure equation is presented as follows:

$$a_p \mathrm{p}_p = \sum_{nb} a_{nb} p_{nb} + b_p + a_p^0 \mathrm{p}_p^n \tag{5.37}$$

Where,

$$a_E = \ T_{x,i+\frac{1}{2}}^{n+1} \tag{5.38}$$

$$a_w = T_{x,i-\frac{1}{2}}^{n+1} \tag{5.39}$$

$$a_p^0 = 0 \tag{5.40}$$

$$b_p = \sum_{k=1}^{t/\Delta t} \psi(\gamma, k) \delta_i^{n+1-k} - \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{5.41}$$

$$a_p = T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} \tag{5.42}$$

Eqs. (5.37) to (5.42) are only applicable to the interior grid blocks. Refer to appendix A for a step by step discretization of the fully implicit MB pressure equation.

OB Model

The final form of the pressure equation for the OB model under NOLTE is:

$$a_p \mathrm{p}_p = \sum_{nb} a_{nb} p_{nb} + b_p + a_p^0 \mathrm{p}_p^n \tag{5.43}$$

Where,

$$a_E = T_{x,i+\frac{1}{2}}^{n+1} \tag{5.44}$$

$$a_w = T_{x,i-\frac{1}{2}}^{n+1} \tag{5.45}$$

$$a_p^0 = 0 \tag{5.46}$$

$$b_p = -\frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{5.47}$$

$$a_p = T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} \tag{5.48}$$

Equations (5.43) to (5.48) are only applicable to the interior grid blocks. Appendix B presents a step by step discretization of the OB pressure equation.

### 5.2.3 Discretization of Energy Equation

IMPIT Scheme / OB Model

In order to solve Eqs. (4.20) and (4.21) numerically, these equation need to be discretized and converted to an algebraic form. The final form of the energy equation for rock matrix and fluid phase are presented below. Refer to Appendix D for complete formulation.

**Rock Matrix**

$$a_p T_{s,i}^{n+1} = a_w T_{s,i-1}^{n+1} + a_e T_{s,i+1}^{n+1} + a_p^o T_{s,i}^n + b \tag{5.49}$$

Where the coefficients are defined below

$$a_w = D_{s,i-\frac{1}{2}} = A_x \left\{\frac{k_s(1-\phi)}{\Delta x}\right\}_{i-\frac{1}{2}}^{n+1} \tag{5.50}$$

$$a_e = D_{s,i+\frac{1}{2}} = A_x \left\{\frac{k_s(1-\phi)}{\Delta x}\right\}_{i+\frac{1}{2}}^{n+1} \tag{5.51}$$

$$a_p^o = \frac{V_b}{\Delta t}\left\{(1-\phi)\rho_s C_{ps}\right\}_i^n \tag{5.52}$$

$$a_p = a_e + a_w + \frac{V_b}{\Delta t}\left\{(1-\phi)\rho_s C_{ps}\right\}_i^{n+1} + h_i^{n+1} V_b \tag{5.53}$$

$$b = h_i^{n+1} V_b T_{w,i}^{n+1} \tag{5.54}$$

**Fluid phase**

$$a_p T_{w,i}^{n+1} = a_w T_{w,i-1}^{n+1} + a_e T_{w,i+1}^{n+1} + a_p^o T_{w,i}^n + b \tag{5.55}$$

Where the coefficients are defined below

$$a_w = D_{w,i-\frac{1}{2}} + F_w \tag{5.56}$$

$$a_e = D_{e,i+\frac{1}{2}} \tag{5.57}$$

$$a_p^o = \frac{V_b}{\Delta t}\left(\phi\rho_w C_{pw}\right)_i^n \tag{5.58}$$

$$a_p = D_{e,i+\frac{1}{2}} + D_{w,i-\frac{1}{2}} + F_e + \frac{V_b}{\Delta t}\left(\phi\rho_w C_{pw}\right)_i^{n+1} + h_i^{n+1} V_b \tag{5.59}$$

$$b = h_i^{n+1} V_b T_{s,i}^{n+1} \tag{5.60}$$

In the numerical algorithm, the temperature profile in the water saturated porous media is obtained by solving Eqs. (5.37) or (5.43), (5.49) and (5.55). Eq. (5.37) or (5.43) is solved for the pressure within the porous domain to obtain the velocity, whereas Eqs. (5.49) and (5.55) are solved for the temperature profile in the rock matrix and fluid respectively.

## 5.3 Treatment of Nonlinearities

The partial differential equations presented above contain several nonlinear functions. Thus, to obtain physically realistic and stable solutions, these terms must be treated with care. Fortunately, the nonlinearities in the proposed equations are weak in nature. The inter-block transmissibility which accounts for fluid and rock properties are approximated in space using the harmonic averaging method as presented in Eqs. (5.61) and (5.62).

Mathematically we employed the following,

$$T_{i+\frac{1}{2}} = \frac{2T_i T_{i+1}}{T_i + T_{i+1}} \tag{5.61}$$

and

$$T_{i-\frac{1}{2}} = \frac{2T_i T_{i-1}}{T_i + T_{i-1}} \tag{5.62}$$

For time approximation, the fluid phase density, viscosity, thermal conductivity, porosity, specific heat capacity and the permeability are evaluated at the current time level *n+1* (one iteration step behind). Although, to reduce the nonlinearity of the nonlinear algebraic equations, the fluid and rock properties can be evaluated at time level *n*. However, if this approach is taken the time step should be small to ensure accurate results are predicted.

## 5.4 Solution Scheme

In this section, the solution procedure is summarized for a time step:

I. The coefficients of the pressure equation are first calculated using values of pressure and temperature at the end of the of the previous time step as an initial guess. For the first time step, the values at the initial conditions are used.

II. The pressure equation is solved using the successive over relaxation method.

III. The coefficients of the energy equation are updated using the new pressure estimated above.

IV. Solve the energy equation to obtain the temperature at the next time step.

V. Compare the obtained pressure and temperature with the previous iteration estimates to the accepted tolerance.

VI. If no, use the new estimate of pressure and temperature to obtain the coefficient of the pressure equation and repeat step II to V.

VII. When the tolerance is satisfied move to the next time step.

Figure 5.3 illustrates the algorithm which the numerical simulator employs at each time step to obtain the pressure and temperature at each location.

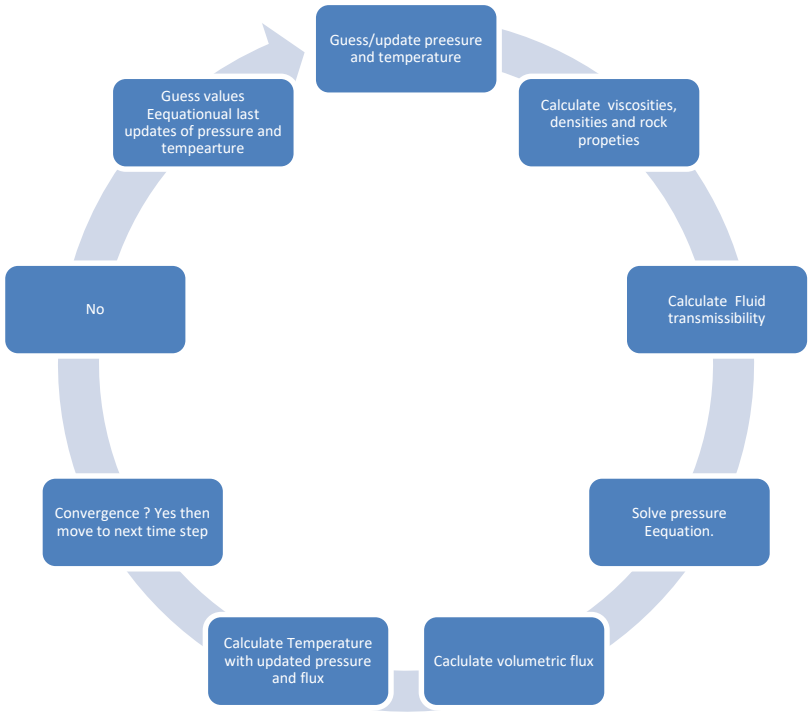


**Figure 5.3 Flowchart for numerical simulator per time step**

# CHAPTER 6

# RESULTS AND DISCUSSION

The result and discussion chapter comprises of two chapters (Chapter 6 and Chapter 7). The results presented in this chapter is presented in two sub-sections. In the first sub-section results from the isothermal memory-based formalism are presented employing an explicit finite difference scheme.

## 6.1 Explicit finite difference approximation

Before investigating the non-isothermal fluid flow problem, it seemed necessary to understand/ quantify the effect of the memory formulation on the pressure and flux in a reservoir. The governing equations describing fluid flow during isothermal fluid flow in an oil reservoir are the same as the equations presented in Chapter 4, and subsequently discretized in Chapter 5. However, for isothermal fluid flow problem the energy equation is not required. Secondly, the fluid and rock properties are a function of the pore pressure. Thus in this section, the mathematical models and discretization methods are not repeated.

For the isothermal case, we begin by presenting the treatment of the accumulation term in the fluid mass balance equation, then the definition of the composite variable employed for numerical simulation is shown, furthermore the stability analysis of the numerical scheme and a newly developed wellbore model are derived. Finally, we show through different case studies the effect of the memory exponent on reservoir pressure, porosity, wellbore pressure and the composite variable "$\eta$".

### 6.1.1 Treatment of accumulation term

For the case of isothermal fluid flow the right hand side (RHS) of Eq. (4.7) was treated as follows [191]:

$$\frac{\partial}{\partial t}\left(\frac{\phi}{B_o}\right) \approx \frac{\phi_o(c_s+c_o)}{B_{ob}\Delta t}\ (p^{n+1}-p^n) \tag{6.1}$$

Where, $B_o$, $B_{ob}$, $\phi_o$, and $c_o$ are the oil formation volume factor, the oil formation volume factor at bubble point pressure, the reference porosity and the oil compressibility respectively. This conservative approximation of the accumulation terms has the advantage of speeding up convergence [191].

### 6.1.2 Definition of composite variable

One of the important parameters in the above mathematical model is the composite variable $\eta$, which is time dependent rock and fluid property (e.g. rock permeability, time and fluid viscosity) as presented in Eq. (13). Thus, proper characterization of the type of reservoir fluid and rock is paramount to proper flow modelling. In the literature, we acknowledge that the memory coefficient is addressed as a pseudo permeability which is a dynamic property which at present cannot be measured in the laboratory [182]. Therefore, it has no physical meaning and thus not related to the physical property of the rock. Similarly, the memory exponent is an anomalous coefficient and has nothing to do with the static property of the rock and also cannot be obtained in the laboratory. In other approaches [88,182,192], the memory coefficient and the memory exponent are obtained by inverse modelling i.e. minimizing the difference between experimental data and the mathematical model. This implies a fitting between field data or experimental data from which the memory parameters are obtained. However, in this research the memory approach presented by Hossian et *al.*[3,5,178] was employed throughout with its validity not yet established.

To solve the fractional model Eq. (12), Carman-Kozeny (1939) permeability correlation is used as an example case study for simplicity. However, [193] model may be employed instead of Carman-Kozeny model because it is established as the best model in explaining complex reservoir [194].

The composite variable $\eta$ is defined as follows for the case study to be presented:

$$\eta = \frac{\beta_c[K(\phi,\ d_p)]}{\mu_{ab}\ exp[c_\mu(p-p_b)]} t^{1-\gamma}\ ,\ \ K = \Upsilon_c \frac{1}{72\tau}\frac{\phi^3 d_p^2}{(1-\phi)^2} \tag{6.2}$$

Where $\Upsilon_c$ is a conversion factor equal to 0.9869e-15.

For analysis, we consider a single phase slightly compressible fluid with porosity variation with pressure described by Eq. (6.4).

$$\phi = \phi^0 exp[c_s(p-p_b)] \tag{6.3}$$

In addition, in this study we assume that the following correlations describe the reservoir fluid properties. Employing Almehaideb [195] correlations for UAE crude oil we have as follows:

The oil viscosity above the bubble-point pressure can be correlated by:

$$\mu = \mu_{ob} exp\left[c_{\mu p}(p - p_b)\right] \tag{6.4}$$

Where $\mu_{0b}$, is the oil viscosity at the bubble-point pressure obtained from Eq. (6.6).

$$\mu_{ob} = 6.59927 \times 10^5 R_s^{-0.597627} T^{-0.941624} \times \gamma_g^{-0.555208} API^{-1.487449} \tag{6.5}$$

Likewise, the following equations present the correlations used for determining other reservoir fluid properties:

$$p_b = -620.592 + 6.23087\ \frac{R_s \gamma_o}{\gamma_g B_o^{1.38559}} + 2.89868\ T \tag{6.6}$$

$$B_o = B_{ob} exp[-c_o(p - p_b)] \tag{6.7}$$

$$B_{ob} = 1.122018 + 1.410 \times 10^{-6} \frac{R_s T}{\gamma_o^2} \tag{6.8}$$

$$c_o = \frac{\left(1433 + 5R_s + 17.2T - 1180\gamma_g + 12.61\ API\right)}{(p \times 10^5)} \tag{6.9}$$

The above fluid correlations were developed specifically for UAE crude oil between some specific range of pressure, temperature, API, gas gravity, formation volume factor and solution gas oil ratio.

### 6.1.3 Development of numerical scheme

In the finite difference methods, the space-time solution's domain is discretized. We shall use the following notation: $\Delta t$ is the temporal mesh or time step, $\Delta x$ is the spatial mesh along the reservoir length. The coordinates of the mesh points are $x_i = i\Delta x$ and $t_n = n\Delta t$, and the values of the solution $P(x,t)$ on these grid points are $P(x_i, t_n) \equiv p_i^n \approx P_i^n$, where we denote by the numerical estimate of the exact value of $P(x,t)$at the point $(x_i, t_n)$ by $p$.

Combining Eq. (4.7) and Eq. (6.1) gives the following:

$$\frac{\partial}{\partial x}\left\{g_x(p) D_t^{1-\gamma} \frac{\partial p}{\partial x}\right\} \Delta x + q_{sc} = \chi(p) \frac{(p^{n+1} - p^n)}{\Delta t} \tag{6.10}$$

The coefficients, $B_o^0 = B_{ob},\ \chi(p) = \frac{V_b}{\alpha_c}\frac{\phi^0(c_o+c_s)}{B_o^0 \Delta t}$ , and $\phi^0$ are evaluated at the bubble point pressure.

In order to discretize the $D_t^{1-\gamma}$ operator, Eq. (4.3) was employed

Next, we define a function $\psi(\gamma, k)$,

$$\psi(\gamma, k) = \frac{(-1)^k\, \Gamma(2-\gamma)}{k!\, \Gamma(2-\gamma-k)} \tag{6.11}$$

Noting that $\psi(\gamma, 0) = 1$, and

$$\psi(\gamma, k) = -\psi(\gamma, k-1)\frac{(2-\gamma-k)}{k}, \text{ for } k > 0 \tag{6.12}$$

So Eq. (6.10) can be re-written as:

$$\chi(p)\frac{(p^{n+1}-p^n)}{\Delta t} = \lim_{\tau \to 0} \tau^{\gamma-1} \sum_{k=0}^{t/\tau} \psi(\gamma, k) \frac{\partial}{\partial x}\left\{ g_x(p) \frac{\partial p(t-k\tau)}{\partial x} \right\} \Delta x + q_{sc} \tag{6.13}$$

For numerical implementation, we discretize next the spatial derivative in Eq. (6.13) using Forward in Time and Central in Space Scheme (FTCS):

$$\frac{\chi_i^{n+1}}{\Delta t}(p_i^{n+1} - p_i^n) = q_{sc,i}^n +$$

$$\frac{1}{(\Delta t)^{1-\gamma}} \sum_{k=0}^{t/\Delta t} \psi(\gamma, k) \left[ \left\{ \frac{g_x(p)}{\Delta x} \right\}_{i+\frac{1}{2}}^{n-k} \left( p_{i+1}^{n-k} - p_i^{n-k} \right) - \left\{ \frac{g_x(p)}{\Delta x} \right\}_{i-\frac{1}{2}}^{n-k} \left( p_i^{n-k} - p_{i-1}^{n-k} \right) \right] \tag{6.14}$$

Therefore Eq. (6.14) can be further simplified

$$\frac{\chi_i^{n+1}}{\Delta t}(p_i^{n+1} - p_i^n) - {q_{sc,i}}^n =$$

$$\sum_{k=0}^{t/\Delta t} \psi(\gamma, k) \left[ T_{x,i-\frac{1}{2}}^{n-k} p_{i-1}^{n-k} - \left( T_{x,i-\frac{1}{2}}^{n-k} + T_{x,i+\frac{1}{2}}^{n-k} \right) p_i^{n-k} + T_{x,i+\frac{1}{2}}^{n-k} p_{i+1}^{n-k} \right] \tag{6.15}$$

In Eq. (6.15) the fluid transmissibility and pseudo-compressibility term were introduced

$$T = \frac{\beta_c \eta A_x}{B \Delta x (\Delta t)^{1-\gamma}}, \text{ and } C_{p,i}^{n+1} = \frac{\chi_i^{n+1}}{\Delta t}$$

Taking all the like terms together we have

$$p_i^{n+1} = p_i^n + \frac{1}{C_{p,i}^{n+1}} \left\{ {q_{sc,i}}^n + \sum_{k=0}^{n} \psi(\gamma, k) \delta_i^{n-k} \right\} \tag{6.16}$$

Where $\delta_i^{n-k}$is the finite difference kernel given below

$$\delta_i^{n-k} = T_{x,i-\frac{1}{2}}^{n-k} p_{i-1}^{n-k} - \left(T_{x,i-\frac{1}{2}}^{n-k} + T_{x,i+\frac{1}{2}}^{n-k}\right) p_i^{n-k} + T_{x,i+\frac{1}{2}}^{n-k} p_{i+1}^{n-k} \quad (6.17)$$

Equations (6.16) and (6.17) are applicable to the interior control volumes$1 \leq i \leq N_x$, where *i* is the grid counter, and $N_x$ the number of control volume in x direction.

Similarly, Eqs. (6.16) and (6.17) can be extended for a 2-dimensional flow by intuition as shown below

$$p_{i,j}^{n+1} = p_{i,j}^{n} + \frac{1}{C_{p,i,j}^{n+1}} \left\{ {q_{sc,i,j}}^{n} + \sum_{k=0}^{n} \psi(\gamma, k) \delta_{i,j}^{n-k} \right\} \quad (6.18)$$

However the finite difference kernel becomes:

$$\delta_{i,j}^{n-k} = T_{x,i-\frac{1}{2},j}^{n-k} p_{i-1,j}^{n-k} + T_{y,i,j-\frac{1}{2}}^{n-k} p_{i,j-1}^{n-k} - \left(T_{x,i-\frac{1}{2},j}^{n-k} + T_{y,i,j-\frac{1}{2}}^{n-k} + + T_{x,i+\frac{1}{2},j}^{n-k} + T_{y,i,j+\frac{1}{2}}^{n-k}\right) p_{i,j}^{n-k} +$$

$$T_{x,i+\frac{1}{2},j}^{n-k} p_{i+1,j}^{n-k} + T_{y,i,j+\frac{1}{2}}^{n-k} p_{i,j+1}^{n-k} \quad (6.19)$$

Equations (6.18) and (6.19) are applicable to the interior control volumes$1 < i < N_x; 1 < j < N_y$ where *i* and *j* are the grid counters, and $N_x$ and $N_y$ are the number of control volume in x and y directions.

It is always preferable to refer to coefficients in multi-dimensional flow problems (2D, or 3D) with a single counter. In here the two dimensional problem to be considered herein would be indexed using a single index (counter) using the following procedure.

We count the cells in the y direction (vertical direction) first with the *i* index. Then the x direction with the *h* index, then the y direction again and so forth. This counting algorithm can be expressed mathematically as:

$$m = i + (h - 1)Ny \quad (6.20)$$

Therefore, Eqs. (6.18) and (6.19) can be re-written as follows:

$$p_m^{n+1} = p_m^{n} + \frac{1}{C_{p,m}^{n+1}} \left\{ {q_{sc,m}}^{n} + \sum_{k=0}^{n} \psi(\gamma, k) \delta_m^{n-k} \right\} \quad (6.21)$$

and the finite difference kernel becomes:

$$\delta_m^{n-k} = T_{m,m-Ny}^{n-k} p_{m-Ny}^{n-k} + T_{m,m-1}^{n-k} p_{m-1}^{n-k} -$$

$$\left(T_{m,m-Ny}^{n-k} + T_{m,m-1}^{n-k} + + T_{m,m+Ny}^{n-k} + T_{m,m+1}^{n-k}\right) p_m^{n-k} + T_{m,m+Ny}^{n-k} p_{m+Ny}^{n-k} + T_{m+1}^{n-k} p_{m+1}^{n-k} \quad (6.22)$$

### 6.1.4 Wellbore model

The wellbore pressure is usually calculated using the steady-state model proposed by Peaceman [196]. However, due to the memory formalism, some modification is required to obtain wellbore pressure. Starting from Eq. (4.2), the volumetric rate can be expressed as:

$$q_{sc} = -\frac{2\pi\,\beta_c\,\eta_m^{n+1}\,\Delta z}{B_{o,m}^{n+1}\,ln\left(\frac{r_{eq}}{r_w}\right)} D_t^{1-\gamma}\left(P_m^{n+1} - P_{wf,m}^{n+1}\right) \tag{6.23}$$

Where $r_{eq} = 0.14\sqrt{(\Delta x)^2 + (\Delta y)^2}$ is the equivalent radius and $r_w$ is the wellbore radius.

Recall that,

$$D_t^{1-\gamma} f(t) = \Delta t^{\gamma-1} \sum_{k=0}^{t/\Delta t}\{\psi(\gamma,k) f(t^{n+1-k})\} \tag{6.24}$$

Thus, the wellbore pressure is calculated by:

$$P_{wf,m}^{n+1} = P_m^{n+1} + \left[\left(\frac{q_{sc}\,B_{o,m}^{n+1}}{2\pi\,\Delta z\,\beta_c \Delta t^{\gamma-1}\,\eta_{H,m}^{n+1}}\right) ln\left(\frac{r_{eq}}{r_w}\right) + \sum_{k=1}^{t/\Delta t}\left\{\psi(\gamma,k)\;\delta_{wf}^{n+1-k}\right\}\right] \tag{6.25}$$

Where $\delta_{wf}^{n+1-k} = \left(P_m^{n+1-k} - P_{wf,m}^{n+1-k}\right)$, $\eta_H = \sqrt{\left(\eta_x \times \eta_y\right)}$

### 6.1.5 Applications

To test our finite difference scheme we consider a reservoir 1463. 04 m(4800 feet) long, 1463.04 m (4800 feet) wide and 30.48 m (100 feet) thick. Given the difficulty of measuring rock properties, it is common to use geostatistical methods to make realizations of porosity and permeability. In this study, we will generate the porosity $\phi$ values as a Gaussian field. As a simple approximation to a Gaussian field, we generate a field of independent normally-distributed porosity values as presented in Figure 6.1. To get a crude approximation to the porosity-permeability- relationship, we assume that our medium is made up of uniform spherical grains of diameter $d_p$ = 10 $\mu$m and $\tau$ = 0.81 for which the specific surface area is $A_v = 6/d_p$. Using the Carman-Kozeny relation (Eq. 6.2), we can then calculate the isotropic permeability field as shown in Figure 6.2.

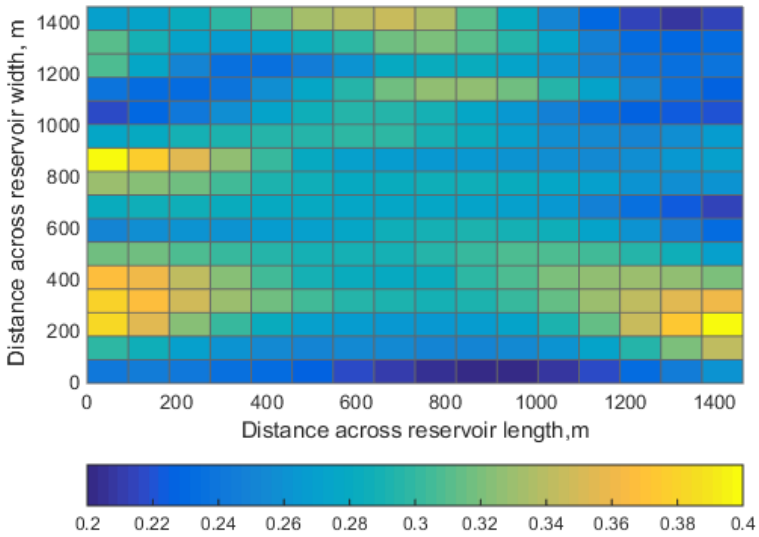


Figure 6.1 Generated initial porosity distribution

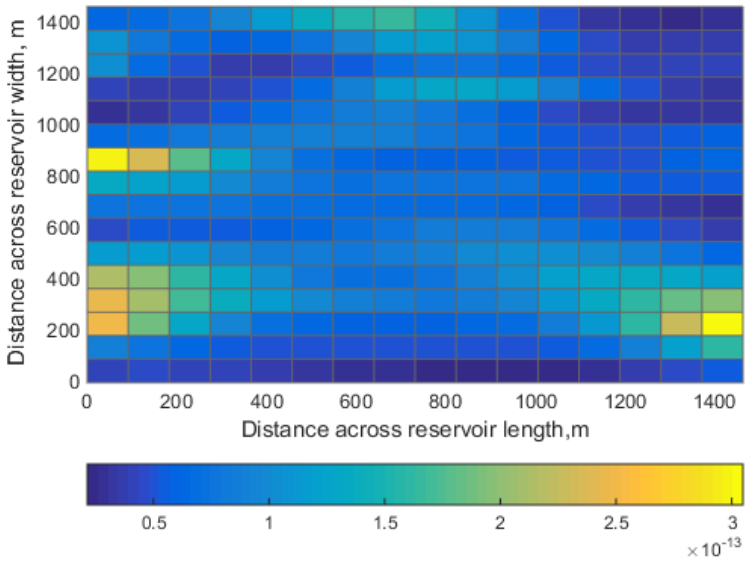


Figure 6.2 Generated initial permeability distribution

The reservoir is assumed to be completely sealed at the right boundary with flow entering the left boundary. Similarly, a constant pressure boundary is assumed at the bottom boundary with flow leaving the reservoir at the top boundary (values presented as the first four parameters in Table 6.1). The rock grain is assumed to be made up of spherical grains with diameter $d_p$ of 10$\mu$m and tortuosity 0.81. The properties of the crude oil used for this computation is also presented in Table 1. Due to the uncertainty of the fractional order derivative we choose $\gamma$ to vary between 0.4 and 1, with $\Delta x$= 91.44 m, $\Delta y = 91.44$ m,$\Delta t =$ 0.0002 days and t = 20 days. The memory-based flow model is solved with MATLAB computing language using Eq. (6.21), with the following initial condition: $p(x, 0) = p_0$.

**Table 6.1 Reservoir parameters and boundary condition**

| Parameter | Value |
|---|---|
| $Q_{x0}$ | 3.68e-4 $m^3/s$ (200 bbl/day) |
| $p_{y0}$ | 3.1e4.kPa (4500psi) |
| $Q_{yL}$ | 5.55e-4 $m^3/s$ (300 bbl/day) |
| $API$ | 31° |
| $T$ | 366 K |
| $R_s$ | 23 $sm^3/sm^3$ (129 scf/stb) |
| $\gamma_g$ | 0.748 |
| $p_i$ | 3.1e4.kPa (4500 psi) |
| $\gamma$ | 0.9 |
| $c_s$ | 5.07632082e-7 kPa (3.5e-6 $psi^{-1}$) |
| $c_\mu$ | 1.22e-5 kPa (8.422e-5 $psi^{-1}$) |

At the left and top boundaries, the rates are specified for the whole reservoir boundary i.e. $Q_{x0}$ and $Q_{yL}$. Thus for each grid block some sort of prorating $q_{x0}$ and $q_{yL}$ among all boundary grid blocks that share that boundary was employed as follows:

$$q_{x0,m} = \frac{T^n_{x,bB}}{\sum_{l\epsilon\psi_b} T^n_{xb,l}} Q_{x0} \text{ and } q_{yL,m} = \frac{T^n_{y,bB}}{\sum_{l\epsilon\psi_b} T^n_{yb,l}} Q_{yL} \quad (6.26)$$

Where $\psi_b$ is the set that contains all the boundary grid blocks that share the specified boundary condition. Subscript bB refer to the boundary block, and $T_{xb,l}$ and $T_{yb,l}$ refers to the transmissibility between the reservoir boundary and boundary grid block $l$.

### 6.1.6 Result Discussion

**Case 1**: Model consistency check with Classical Fluid Model i.e. no memory effect

Before proceeding with any investigations it is necessary to establish the accuracy of the proposed numerical scheme. However, deriving an analytical solution to Eq. (4.1) seems impossible due to the nonlinear nature of the equation. Nevertheless, we propose validating our proposed numerical scheme for $\gamma = 1$ with the classical fluid flow model. This is because the GL fractional derivative operator ($D_t^{1-\gamma}$) reduces to the identity operator. In addition, the term $\eta$ simplifies to:

$$\eta = \frac{\beta_c K(p)}{\mu(p)} \tag{6.27}$$

The producer and injector well locations and flow rates used for our simulations are presented in Table 6.2.

**Table 6.2 Producer and Injector locations**

| $h$ | $i$ | $m$ (based on Eq. (6.20) | $q_{sc}$ stb/day | $q_{sc}$ $m^3/s$ |
|---|---|---|---|---|
| 13 | 2 | 194 | 600 | $1.1\times 10^{-3}$ |
| 2 | 10 | 26 | 800 | $1.47\times 10^{-3}$ |
| 3 | 1 | 33 | -650 | $-1.2\times 10^{-3}$ |
| 7 | 9 | 105 | -750 | $-1.38\times 10^{-3}$ |
| 12 | 14 | 190 | -850 | $-1.56\times 10^{-3}$ |

Figures 6.3 and 6.4, present the predicted wellbore pressure history at block 26, and 33 and the control volume pressure history at blocks 56 and 232 for both the newly proposed memory-based model, and the classic fluid flow model (Darcy based equation) respectively. The classic fluid flow model was solved using an explicit finite-difference scheme. As expected, the results of both models are approximately equal throughout the simulation run time.

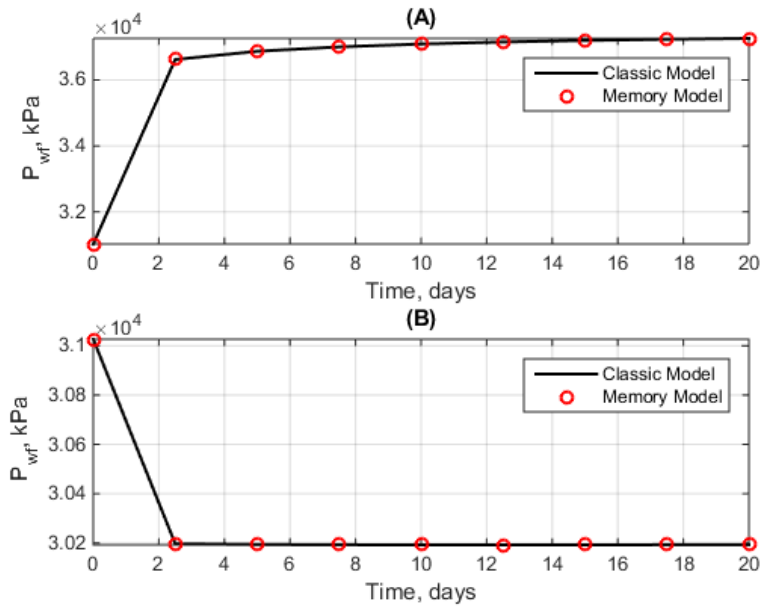


**Figure 6.3 Model consistency check with classic model based on wellbore pressures for (A) Injector at block 26 (B) Producer at block 33**

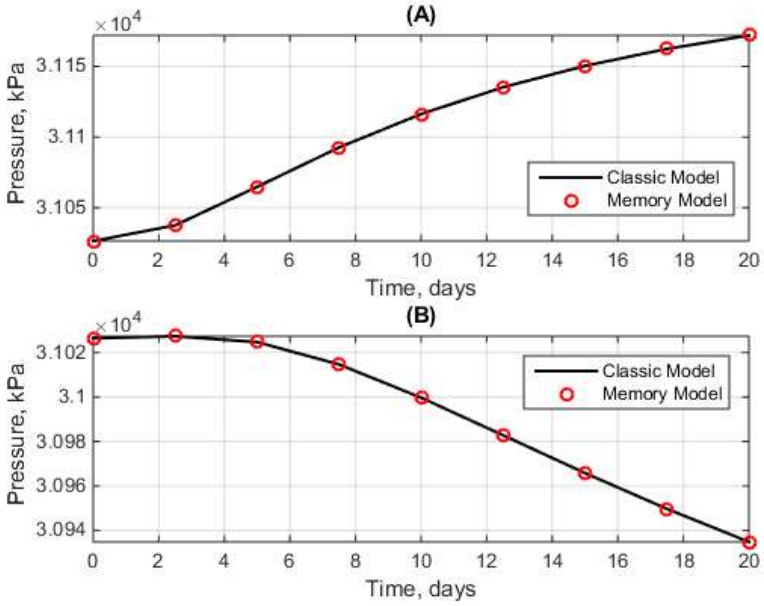


**Figure 6.4 Model consistency check with classic model based on block pressures for (a) Block 56 (b) Block 232**

**Case 2**: Producer and injector wells present in reservoir with boundary condition described earlier

Having validated the accuracy of our proposed numerical scheme, we present the results of a case study based on the proposed memory-based model. Figures 6.5 and 6. 6 presents a contour plot of the pressure distribution across the reservoir at different times (0.5 and 20 days) assuming the reservoir flow can be best characterized with $\gamma = 0.9$. It is always recommended to carry out a material balance check to ensure the predicted results from the simulator conserves mass since the reliability of simulators is questionable either due to human error or truncation error. The incremental material balance at the end of each time step was calculated to ensure or ascertain the simulator results obeyed the conservation of mass using Eq. (6.28) modified from that proposed in [180]

$$I_{MB} = \frac{\sum_{m=1}^{M} \frac{V_{bm}}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_o}\right)_m^{n+1} - \left(\frac{\phi}{B_o}\right)_m^{n}\right]}{\sum_{m=1}^{M}\{q_{sc,m}^{n+1} + \sum_{l \epsilon \xi_m}(q_{sc,m}^{n+1}) + \sum_{m=1}^{M}(q_{sc,memory}^{n+1})\}} \tag{6.28}$$

Where $q_{sc,memory,}$ represents a fictitious source term capturing the contribution of the memory formalism.

Accordingly, the incremental material balance should return a value very close or preferable equal to one to ensure reliable solutions.

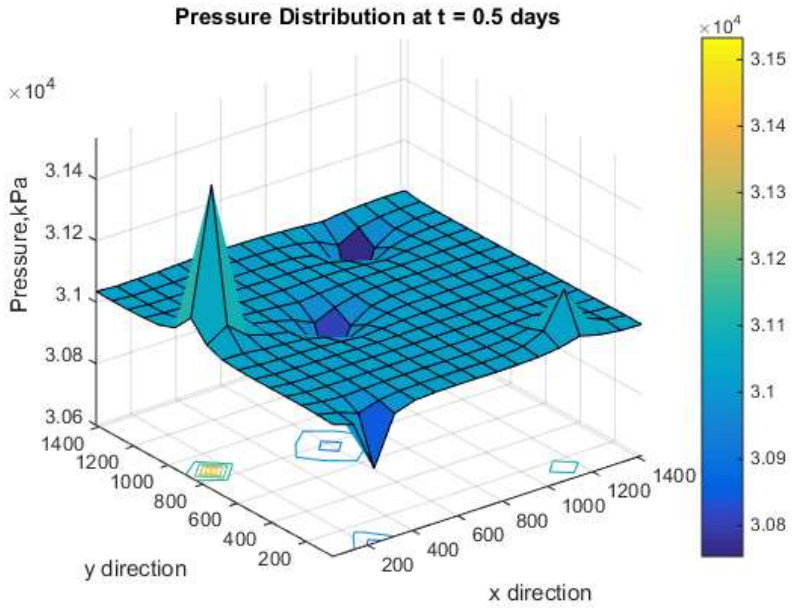


**Figure 6.5 Reservoir pressure distribution at 0.5 days for $\gamma = 0.9$**

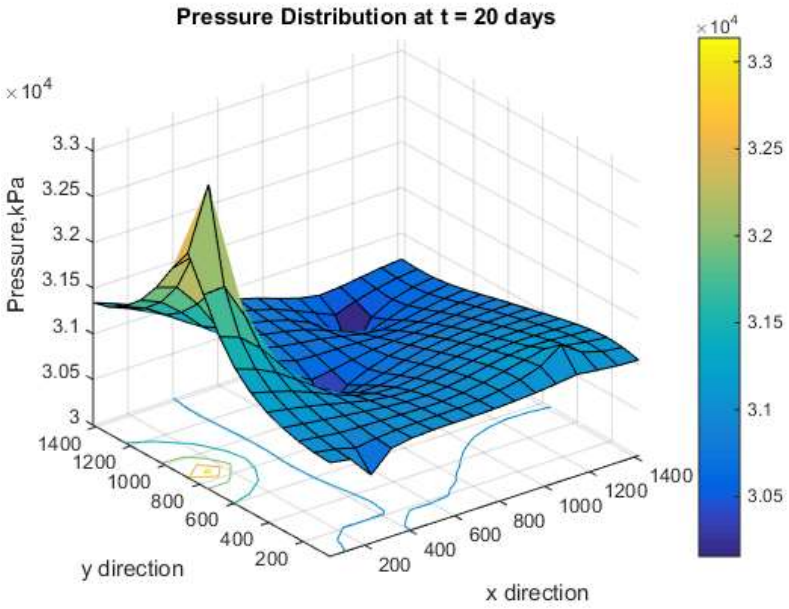


**Figure 6.6 Reservoir pressure distribution at 20 days for $\gamma = 0.9$**

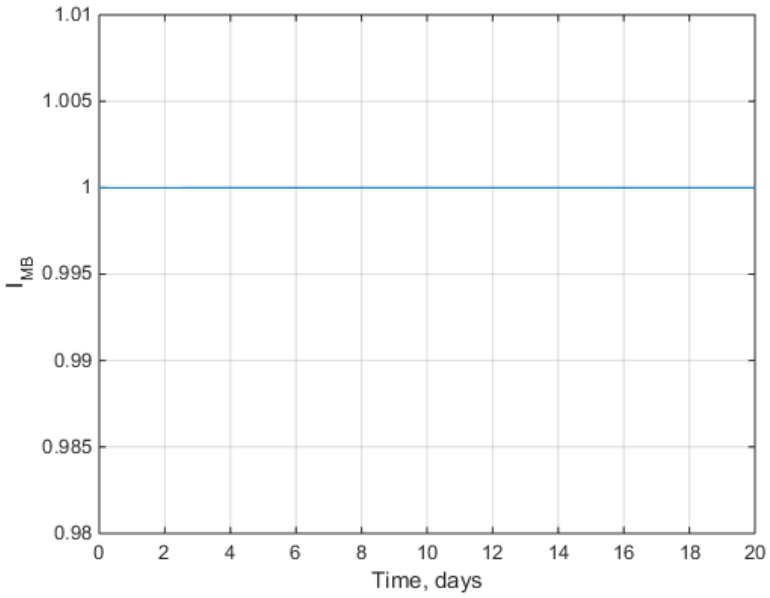


**Figure 6.7 Incremental material balance check for $\gamma = 0.9$**

As can be observed from Figure 6.7, the incremental balance check at each time step was equal to one. Thus, we can carry out further investigation with confidence in the output of the simulator. In addition, the pressure profile at the centerline across the reservoir domain is presented in Figure 6.8 at different time intervals. As can be seen, a rapid increase in pressure is observed at the left boundary due to the influx from that boundary. However, it

can be noticed that around 609 m (2000ft.) from the left boundary the effect of a producer is felt due to the drop in pressure observed around this region.

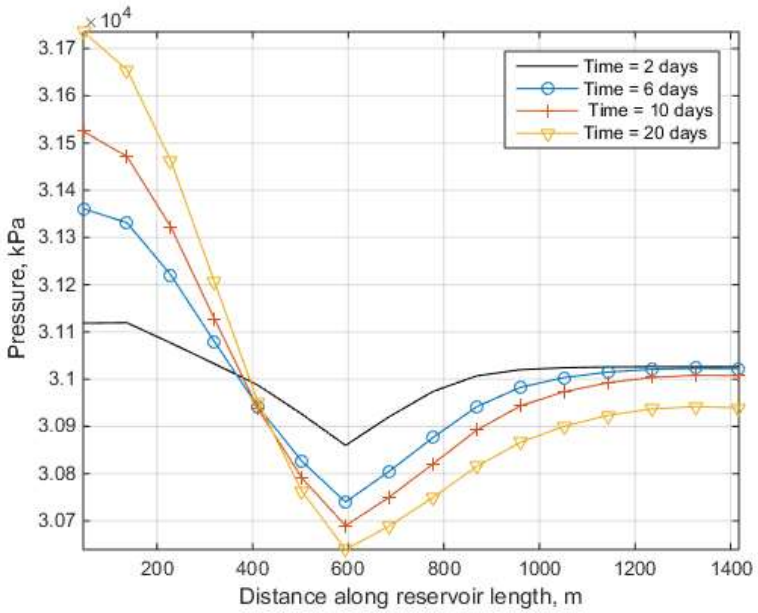


**Figure 6.8 Pressure profile at centerline for $\gamma = 0.9$**

To get a clearer picture of the predicted pressure behavior, we investigated the pressure history at two suitable blocks in the same vicinity as shown in Figure 6.9. These blocks, 8 and 104, are located at the centerline of the reservoir.

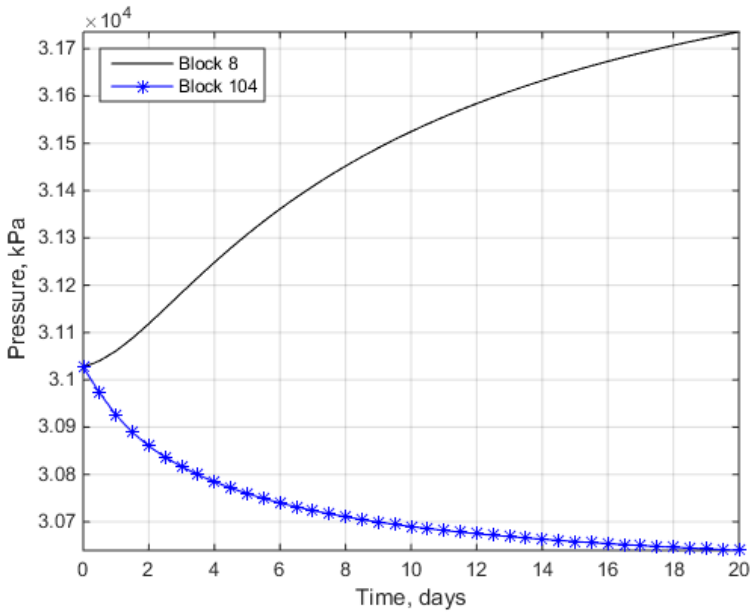


**Figure 6.9 Block pressure history for $\gamma = 0.9$**

Noting that initially, the pressure is constant in the whole domain. However, due to some reason(s) known (fluid influx from left-hand side), the pressure predicted at block 8 is always greater than at block 104. Logically, this is expected to be felt first in blocks adjacent the boundary, i.e. block 8, leading to the rapid increase in pressure compared to block 104. Again, the observed pressure drop in block 104 is attributed to the presence of the two wells in this vicinity, one of which is producing at the rate of 1.56 e-3 $m^3$/s (850 stb/d) and the other injecting at a rate of 1.2 e-3 $m^3$/s (650 stb/day). Based on the above reasons we expect the pressure predicted at block 8 to be higher than the other block, which is the case in Figure 6.9.

**Variation of $\eta$ with distance**. The variation of $\eta$ in x direction along the reservoir is shown in the surface plots presented in Figures 6.10 and 6.11 for different time intervals (0.5, and 20 days respectively). We observe that the distribution of $\eta$ is not uniform across the reservoir at each time interval. This can be attributed to the heterogeneous nature of the reservoir since this term is strongly dependent on permeability and time.

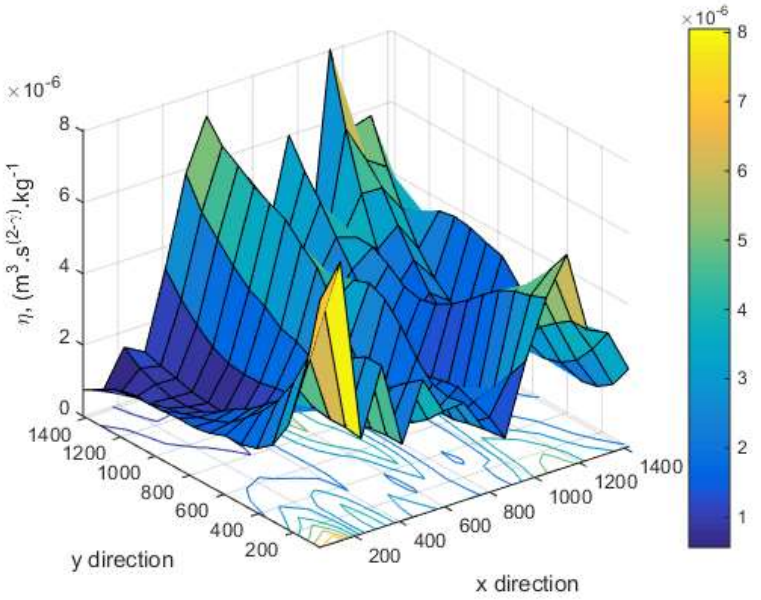


**Figure 6.10 Reservoir distribution of $\eta$ in the x direction at 0.5 days for $\gamma = 0.9$**

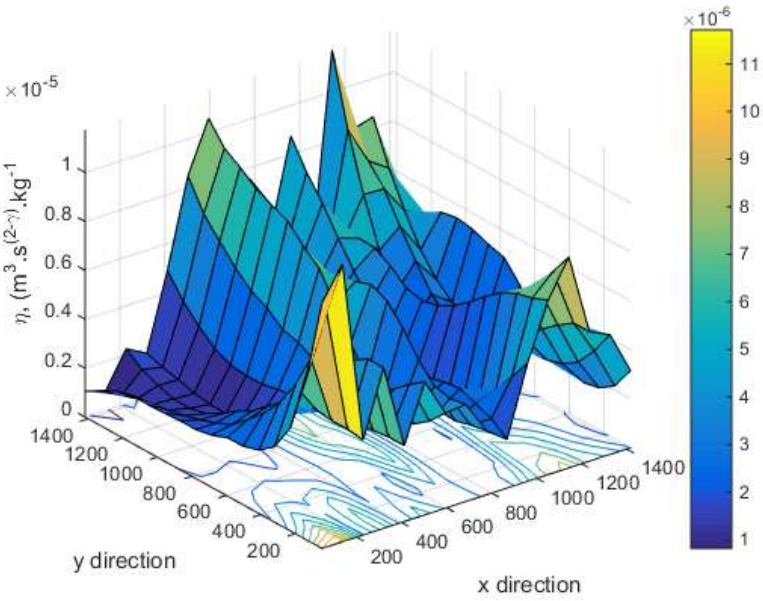


**Figure 6.11 Reservoir distribution of $\eta$ in the x direction at 20 days for $\gamma = 0.9$**

**Variation of wellbore pressure with time**. Based on the proposed memory-based model, the predicted wellbore pressures for all wells within the reservoir domain is presented in Figure 6.12.

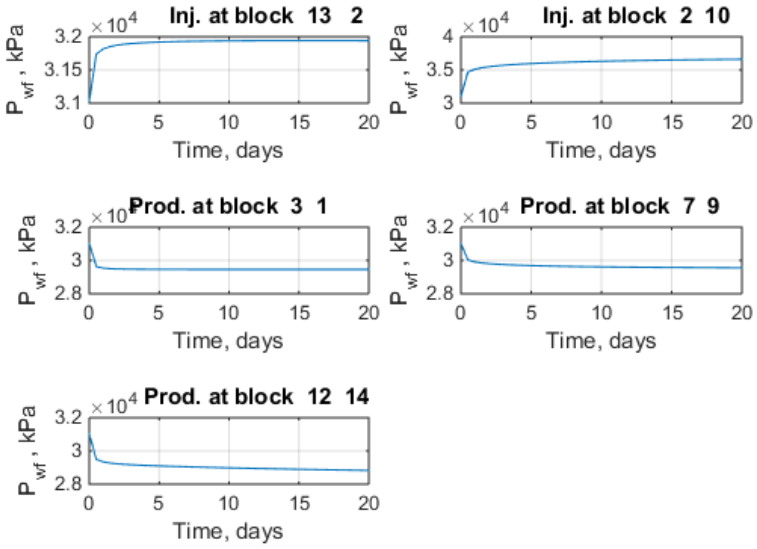


Figure 6.12 Bottom-hole pressure history for all wells in the domain

In all cases, an initial drop in wellbore pressure is observed in the production wells due to the drawdown needed to maintain the constant flow rate at the wellbore. However, due to the influx from the left side, and the effects of other injection wells we notice a reversal in the behavior of the curve. That is an increase in the wellbore pressures is observed as a result of the increasing block pressures. Hence, to maintain the constant production rate the wellbore pressure keeps increasing. The rate at which this change in wellbore pressure occurs is related to how fast the combined effect of the influx and the injector well is felt. On the other hand, the injection wellbore pressure is observed to undergo an initial rapid increase. However, the rate of increase drops with time and almost approaches a plateau.

It seemed logical to further investigate the effect of the fractional order term $\gamma$. However, due to space restrictions we will only investigate its influence on the block pressures and wellbore pressure.

**Effect of $\gamma$ on block pressure.** To investigate the significance of the fractional order derivative on the pressure profile, four different values of the memory exponent were employed to predict the pressure history in a block in the reservoir. Figure 6.13 presents the pressure history in blocks 8 and 104 respectively in the reservoir.

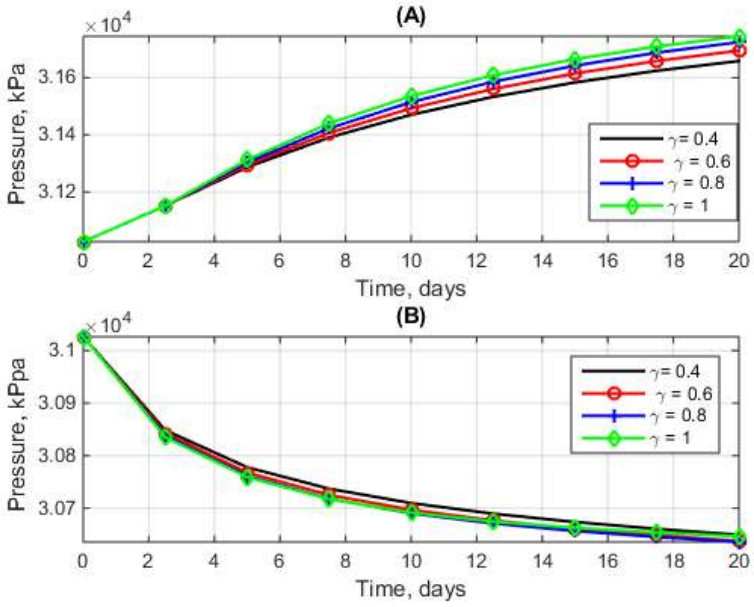


**Figure 6.13 Block pressure history for different values of memory exponent**

From the above figures, we can notice that the magnitude of the memory exponent on the block pressures is not so significant although some differences exists. Consider Fig 6.13 (a) for instance, we notice that as the magnitude of the memory exponent increases so does the magnitude of the predicted block pressure. The maximum pressure difference at the end of the simulation at block 8 is merely around 90 kPa (13 psia). On the other hand, at location 104, the pressure history for different values of the memory exponent are almost identical in magnitude except for the case of memory exponent equal 0.4. The incremental material balance check for the different values of memory exponent considered are presented in Figure 6.14 to verify mass conservation at each time step.

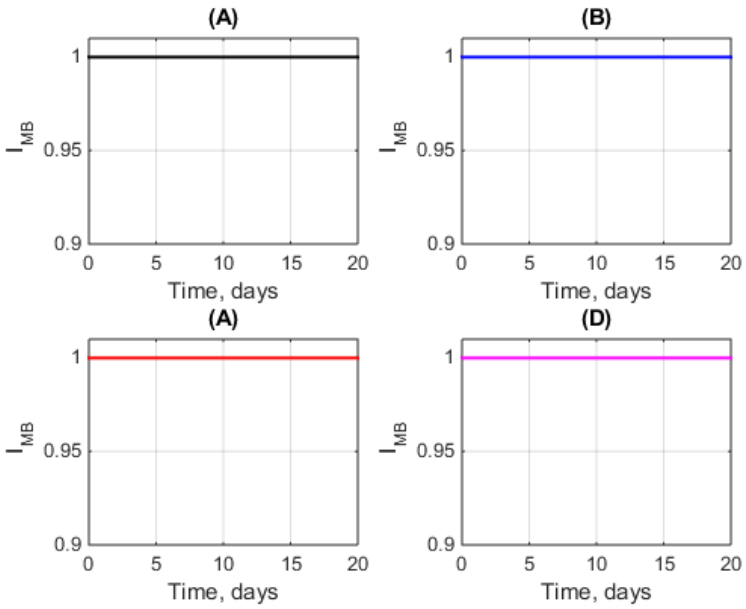


**Figure 6.14 Incremental material balance check for $\gamma$ = (A) 1, (B) 0.8, (C) 0.6, and (D) 0.4**

**Effect of $\gamma$ on wellbore pressure.** One of the most valuable reservoir management data comes from the measured wellbore pressures in giant reservoirs. A major application is in well testing where the measured pressures are usually used to estimate reservoir parameters. Therefore, it seemed proper to investigate how the memory exponent affects the measured wellbore pressure. The wellbore pressure history at two different locations in the reservoir is presented in Figure 6.15 for different values of the memory exponent employing the wellbore flow model derived in section 3.2

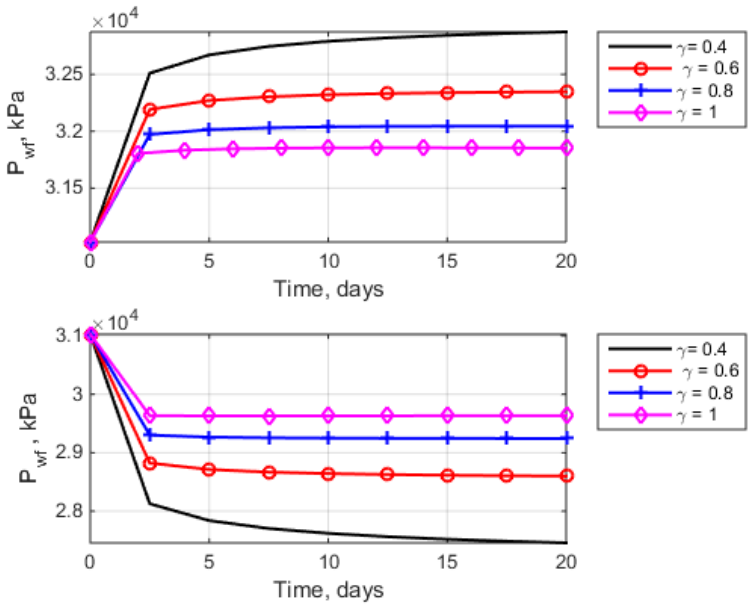


**Figure 6.15 Wellbore pressure history at location (a) 194, and (b) 33**

Figure 6.15 suggest that the memory exponent is an important input variable for the memory based approach to modelling fluid flow in a porous medium. Figure 6.15(a) which presents the injector wellbore pressure situated at block 194 suggest that as the memory exponent increases the wellbore flowing pressure decreases in magnitude. A maximum difference of around 1200 kPa (175 psia) was observed after 20 days for values of memory exponent of 1 and 0.4. Figure 6.15 (b) depicts the wellbore pressure for the producer well located at block 33, the results suggest that the predicted wellbore pressure decreases as the memory exponent decreases. In this case as much as 2200 kPa (320 psia) was observed this may be due to other reasons other than the memory exponent for example due to influx at the reservoir boundaries.

The proposed explicit numerical scheme suffers from time step restriction and thus not suitable for practical real world applications. In the next section, we develop a stable implicit numerical scheme to overcome the above mentioned limitation.

## 6.2 Implicit finite difference approximation

In this section we present an implicit finite difference numerical scheme to the isothermal memory based flow model i.e. Eq. (4.1). The Noorudin and Hossain [193] permeability porosity correlation is employed for the definition of the composite variable $\eta$. Furthermore, a stability analysis is presented to investigate the stability condition of the proposed the numerical scheme. Finally, we show through different case studies the effect of the memory exponent on reservoir pressure, porosity, wellbore pressure and the composite variable "$\eta$".

### 6.2.1 Definition of composite variable

As mentioned earlier a key parameter in the memory based models is the composite term "$\eta$", which is strongly dependent on the particular rock-fluid system considered. Therefore, proper definition of the type of reservoir fluid and formation properties is essential to obtaining accurate solution [127].

To solve the fractional model above Eq. (4.7), the permeability-porosity correlation by Noorudin and Hossain [193] was employed. Following the approach in Hossain et *al.* [127], the composite variable can be defined by Eq. (6.29).

$$\eta = \beta_c \frac{[K(FZI,\phi)]}{\mu(p)} t^{1-\gamma} \text{ , where } K = \left(\frac{FZI^2}{10\times 10^{11}}\right)\frac{\phi^{2m+1}}{(1-\phi)^2} \tag{6.29}$$

Combining Eqs. (6.3) to (6.10) with Eq. (6.29), gives the complete form of the isothermal memory-based flow model to be solved:

$$\frac{\partial}{\partial x}\left\{\frac{\left(\frac{1}{f_g a^2 S_{Vgr}^2}\right)\frac{\phi^{2m+1}}{(1-\phi)^2}}{\mu_{ob}\, e^{c_{\mu p}(p-p_b)}}\frac{A_x \beta_c t^{1-\gamma}}{B_{ob}\, e^{-c_o(p-p_b)}} D_t^{1-\gamma}\left(\frac{\partial p}{\partial x}\right)\right\}\Delta x + q_{sc}$$
$$= \frac{V_b \phi_o (c_s + c_o)}{B_{ob}\alpha_c \Delta t}(p^{n+1} - p^n) \tag{6.30}$$

### 6.2.2 Discretized Memory-Based Pressure Equation

The final form of the discretized pressure equation for the isothermal case is presented below:

$$-T_{i-\frac{1}{2}}^{n+1}\,P_{i-1}^{n+1}+\left(T_{i-\frac{1}{2}}^{n+1}+T_{i+\frac{1}{2}}^{n+1}+C_i^{n+1}\right)P_i^{n+1}-T_{i+\frac{1}{2}}^{n+1}\,P_{i+1}^{n+1}$$
$$=C_i^{n+1}P_i^n+{q_{sc,i}}^{n+1}+\sum_{k=1}^{t/\Delta t}\psi(\gamma,k)\delta_i^{n+1-k} \tag{6.31}$$

Where $\delta_i^{n+1-k}$ is defined as previously, the fluid transmissibility $T$ and pseudo-compressibility $C$ introduced above are defined as follows:

$$T_i^{n+1}=\frac{\beta_c A_{x,i}n_i^{n+1}}{\Delta x B_{o,i}^{n+1}(\Delta t)^{1-\gamma}},\ \text{and}\ C_i^{n+1}=\frac{V_b\phi_o(c_s+c_o)}{B_{ob}\alpha_c\Delta t} \tag{6.32}$$

The term $T_{i\pm\frac{1}{2}}$, refers to the harmonic average between two adjacent grid blocks for the case of isothermal fluid flow.

### 6.2.3 Stability analysis

Numerical schemes are usually plagued with stability and consistency issues due to either the nature of the discretization method or truncation error due to the Taylor series approximation of the PDE in our case fractional PDE. In this section we show through Fourier series expansion that the above proposed scheme is unconditionally stable.

To begin we define the error ($\epsilon_i^n$) resulting from the approximate nature of the finite difference scheme by:

$$\epsilon_i^n=\left|P_{i,exact}^n-P_i^n\right| \tag{6.33}$$

$$\epsilon_i^{n+1}=\left|P_{i,exact}^{n+1}-P_i^{n+1}\right| \tag{6.34}$$

Noting that if

$$\varsigma=\left|\frac{\epsilon_i^{n+1}}{\epsilon_i^n}\right|\leq 1,\ \text{the scheme is unconditionally stable.} \tag{6.35}$$

While if,

$$\varsigma = \left|\frac{\epsilon_i^{n+1}}{\epsilon_i^n}\right| > 1, \text{ the numerical scheme is unstable.} \tag{6.36}$$

Therefore, we investigate in the next few pages if the proposed implicit numerical scheme Eq. (6.31) satisfies Eq. (6.35).

Employing Fourier series expansion we introduce the following;

$$\epsilon_i^n = \left|P_{i,exact}^n - P_i^n\right| = \varepsilon_k^n e^{j(\beta_k i\Delta x)} \tag{6.37}$$

Where $j = \sqrt{-1}$, $\varepsilon_k$ is the amplitude of the $k^{th}$ harmonic, $\beta_k = \frac{k\pi}{l}$, and $l$ being the length of the interval through which the function is defined.

Likewise, the transmissibility and pseudo compressibility are taken as constants in Eq. (6.31) in order to reduce the nonlinearity of the equation. Thus Eq. (6.31) simplifies to:

$$P_i^{n+1} - P_i^n =$$

$$\bar{S}\sum_{k=0}^{n}\psi(\gamma,k)\left\{P_{i+1}^{n+1-k} - 2P_i^{n+1-k} + P_{i-1}^{n+1-k}\right\} \tag{6.38}$$

Where $\bar{S} = \frac{T}{C(\Delta t)^{1-\gamma}}$

Now substituting Eq. (6.37) into Eq. (6.38) results in

$$\epsilon_i^{n+1} - \epsilon_i^n = \bar{S}\left[\sum_{k=0}^{n}\psi(\gamma,k)\left\{\epsilon_{i+1}^{n+1-k} - 2\epsilon_i^{n+1-k} + \epsilon_{i-1}^{n+1-k}\right\}\right] \tag{6.39}$$

Eq. (6.39) can be written as

$$e^{j(\beta_k i\Delta x)}(\varepsilon_k^{n+1} - \varepsilon_k^n) = \bar{S}\left[\sum_{k=0}^{n}\psi(\gamma,k)\left\{\varepsilon_k^{n+1-k}\left(e^{j\beta_k\Delta x} - 2 + e^{-j\beta_k\Delta x}\right)e^{j\beta_k i\Delta x}\right\}\right] \tag{6.40}$$

Further simplification and omitting subscripts for convenience yields

$$-4Sin^2\left(\frac{\beta\Delta x}{2}\right)[\bar{S}\{\varepsilon^{n+1} + \sum_{k=1}^{n}\psi(\gamma,k)\varepsilon^{n+1-k}\}] = \varepsilon^{n+1} - \varepsilon^n \tag{6.41}$$

Rearranging, we obtain

$$\left[1 + 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}\right]\varepsilon^{n+1} = \varepsilon^n - 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}\,[\sum_{k=1}^{n}\psi(\gamma,k)\varepsilon^{n+1-k}] \tag{6.42}$$

Noting that

$$\sum_{k=1}^{n}\psi(\gamma,k)\varepsilon^{n+1-k} = \sum_{k=0}^{n-1}\psi(\gamma,k+1)\,\varepsilon^{n-k} \tag{6.43}$$

Thus

$$\sum_{k=1}^{n} \psi(\gamma,k)\varepsilon^{n+1-k} = \psi(\gamma,1)\,\varepsilon^{n} + \sum_{k=1}^{n-1} \psi(\gamma,k+1)\varepsilon^{n-k} \tag{6.44}$$

Therefore

$$\sum_{k=1}^{n} \psi(\gamma,k)\varepsilon^{n+1-k} = \begin{cases} \psi(\gamma,1)\varepsilon^{n} & n=1, \\ \psi(\gamma,1)\,\varepsilon^{n} + \sum_{k=1}^{n-1} \psi(\gamma,k+1)\varepsilon^{n-k} & n \geq 2. \end{cases} \tag{6.45}$$

Combining Eqs. (6.42) and (6.45) for n=1 gives

$$\left[1 + 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}\right]\varepsilon^{n=2} = \varepsilon^{n=1} - 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}\psi(\gamma,1)\,\varepsilon^{n=1} \tag{6.46}$$

Simplifying further

$$\left[1 + 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}\right]\varepsilon^{n=2} = \left[1 - 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}\,\psi(\gamma,1)\right]\varepsilon^{n=1} \tag{6.47}$$

Hence

$$\varepsilon^{n=2} = \frac{1-4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}\,\psi(\gamma,1)}{1+4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}}\;\varepsilon^{n=1} \tag{6.48}$$

Therefore Eq. (6.48) implies $|\varepsilon^{n=2}| \leq |\varepsilon^{n=1}|$

For the conditions when $n \geq 2$, Eqs. (6.42) and (6.45) result in

$$\left[1 + 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}\right]\varepsilon^{n+1} = \varepsilon^{n} - 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}\left[\psi(\gamma,1)\,\varepsilon^{n} + \sum_{k=1}^{n-1}\psi(\gamma,k+1)\varepsilon^{n-k}\right] \tag{6.49}$$

Re-arranging Eq. (6.49)

$$\varepsilon^{n+1} = \frac{1-4\bar{S}\psi(\gamma,1)Sin^2\left(\frac{\beta\Delta x}{2}\right)}{1+4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}}\;\varepsilon^{n} - \left\{\frac{4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}}{1+4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}}\right\}\sum_{k=1}^{n-1}\psi(\gamma,k+1)\varepsilon^{n-k} \tag{6.50}$$

Suppose that $|\varepsilon^{k}| \leq |\varepsilon^{n=1}|$ for $2 \leq k \leq n$ and we want to show that $|\varepsilon^{n+1}| \leq |\varepsilon^{n=1}|$

$$\varepsilon^{n+1} \leq \frac{1-4\bar{S}\psi(\gamma,1)Sin^2\left(\frac{\beta\Delta x}{2}\right)}{1+4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}}|\varepsilon^{n}| - \left\{\frac{4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}}{1+4Sin^2\left(\frac{\beta\Delta x}{2}\right)}\right\}\sum_{k=1}^{n-1}\psi(\gamma,k+1)|\varepsilon^{n-k}| \tag{6.51}$$

$$\leq \left[\frac{1-4\bar{S}\psi(\gamma,1)Sin^2\left(\frac{\beta\Delta x}{2}\right)}{1+4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}} - \left\{\frac{4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}}{1+4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}}\right\}\sum_{k=1}^{n-1}\psi(\gamma,k+1)\right]|\varepsilon^{n=1}|$$

Therefore

$$\leq \left[ \frac{1 - 4\bar{S}\{\psi(\gamma,1) + \sum_{k=1}^{n-1}\psi(\gamma,k+1)\}Sin^2\left(\frac{\beta\Delta x}{2}\right)}{1 + 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}} \right] |\varepsilon^{n=1}|$$

Simplifying further results in

$$= \left[ \frac{\left[1 - 4\bar{S}\psi(\gamma,1)Sin^2\left(\frac{\beta\Delta x}{2}\right)\right]}{1 + 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}} \right] |\varepsilon^{n=1}|$$

Finally we have

$$\left[ \frac{\left[1 - 4\psi(\gamma,1)\bar{S}Sin^2\left(\frac{\beta\Delta x}{2}\right)\right]}{1 + 4Sin^2\left(\frac{\beta\Delta x}{2}\right)\bar{S}} \right] |\varepsilon^{n=1}|$$

Therefore we can conclude that

$$\varepsilon_k^{n+1} \leq |\varepsilon_k^{n=1}| \tag{6.52}$$

### 6.2.4 Numerical simulation

To test the finite difference scheme (Eq. 6.31), we consider a reservoir of length 1463.04 m, width 91.44 m, and height of 30.48 m. Given the difficulty of measuring rock properties, it is common to use geostatistical methods to make realizations of porosity and permeability. In this study, we will generate the porosity $\phi$ values as a Gaussian field. As a simple approximation to a Gaussian field, we generate a field of independent normally-distributed porosity values as presented in **Figure 6.16**. To get a crude approximation to the porosity-permeability- relationship, we assume that our medium can be characterized with a *FZI* = 10 $\mu$m and $m$ = 2.1. Using Eq. (6.29), the calculated isotropic permeability is presented in **Figure 6.17**.

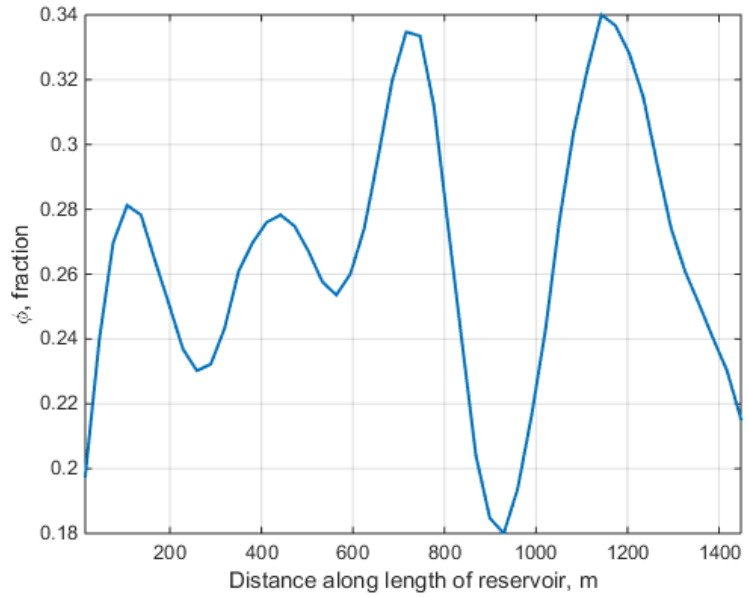


**Figure 6.16 Reservoir domain with randomly generated initial porosity**

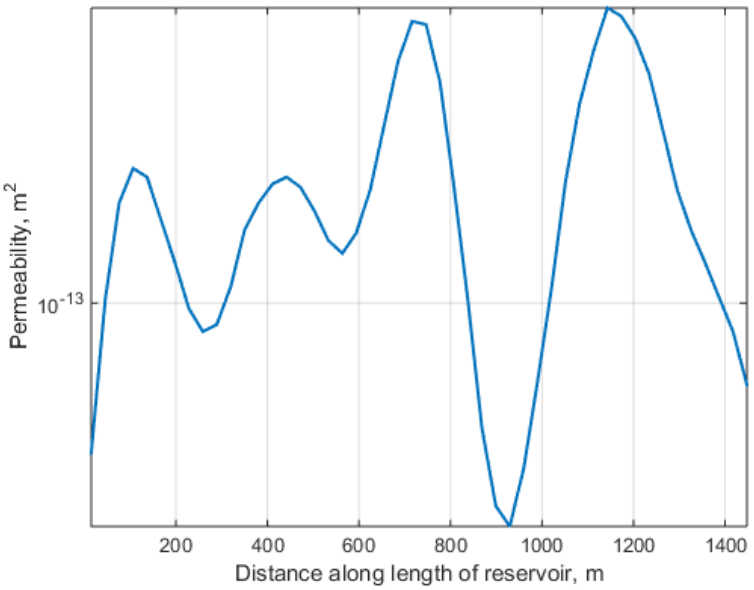


**Figure 6.17 Initial Reservoir permeability distribution**

The properties of the crude oil used for this computation is presented in Table 6.3. Due to the uncertainty of a representative value for the fractional order derivative we consider for case study a fractional order between 0.4 and 1. The grid dimension and time step employed are $\Delta x$ = 30.48 m (100 feet), $\Delta t$ = 5 days and t = 200 days. To solve this flow problem Eq. (6.11) is implemented with MATLAB. To solve this equation, however, $P(x,0) = 51710.68\ kPa$ ( $7500\ psia$) is applied as an initial condition.

**Boundary conditions**: Specified rate at the internal boundary, constant pressure at the external boundary.

**Table 6.3 Reservoir parameters and boundary conditions**

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $FZI = \frac{1}{\sqrt{f_g}\, aS_{V_{gr}}}$ | 10$\mu$m | $R_s$ | 23 $m^3/m^3$ |
| $m$ | 2.1 | $\gamma_g$ | 0.748 |
| $API$ | 31 | $c_s$ | 5.08e-6 $kPa^{-1}$ |
| $Temp$ | 366 K | $c_\mu$ | 1.22e-9 $kPa^{-1}$ |
| $Q_{x0} = 6.44 \times 10^{-4}\ m^3/sec$ | $P_{xL} = 51710.68\ kPa$ | $\Delta t = 432000\ sec$ | |

To implement the boundary conditions, Eq. (6.31) is modified as required.

### 6.2.5 Discussion of simulation results

Three hypothetical cases have been tested, with results presented herein.

**Case 1**: Model consistency check with and without production wells present in the reservoir with constant pressure maintained at the right boundary.

The proposed memory model was checked for consistency with the classical fluid flow model in order to ascertain the accuracy of the proposed scheme. From Eq. (4.2), when $\gamma = 1$, the Grunwald-Letnikov operator reduces to the identity operator. Therefore, Eq. (4.2) simplifies to:

$$u = -\frac{\beta_c[K(FZI,\phi)]}{\mu(p)}\left(\frac{\partial p}{\partial x}\right) \tag{6.53}$$

Following this knowledge, the results from both models at two different cell locations are presented in **Figure 6.18**. Clearly, the pressure profiles predicted for the total length of simulation time were equal. Next, to further ascertain the accuracy of the memory model prior to further investigations, a producer well is introduced at block 5. The predicted

wellbore pressures for both the memory based model and classic flow model is shown in **Figure 6.** 19

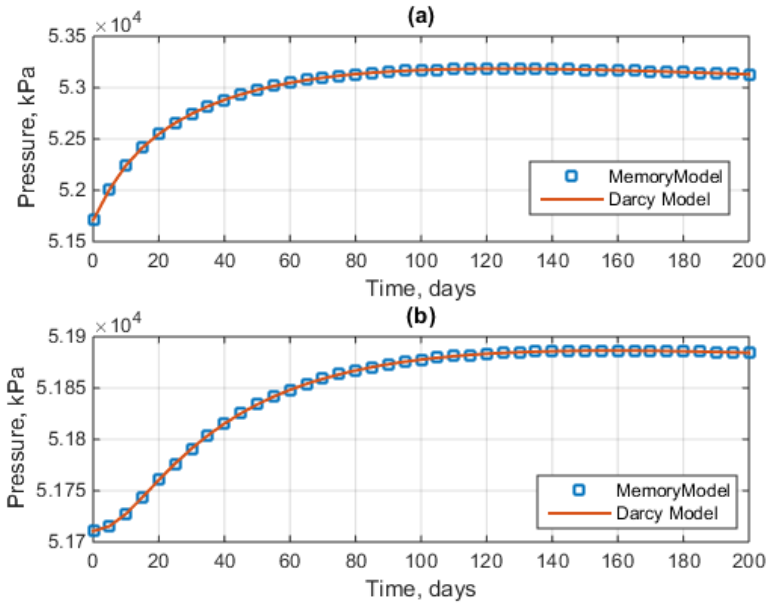


**Figure 6.18 Case 1 - Memory model consistency with Classic Model at (a) Block 10 (b) Block 45**

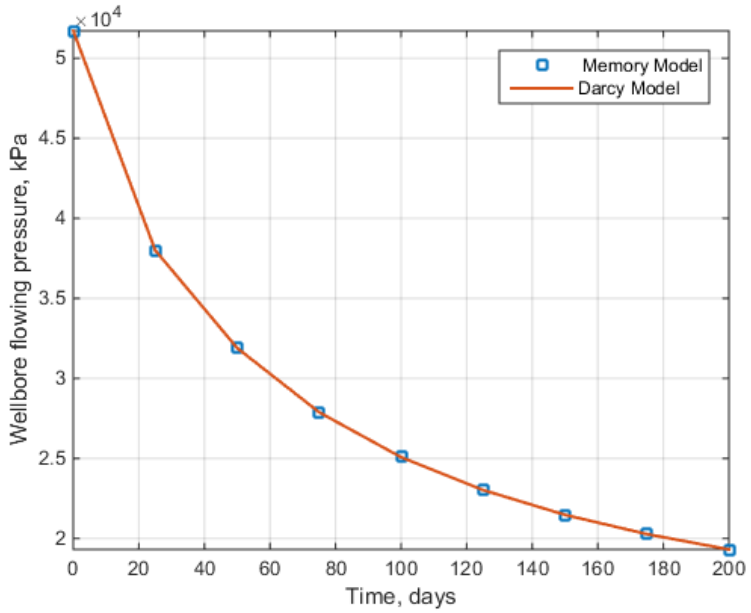


**Figure 6.19 Case 1 - Memory model consistency with Classic Model based on Producer well located at Block 5**

As can be observed from **Figure 6.19**, the proposed memory model indeed reduces to the classic Darcy's model when $\gamma = 1$, since in both plots above, the pressure predicted are identical.

**Case 2**: Effect of Fractional order derivative term on reservoir pressure

In this section, the pressure variation in the reservoir for different values of $\gamma$ in the newly proposed flow model was investigated. Similar to case 1, a specified flux at a rate of 6.44e-4 $m^3/s$ (350 bbl/day) is maintained at the left boundary. In addition, no producer or injector wells were present within the reservoir domain. **Figure 6.20** presents the predicted block pressure profiles based on the new memory-based flow equation for different values of $\gamma$. The developed numerical scheme has been found to be unconditionally stable, i.e. regardless of the value of $\gamma$, and $\Delta t$, consistent and, accurate numerical results are always obtained.

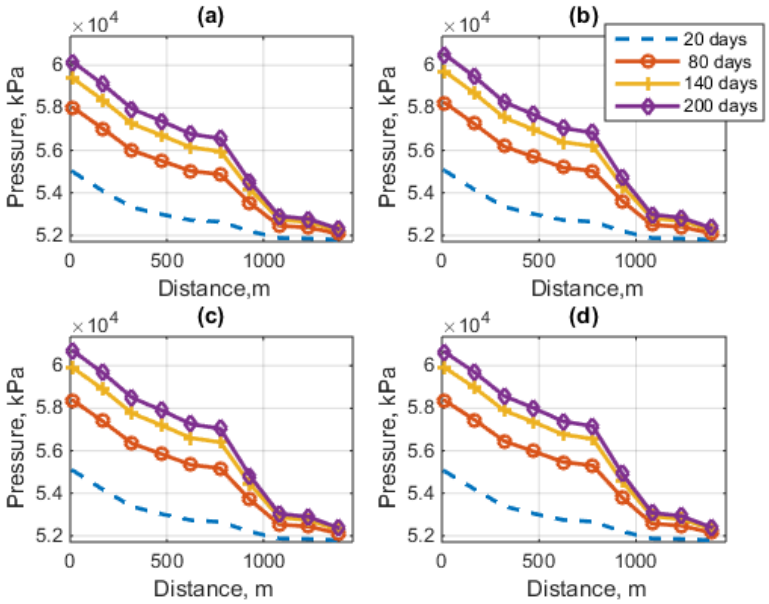


**Figure 6.20 Case 2 - Pressure profile across reservoir length for (a) $\gamma$=0.4, (b) $\gamma$= 0.6, (c) $\gamma$=0.8 and (d) $\gamma$= 1**

A first glance at the pressure profile in **Figure 6.20** suggests that the magnitude of the memory exponent has little or no influence on the predicted block pressures within the reservoir. On the other hand, a closer look proves otherwise. At any instant in time, the block pressure predicted is higher as the value of $\gamma$ increases. For instance, when the term $\gamma = 0.4$, after 200 days of simulation the pressure at the first cell is approximately 6e4 kPa, however for the case of $\gamma = 1$, the pressure at the same location is around 6.09e4 kPa. This difference in predicted reservoir pressure (900 kPa/ 130 psia) can be attributed to the memory formalism.

Further investigations on the effect of $\gamma$ on reservoir pressure at certain time intervals during the simulation was carried out. Due to space restrictions, only the results of the pressure along the reservoir after 20, 80, 140 and 20 days for values of $\gamma$ is presented (see **Figure 6.21**).

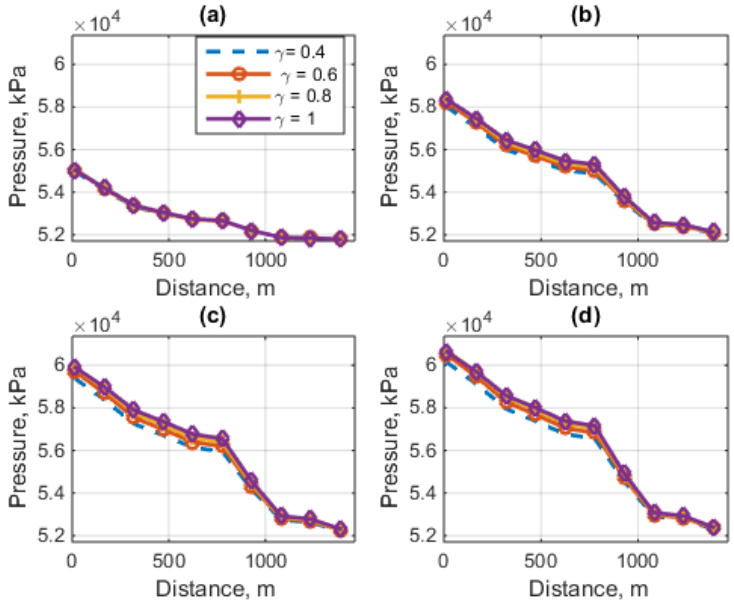


**Figure 6.21 Case 2 - Pressure profile along the reservoir at different times for several values of $\gamma$ at (a) 20 days (b) 80 days (c) 140 days and (d) 200 days.**

Clearly from **Figure 6.21**, we conclude that as the fractional order of derivative increases the block pressures predicted across the reservoir length increases although very slightly and vice-versa. To further establish the accuracy of the simulator result a material balance check is also performed at every time step using a modified form of the conventional incremental material balance check proposed by Abou-Kassem [180].

$$I_{MB} = \frac{\sum_{m=1}^{M} \frac{V_{bm}}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_o}\right)_m^{n+1} - \left(\frac{\phi}{B_o}\right)_m^{n}\right]}{\sum_{m=1}^{M}\left(q_{sc,m}^{n+1} + \sum_{l \epsilon \xi_m} q_{sc,m}^{n+1} + \sum_{m=1}^{M} memory\right)} \tag{6.54}$$

Figure 6.22 presents the results from the material balance check every time step for the different values of the memory exponent employed for the case study. In all cases, the incremental material balance equal one showing that material balance is preserved for all cases.

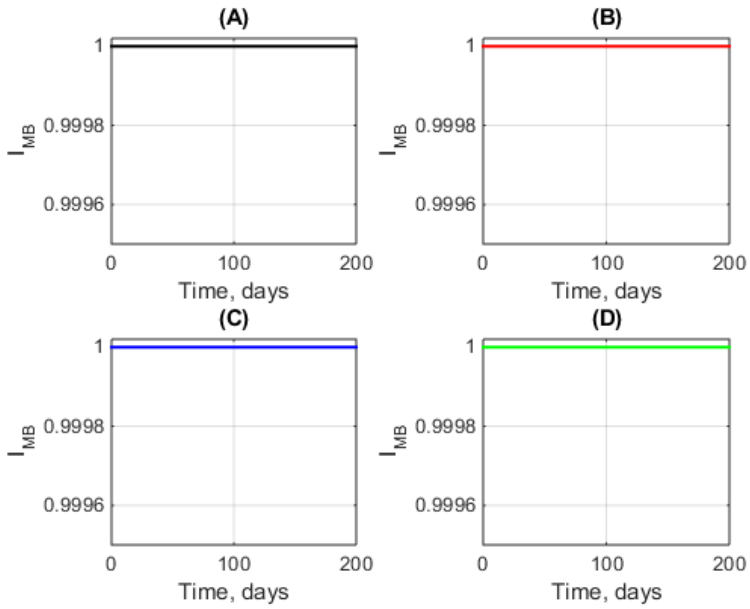


Figure 6.22 Incremental material balance check for $\gamma =$ (A) 0.4, (B) 0.6, (C) 0.8, and (D) 1

**Case 3:** Producer and Injectors distributed in the reservoir.

In this section, we introduce a producer and an injector into the reservoir domain. For simplicity, both wells are operated under constant flow rate for the whole simulation runtime. The location and distribution of the wells within the reservoir is presented in **Table 6.4**.

Our aim is to compare the porosity values predicted at these blocks using the classic fluid flow model and our proposed memory-based model based for $\gamma$ = 0.6 and 0.8.

Table 6.4 Distribution of producers in the domain

| Position | Rate ($m^3/s$) |
|---|---|
| 1 | -2.86e-3 |
| 24 | 8.828e-4 |

**Porosity evolution**: The evolution of porosity at blocks 1 and 24 are presented in **Figure 6.23**. Here, the porosity change with pressure at the block location using the classic fluid model, and the memory-based model for $\gamma$ = 0.8 and 0.6 are compared.

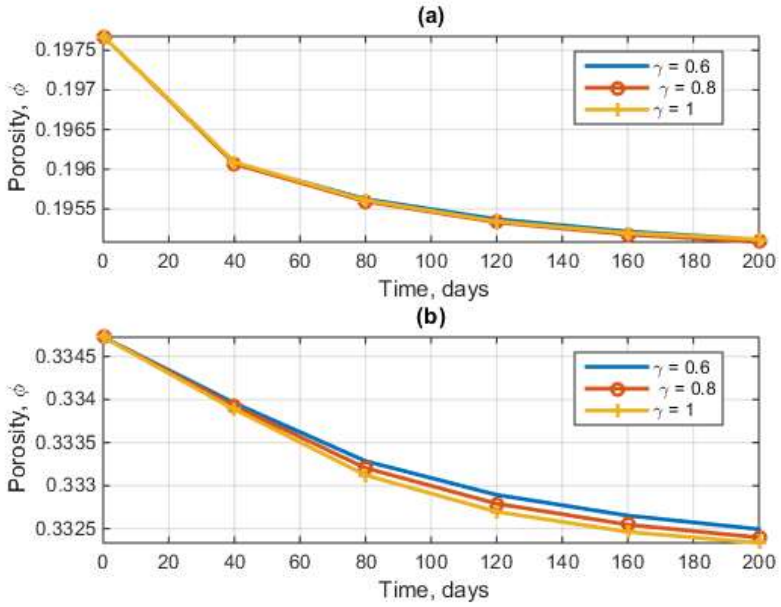


**Figure 6.23 Case 3 – Porosity evolution for (a) Block 1 and (b) Block 24**

The results show that the porosity values predicted in all models do not differ significantly. Figure 6.23(b) reveals that the porosity reduction predicted becomes more severe with increasing value of the fractional order term. It must be mentioned that Blocks 1 and 24 were chosen for study because both blocks observe the biggest change in pressures due to the wells located at both locations. In both blocks, the porosity value is decreasing with time due to the pressure decline across the reservoir resulting from the oil produced by block 1. Furthermore, the little change in porosity values is not strange, since porosity is generally known to be slightly dependent on pressure.

**Variation of velocity with time**: The velocity history in block 1 and 24 are presented in **Figure 6.24** for different values of fractional order derivative. As can be seen in the figure, the fractional order derivative term has minimal effect on the magnitude of velocity in these blocks and thus in the reservoir in general. Take for instance in Figure 6.24 (a), after 40 days for all values of $\gamma$ considered, the magnitude of velocity predicted has reached steady state conditions and is approximately $1.5\times 10^{-10}$ m/sec.

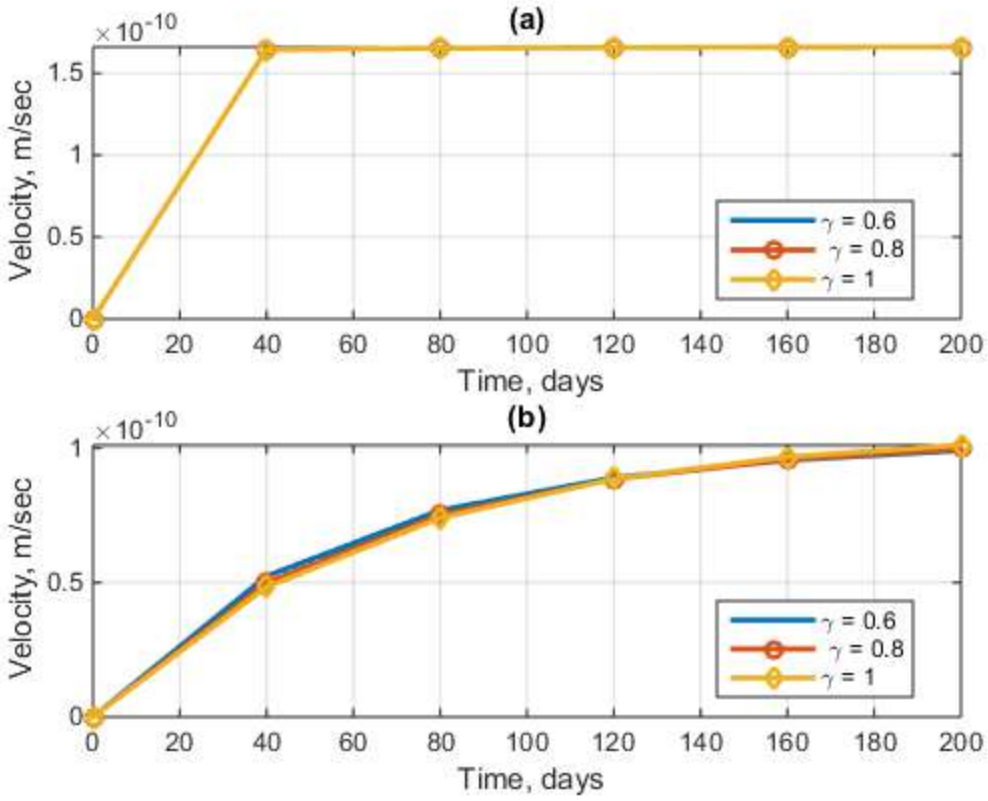


Figure 6.24 Case 3 - Velocity history at (a) Block 1 and (b) Block 24

**Variation of wellbore pressure with time.** The wellbore pressure in Block 1 and 24 are shown in **Figure 6.25** for different values of $\gamma$. Results suggest that the magnitude of wellbore pressure is slightly dependent on the magnitude of $\gamma$. In both blocks, the wellbore flowing pressure continually reduces with time. More importantly, the results predicts slightly lower wellbore pressures as the memory exponent increases in both cases.

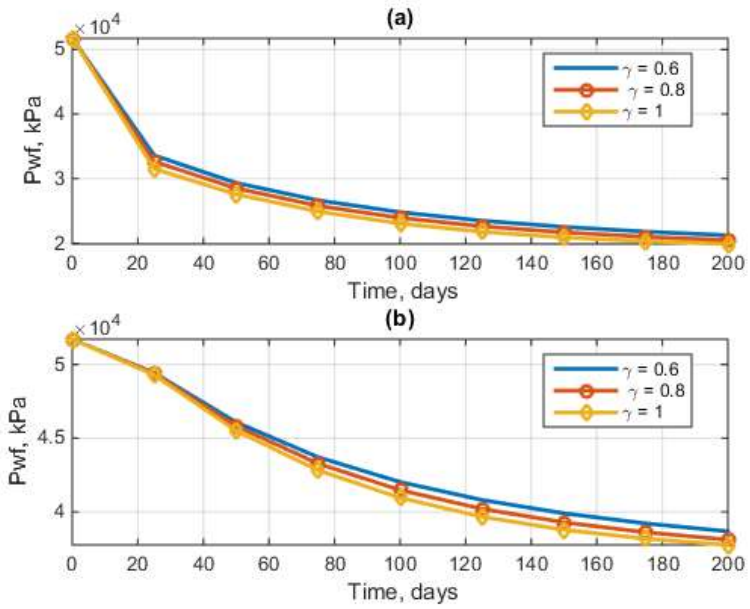


Figure 6.25 Wellbore pressure history (a) Block 1 and (b) Block 24

**Variation of $\eta$ with distance**. The variation of $\eta$ along the reservoir toward the outer boundary is shown in **Figure 6.26** at different time intervals for $\gamma = 0.8$. The memory function captures the evolution of rock, and fluid properties with time. The heterogeneous nature of the reservoir makes it difficult to describe the behavior of the memory variable. However, we notice in some blocks (with high permeable values) significant change in the magnitude of the memory function with due to changes in block pressure with time. Furthermore, as mentioned earlier the memory function strongly depends on the rock type and fluid properties hence proper rock and fluid characterization is essential for our proposed memory-based model.

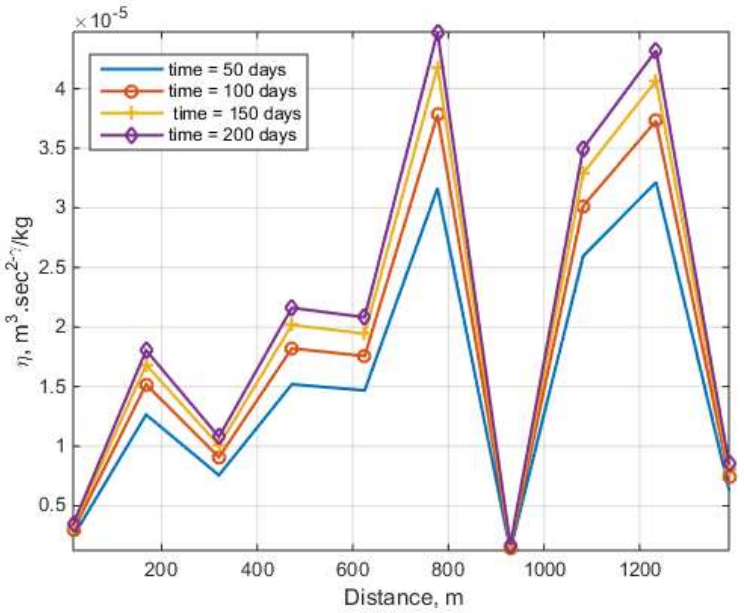


**Figure 6.26 Case 4 - Variation of $\eta$ along the reservoir for $\gamma$= 0.8**

Next, the magnitude of the proposed memory function $\eta$ is compared with its analogous form in the classic fluid model. **Figure 6.27**, presents the variation of $\eta$ along the reservoir based on the proposed memory-based model for $\eta = 0.8$ and the classic fluid model at 100 and 200 days. We observed on the average about 20% difference between both models with greater values for $\eta$ predicted in the memory-based model. This disparity contributes to some of reported differences observed using the memory-based model as opposed to the classic model. For example the early arrival of tracer observed in the field [197], or the delayed influence of pressure at the boundary reported by Caputo [46].

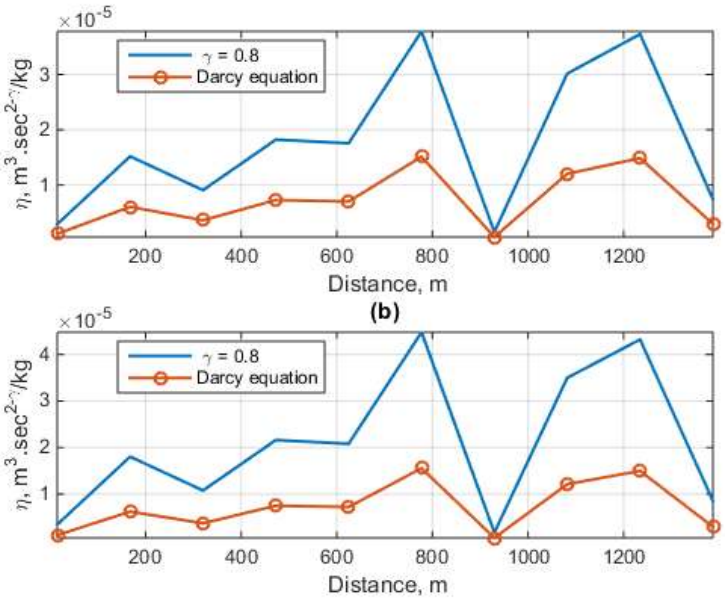


**Figure 6.27 Case 4 - Variation of $\eta$ along the reservoir after 100 and 200 days for the proposed memory model ($\gamma = 0.8$) and the Darcy flow model.**

# CHAPTER 7

# RESULTS AND DISCUSSION

## 7.1 Non-isothermal Numerical Simulators- LTE

In this subsection, the results obtained from runs on the non-isothermal simulation models developed for HFI process under local thermal equilibrium assumption are presented. We begin by validating both numerical models with the analytical solution of the same problem. Then, the observations made when the alteration of rock properties are considered and not considered in the Boussinesq-based models are presented in terms of temperature profile and thermal efficiency are discussed. Subsequently, a comparison between the temperature profile predicted by the modified Boussinesq-based simulator and the memory-based simulator are presented. Finally, a comparison between the classic Boussinesq model and the modified Boussinesq model are presented.

### 7.1.1 Simulator validation with analytical solution of simplified problem

It is usually necessary to validate the accuracy of numerical simulators either through experimental data or analytical solution if they exist. Most of the time however, analytical solutions of the simplified problems can always be derived by employing either the Laplace transform method, Fourier transform method or other similar mathematical tools. In this section, we employ the analytical solution validation approach by considering different number of cells and the data presented in Table 7.1.

The temperature distribution in a porous media can be described as follows [198]:

$$T = T_{\text{init}} + \frac{T_{\text{init}} - T_{\text{inlet}}}{2}\left[erfc\left(\frac{x-vt}{2\sqrt{D\times t}}\right) + e^{\frac{xv}{D}} \times erfc\left(\frac{x+vt}{2\sqrt{D\times t}}\right)\right] \tag{7.1}$$

Where,

$$D = \frac{\lambda_e}{(\rho c_p)_f} + \alpha_L v \tag{7.2}$$

$$v = \frac{u(\rho c_p)_w}{(\rho c_p)_b} \tag{7.3}$$

**Table 7.1 Validation data input**

| Property | Estimate |
|---|---|
| $\phi$ | 0.222 |
| $L$ | 0.3048 m |
| $d$ | 0.0508 m |
| $\rho_w$ | 1000 kg/m$^3$ |
| $\rho_s$ | 2643 kg/m$^3$ |
| $\alpha_L$ | 0 m |
| $T_{inlet}$ | 370 K |
| $T_{init}$ | 300 K |
| $K$ | 3.957469e-13 m$^2$ |
| $q_{inj}$ | 8.3333e-9 m$^3$/sec |
| $D$ | 7.63e-7 m$^2$/sec |
| $v$ | 6.33e-6 m/sec |

Figures 7.1 to 7.4 show that the numerical solution matches the analytical solution as the number of cells increases. The result shows that either 100 cells or 200 cells can be employed as they both give good enough match with the analytical solution. It must be noted that the variations of all rock and fluid properties were not considered in order to fit the analytical solution. This also explains why both numerical models predict the same temperatures. The memory exponent was considered to be equal to one so as to reduce to the classic solution case.

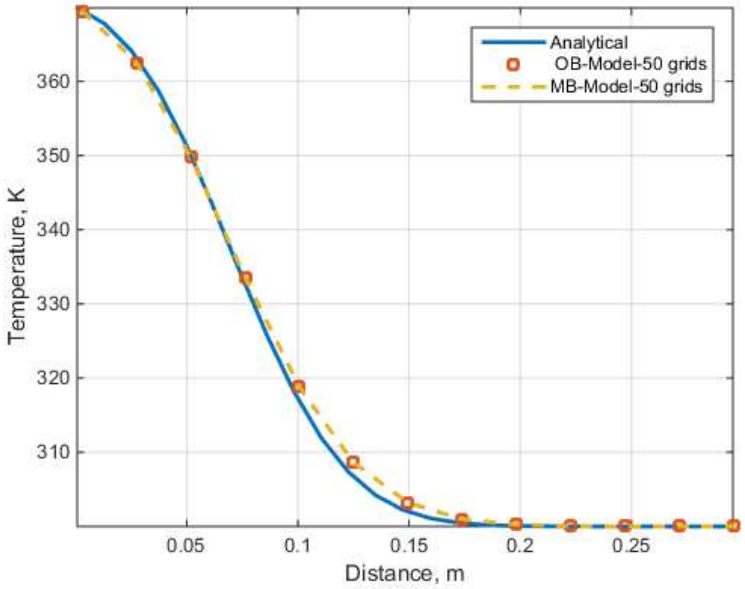


**Figure 7.1 Grid dependent study with 50 cells**

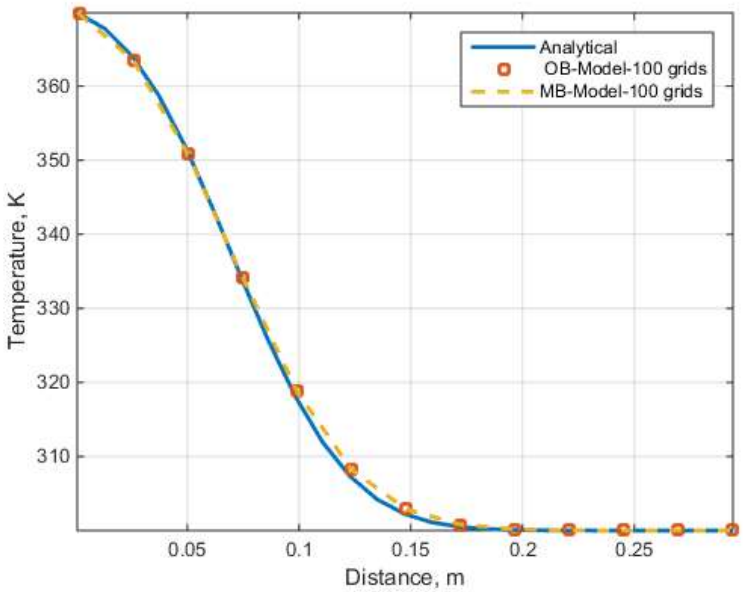


**Figure 7.2 Grid dependent study with 100 cells**

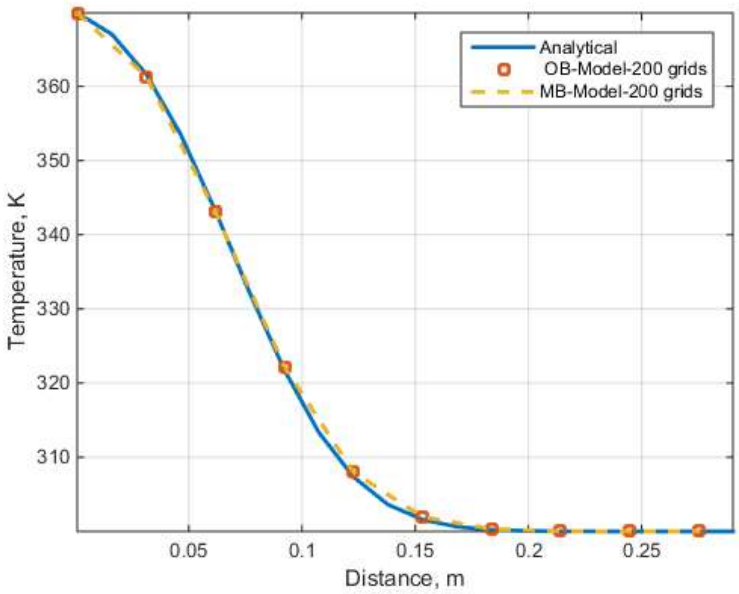


**Figure 7.3 Grid dependent study with 200 cells**

The temperature history at the inlet cell was compared with that predicted by the analytical solution to determine the optimum number of cell. Figure 7.4 clearly shows that using a 200 cells gives a good enough match with the analytical solution. Thus, 200 cells was selected for the rest of the comparison study.

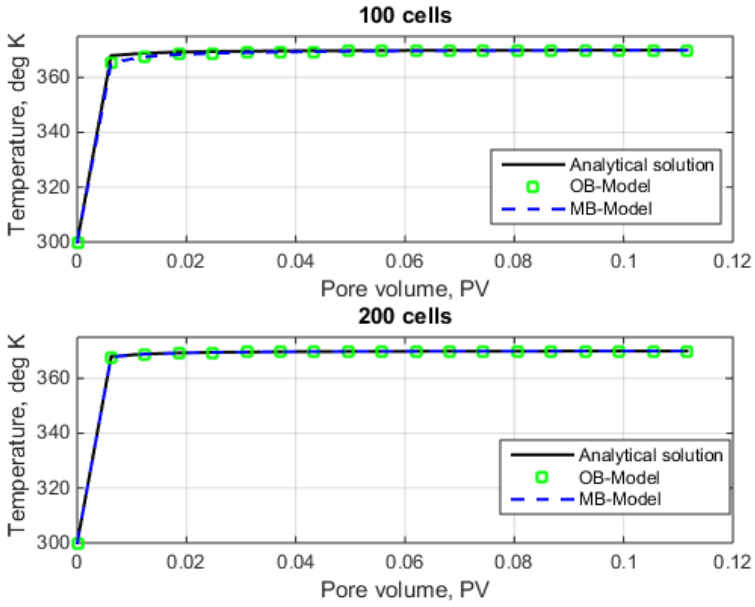


Figure 7.4 Model validation for optimum number of cells

### 7.1.2 Simulator verification with published experimental data

Furthermore, the experimental data generated by Arihara [179] for hot water flooding on Berea sandstone was employed for further verification. In this case, the heat loss to the surrounding, as well as variable fluid properties were considered. The porosity and permeability of the porous medium were considered to be constants. Table 7.2 present the properties of the Berea sandstone considered by the author for his experimental work.

Table 7.2 Physical properties of Berea sandstone core

| Property | Estimate |
|---|---|
| $L$ | 0.6 meter (23. 5 inches) |
| $D$ | 0.0508 m (2 inches) |
| $\phi$ | 0.222 |
| $K$ | 4e-13 (401 mD) |
| $T_{\infty}$ | 298.4278 K |
| $U$ | 8.51745 W/m$^2$K |

Figure 7.5 presents the injection temperature as a function of time in minutes which was used for his experimental runs. Subsequently, the measured temperature profile along the axial length obtained from the experimental study was compared with that obtained from non-isothermal simulators developed herein (see Figure 7.6) to verify the accuracy of the simulators.

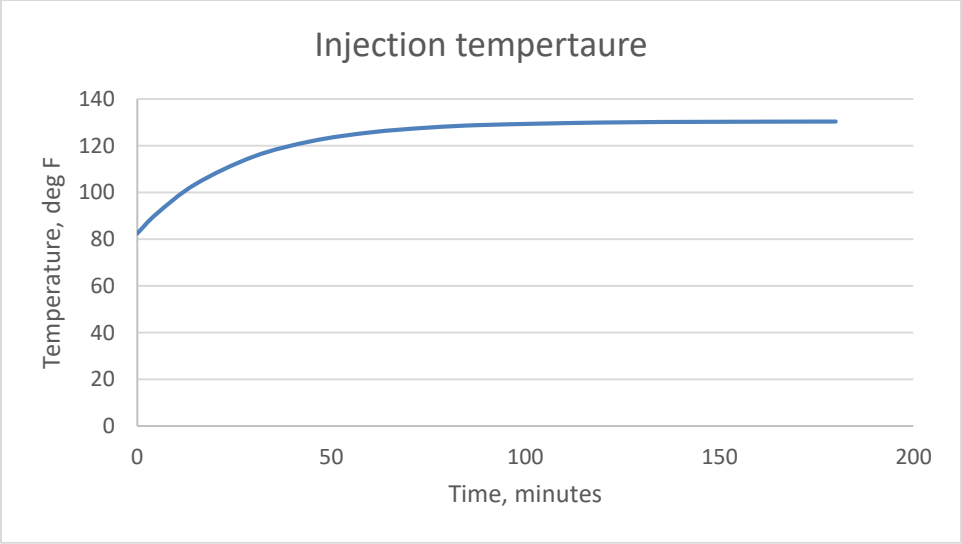


**Figure 7.5 Injection temperature history redrawn from Arihara** [179] **data**

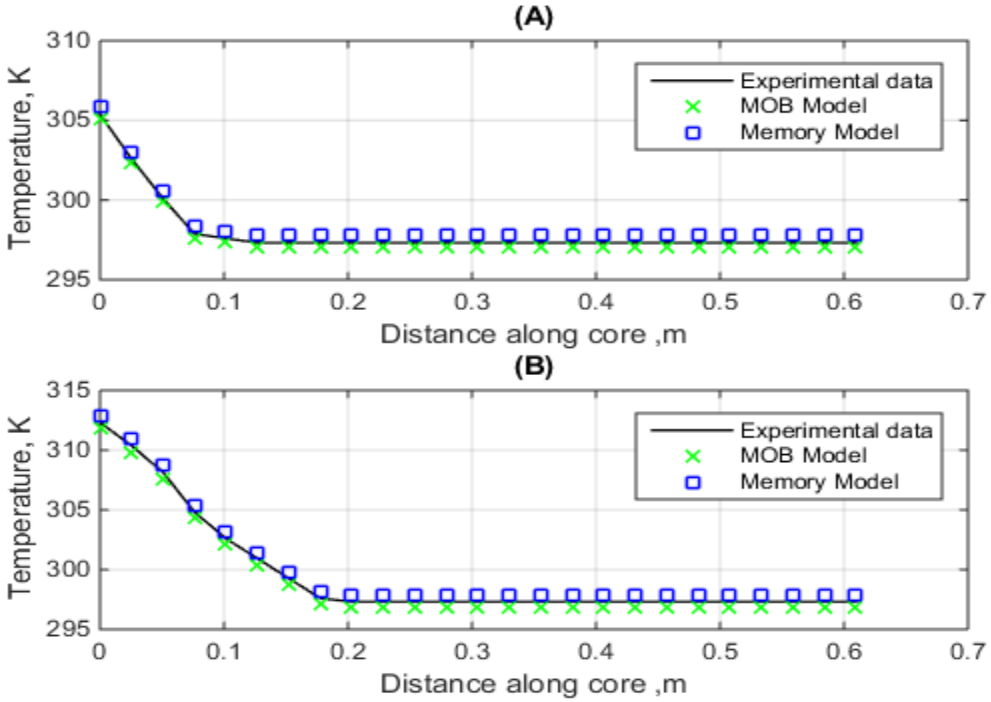


**Figure 7.6 Simulator verification with experimental data by Arihara** [179] **for Berea sandstone 1 (HW-B1) after (A) 5 minutes, and (B) 30 minutes**

As can be seen from Figure 7.6, the developed simulators match the recorded experimental estimates presented in the literature. It is worth noting that the memory exponent was considered to be equal to one for the memory- based model for the verification. The two verification studies establish the accuracy and reliability of the developed numerical simulators and give us confidence to carry out further investigations.

### 7.1.3 Hot fluid Injection: Model comparison

In this section, the results from both non-isothermal simulators are compared in detail using the input data presented in Table 7.3. Also, the region of validity of the proposed modified Boussinesq-based model/simulator is established by comparing the predicted temperature profiles from both numerical simulators at various temperature differential i.e. $\Delta T$. Furthermore, we present some comparison between the results obtained from the classic Boussinesq approach and the newly formulated modified –Boussinesq approach. The differences between the predicted fluid and rock properties in both models are also presented.

**Table 7.3 Rock and fluid properties values for numerical computation**

| Fluid and rock properties | Fluid and rock properties |
|---|---|
| $K = 4.93 \times 10^{-13}$ m$^2$ | $\phi_o = 0.36$ |
| $T_o = 288.15\ °K$ | $\rho_s = 2650\ kg/m\text{^}3$ |
| $T_{ini} = 298.15\ °K$ | $\alpha_L = 5\text{e-}3$ m |
| $T_{inlet} = 353.15\ °K$ | $T_t = 2400\ \ sec$ |
| $P_o = 101.325\ kPa$ | $\rho_f = 1000\ \ kg/m\text{^}3$ |
| $m = 1.5$ | $\mu_o = 0.001\ kg/ms$ |
| $\beta_f = 1.5e-4\ K^{-1}$ | $c_f = 4.4e-10\ Pa^{-1}$ |
| $L = 0.3048\ m$ | $c_\mu = -2.7e-10\ Pa^{-1}$ |
| $Nx = 100$ | $c_T = 2.7e-2\ °K^{-1}$ |
| $d = 0.0508\ m$ | $C_{pw} = 4186$ J/kg/K |
| $k_f = 0.6$ W/(m-K) | $C_{ps} = 820$ J/Kg/K |
| $k_s = 2.5$ W/(m-K) | $a = 2.55 \times 10^{-3}$ |
| $\Delta t = 10\ sec$ | $P_{out} = 276\ kPa$ |
| $c = 0$ | $b = -1 \times 10^{-5}$ |

Figures 7.7 to 7.12 show the temperature profiles predicted by the memory-based numerical simulator (with memory exponent equal 1) and the modified Boussinesq-based numerical simulator for a temperature difference of 25 K, 50 K, and 75 K. i.e. injection temperature of 323.15 K, 348.15 K, and 373.15 K respectively after 0.1 and 0.5 PV injection.

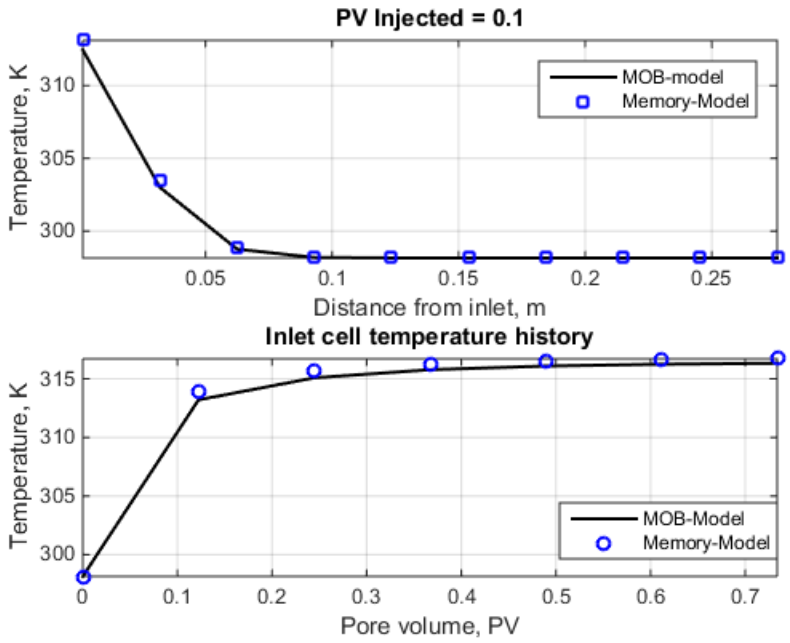


Figure 7.7 Model comparison for temperature difference of 25 K after 0.1 PV injected

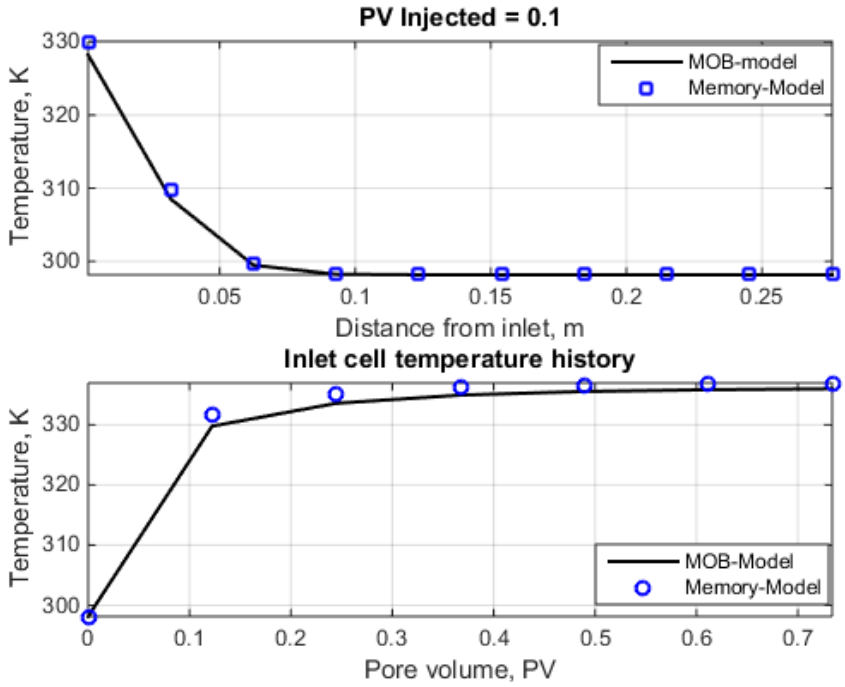


Figure 7.8 Model comparison for temperature difference of 50 K after 0.1 PV injected

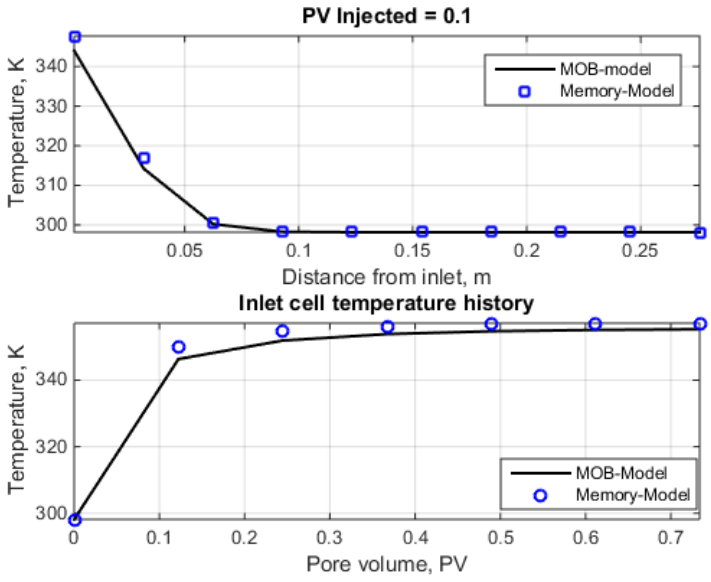


**Figure 7.9 Model comparison for temperature difference of 75 K after 0.1 PV injected**

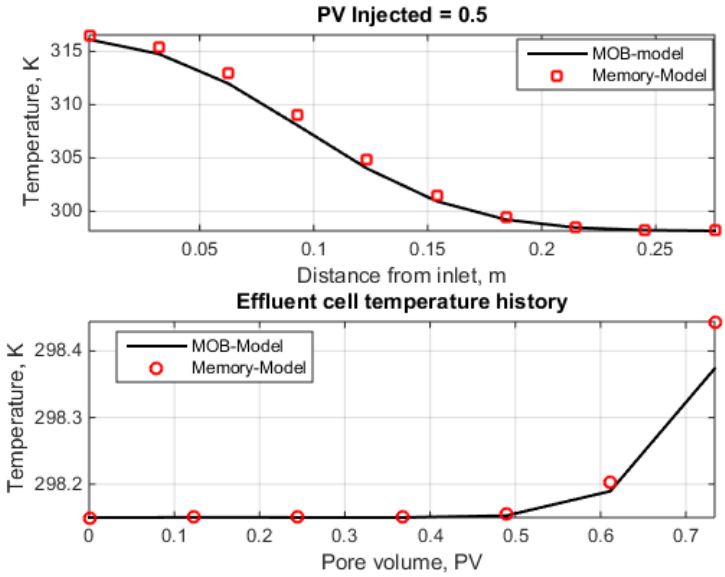


**Figure 7.10 Model comparison for temperature difference of 25 K after 0.5 PV injected**

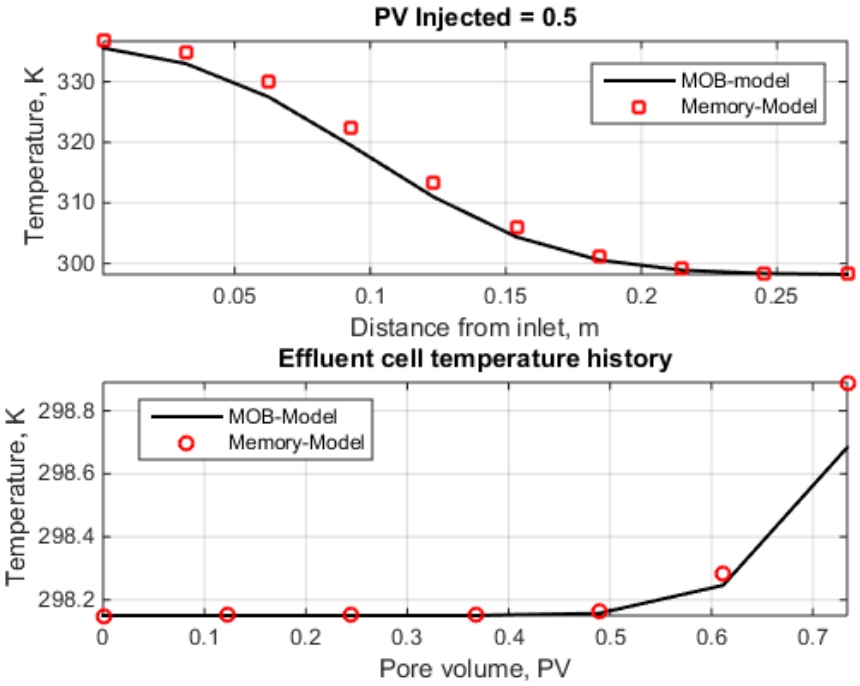


**Figure 7.11 Model comparison for temperature difference of 50 K after 0.5 PV injected**

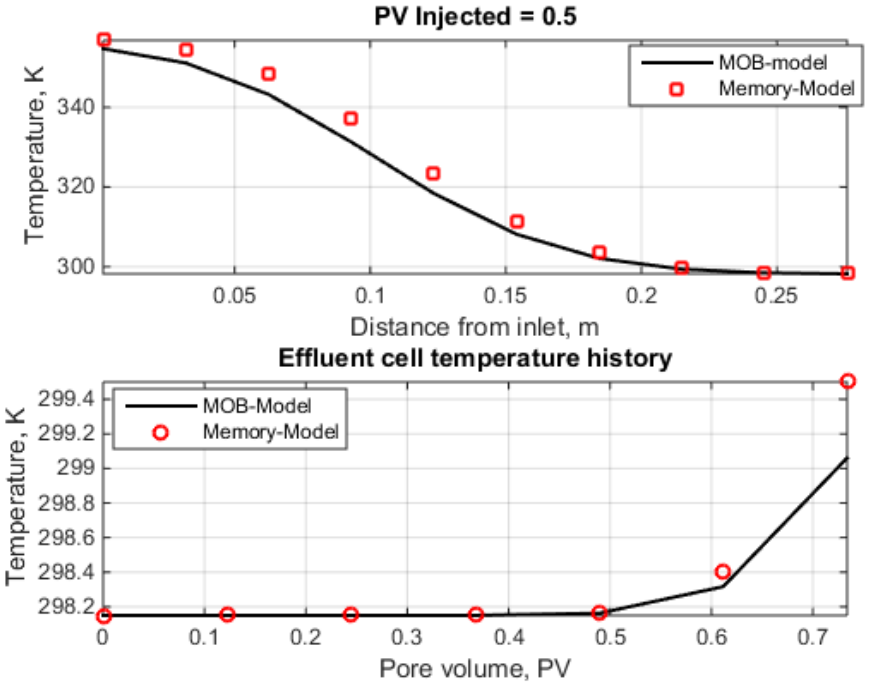


**Figure 7.12 Model comparison for temperature difference of 75 K after 0.5 PV injected**

As can be seen from the above Figures, the simplified modified Boussinesq- based simulator follows the same trend as the more rigorous memory-based simulator but with lower magnitude of temperature predicted along the length of the core. The Figures suggests that the region of validity of the simplified modified Boussinesq-based numerical simulator is around 75 K. Thus, after this temperature limit, the rigorous formulation (memory-based approach) is recommended for accurate temperature prediction.

Furthermore, in order to understand the key parameter contributing to the observed difference, the viscosity, and the permeability profiles predicted from both simulators after 0.5 PV of hot water had been injected is presented below.

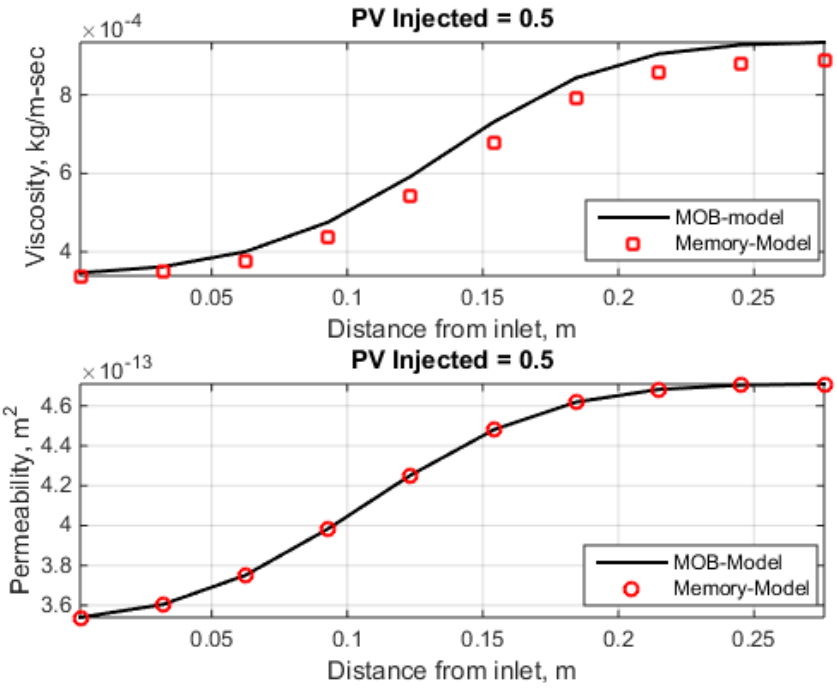


**Figure 7.13 Model comparison after 0.5 PV injection for (a) viscosity profile, and (b) permeability profile**

Clearly, Figures 7.12 to 7.13 reveals that the validity of the modified-Boussinesq simulator is affected by the difference between density, viscosity and velocity calculated from the different models.

### 7.1.4 Effect of Fractional order derivative

Using the same data presented in Table 7.3, the effect of the order of fractional derivative (memory exponent) on the temperature profile was investigated for $\gamma = 0.6$, 0.8, and 1 see Figures 7.14 to 7.18.

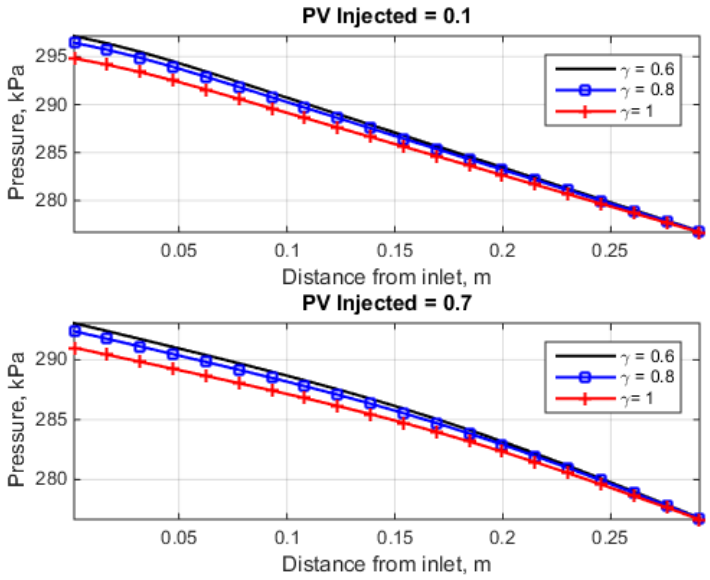


**Figure 7.14 Effect of memory exponent on pressure profile after (a) 0.1, and (b) 0.7 PV injection**

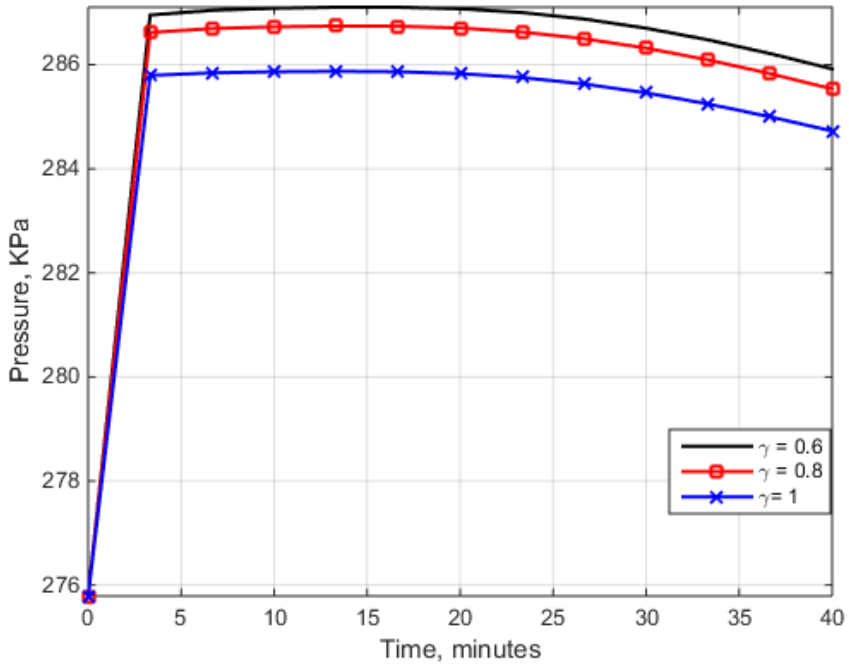


**Figure 7.15 Pressure history at cell 50 for different values of memory exponent**

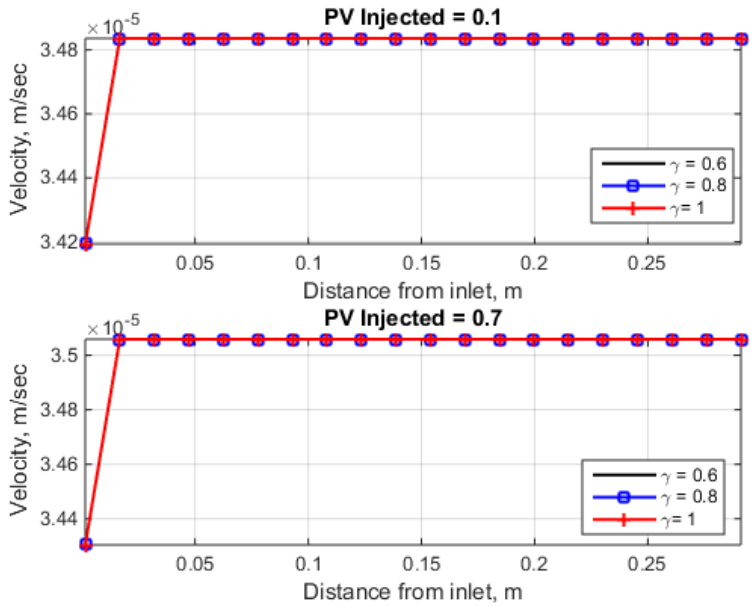


**Figure 7.16 Effect of memory exponent on velocity profile after (a) 0.1, and (b) 0.7 PV injection**

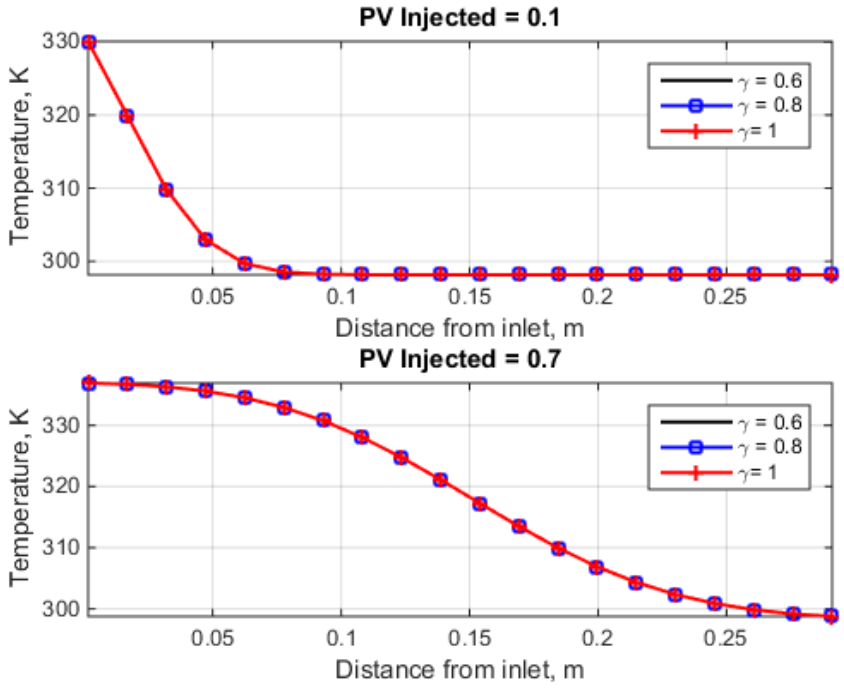


**Figure 7.17 Effect of memory exponent on temperature profile after (a) 0.1, and (b) 0.7 PV injection**

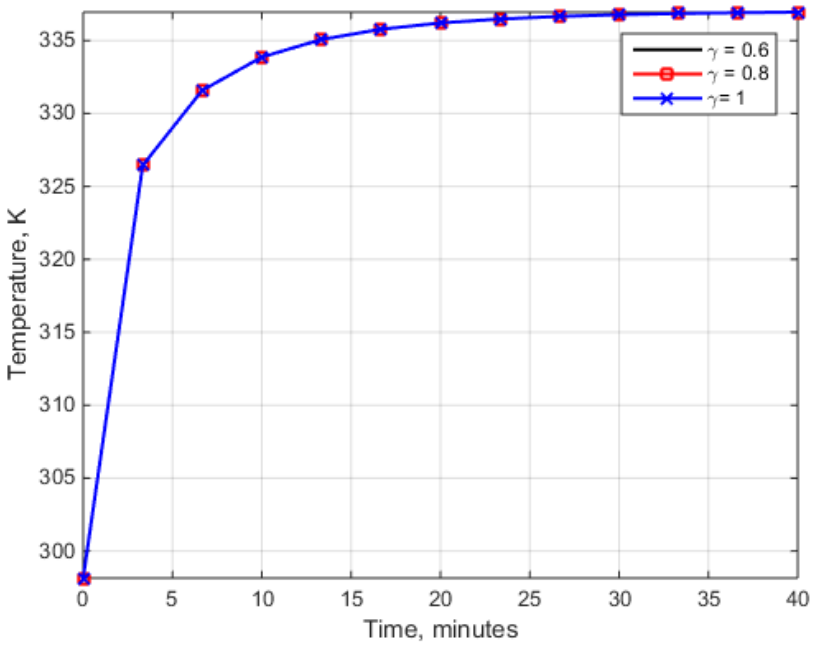


**Figure 7.18 Temperature history at inlet cell for different value of memory exponent**

Figures 7.14 and 7.15 suggest that memory exponent influences the predicted pressure along the core length. However, the effect of the memory exponent is within 3kPa which is very negligible. Figure 7.16 reveals the memory exponent has no influence on the velocity profile due to the minimal effect of the memory exponent on pressure. The effect of the fractional order derivative term on the energy balance although indirect, is as a result of the contribution of the convective energy flux only. This way, the effect of the memory exponent on temperature profile can be attributed to the convective flux alone.

Again, Figures 7.17 and 7.18 reveals that the memory exponent does not impact the temperature profile along the core length. This can be understood through the convective flux term which is strongly dependent on the fluid velocity. Moreover, since Figure 7.16 revealed a negligible effect of the memory exponent on velocity thus the memory exponent would have negligible effect on the temperature profile.

### 7.1.5 Effect of injection rate

The effect of injection rate (mass flow rate) on the fluid velocity in the porous medium investigated using three different injection rate as presented in Figure 7.19.

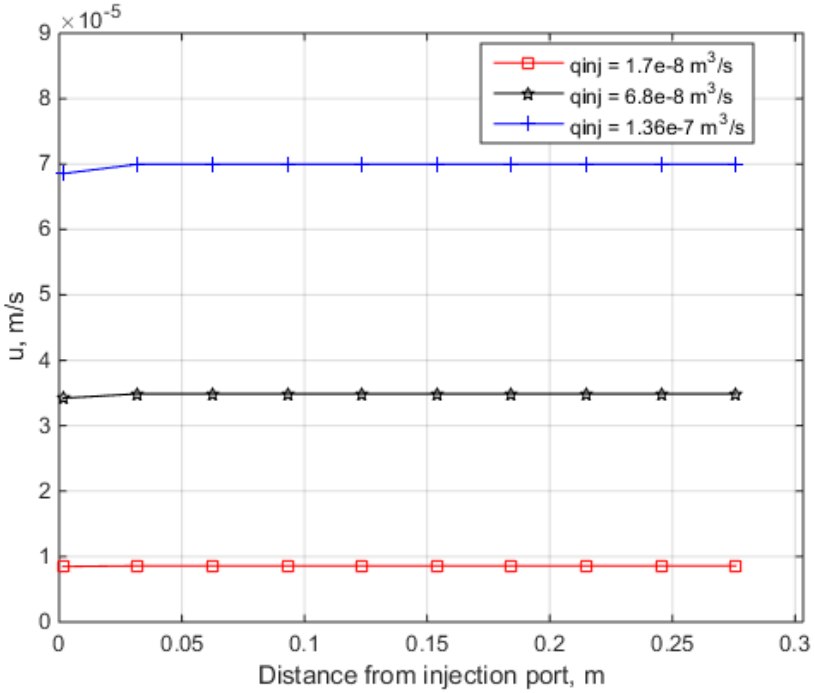


**Figure 7.19 Velocity profile for different injection rates after 10 minutes of HFI**

As can be inferred from Figure 7.19, the velocity magnitude in the porous media is dependent on the hot water injection rate. Results reveal that as the velocity is directly proportional to the injection rate i.e. the higher the injection rate. Therefore, we expect the temperature profile to also be dependent on the hot water injection rate due to the contribution of the convective flux. Figure 7.20 to 7.22 depict the effect of injection rate on temperature, porosity and permeability at different time intervals.

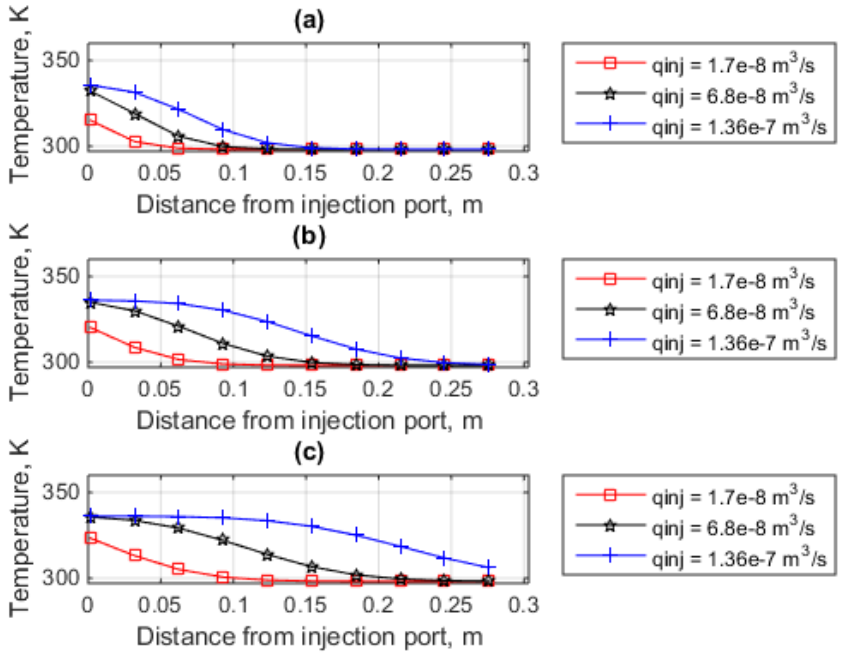


**Figure 7.20 Temperature profile for different injection rates after (a) 10 minutes, (b) 20 minutes, and (c) 30 minutes of HFI**

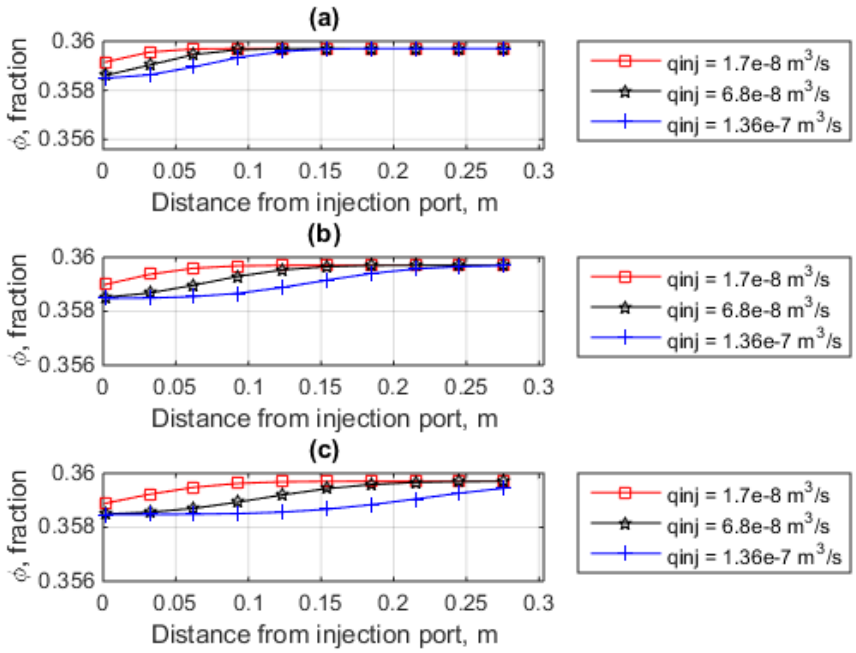


**Figure 7.21 Porosity profile for different injection rates after (a) 10 minutes, (b) 20 minutes, and (c) 30 minutes of HFI**

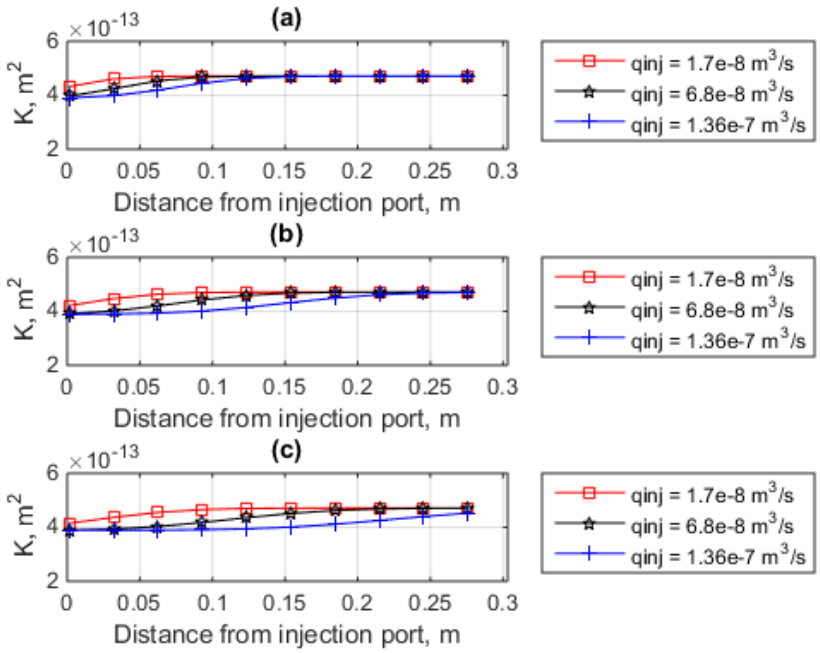


**Figure 7.22 Permeability profile for different injection rates after (a) 10 minutes, (b) 20 minutes, and (c) 30 minutes of HFI**

The numerical results presented in Figure 7.20 establishes the earlier notion that as the rate of hot water injection increases the rate of propagation of the thermal front increases thus leading to higher temperature values being predicted along the length of the core. On the other hand the rate of porosity and permeability impairment increases as the injection rate increases. Thus is a tradeoff between faster heating of the porous media or maintaining the integrity of the rock porosity and permeability. Figures 7.21 and 7.22 suggest that as the injection rate increases the porosity impairment especially close to the injection port reduces severely. A 0.56% reduction in porosity was observed after 30 minutes of injection for an injection rate 1.36e-7 $m^3/s$. Similarly, a 20% permeability reduction was observed close to the injection port after 30 minutes of hot water injection at the same rate of 1.36e-7 $m^3/s$.

## 7.2 Non-isothermal Numerical Simulators- NOLTE

In this section, we present results of the computations carried out for the case of NOLTE. The input data for the simulator are same as those presented in Table 7.3.

The following boundary conditions are considered:

$$T_w = T_s = T_{init}, 0 \leq x \leq L, \ t = 0 \tag{7.4}$$

$$\left(\rho u C_{pw} T\right)_{inlet} = \rho u C_{pw} T - \phi k_w \frac{\partial T_w}{\partial x}, x = 0, \ t > 0 \tag{7.5}$$

$$\frac{\partial T_s}{\partial x} = 0, \ x = 0, t > 0 \tag{7.6}$$

$$\frac{\partial T_s}{\partial x} = \frac{\partial T_w}{\partial x} = 0, \ x = L, t > 0 \tag{7.7}$$

$$P = P_{init}, 0 \leq x \leq L, \ t = 0 \tag{7.8}$$

$$q = q_{inj}, x = 0, \ t > 0 \tag{7.9}$$

$$P = P_{out}, \ x = L, t > 0 \tag{7.10}$$

**Case 1: Effect of Injection rate on fluid velocity and temperature**

Figure 7.23 shows the effect of different hot water injection rates $q_{inj} = 8.33\times 10^{-9}$ $m^3/s$, $3.3\times 10^{-8}$ $m^3/s$, and $6.68\times 10^{-8}$ $m^3/s$ (0.5 ml/min, 2 ml/min, and 4 ml/min) on the velocity profile at the control volume centered along the length of the core plug. In figure 7.23(a) the velocity profile after 10 minutes is presented, similarly figure 7.23(b) after 20 minutes and finally figure 7.23(c) after 30 minutes. The predicted fluid velocity in the core increases with increasing injection rate as one might expect. Furthermore, it must be noted that the velocity profile was calculated using the memory constitutive equation with $\gamma = 1$. Yutse and Acedo [199] pointed out that under such case ( i.e. $\gamma = 1$), the Grunwald-Letnikov operator reduces to identity operator thus the flux relationship reduces to the classic Darcy equation. The figure shows that the velocity varies linearly with distance with an almost constant magnitude in all cases.

Figures 7.24 to 7.26 depicts the effect of fluid injection rates on the fluid and rock temperature in the porous medium for the injection rates as above.

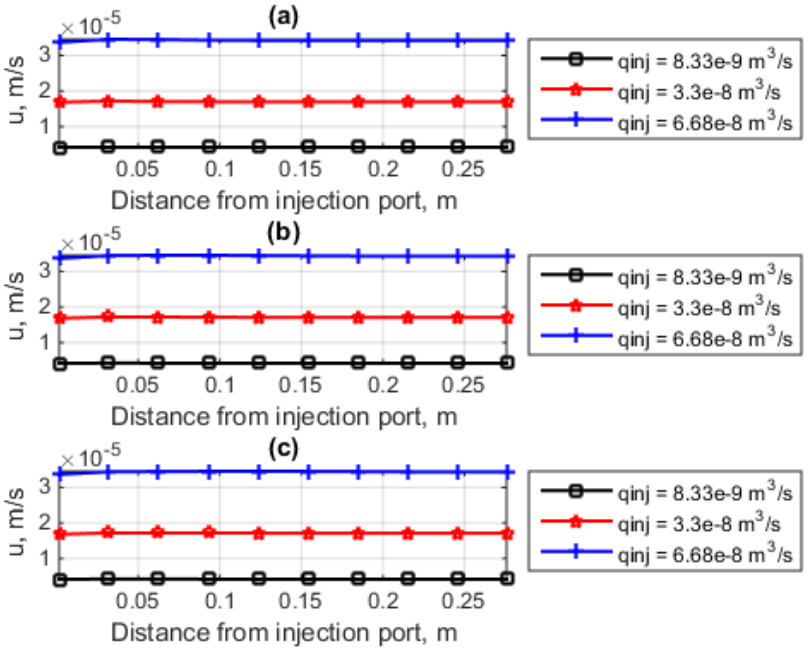


**Figure 7.23 Velocity profile for $\gamma = 1$ after, (a) 10 minutes (b) 20 minutes (c) 30 minutes**

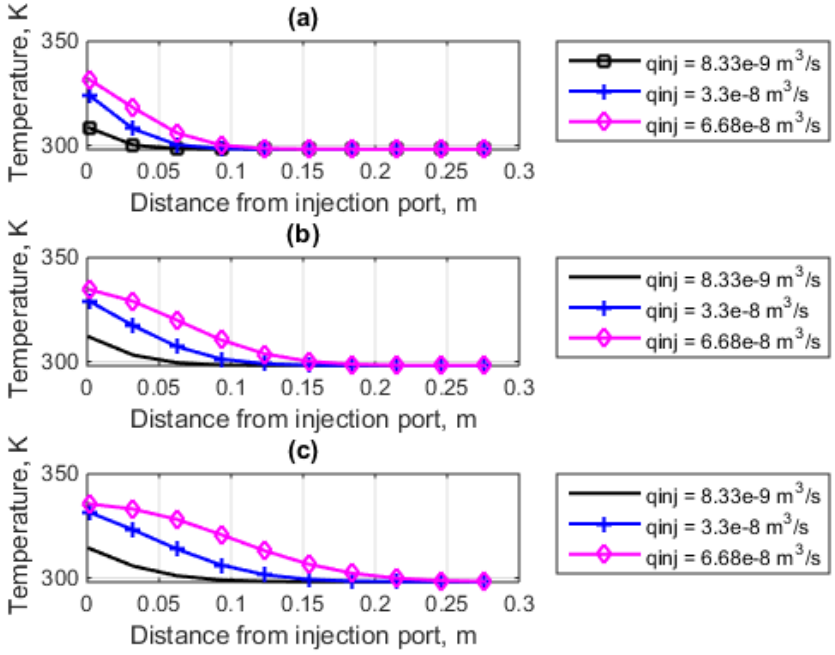


**Figure 7.24 Fluid temperature profile for different injection rates after (a) 10 minutes, (b) 20 minutes, and (c) 30 minutes for $\gamma = 1$**

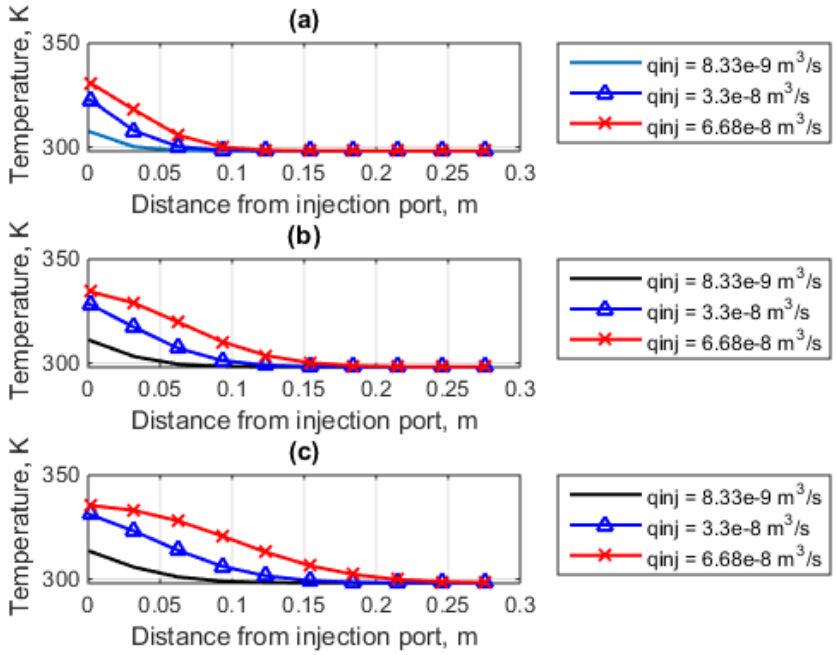


**Figure 7.25 Rock temperature profile for different injection rates after (a) 10 minutes, (b) 20 minutes, and (c) 30 minutes for $\gamma = 1$**

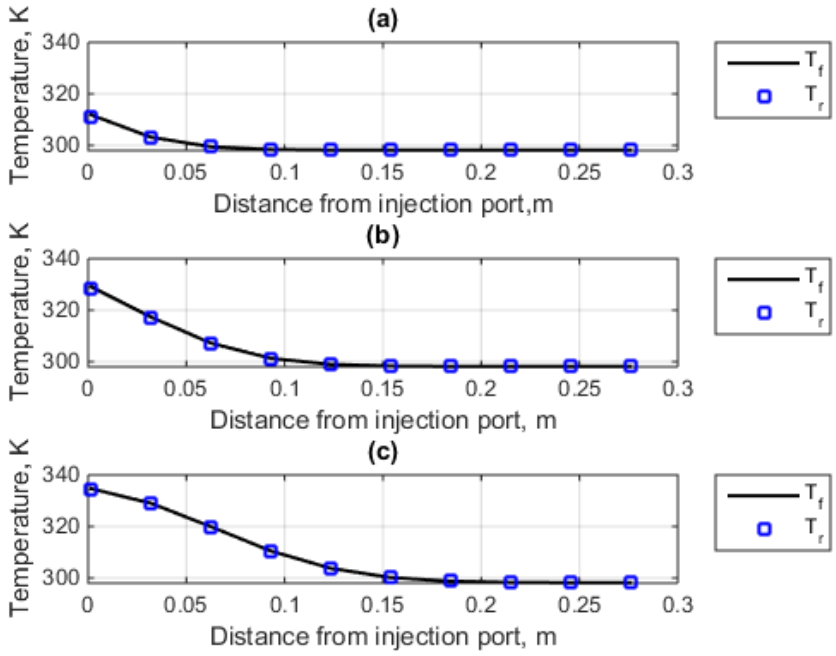


**Figure 7.26 Fluid and rock temperature profile for different injection rates after 20 minutes (a) 0.5 ml/minutes, (b) 2 ml/minutes, and (c) 4 ml/minutes for $\gamma = 1$**

Figure 7.24 shows the fluid temperature is dependent on the injection rate as this affects the propagation of the hot water front in the porous medium. A similar conclusion can be inferred for the rock temperature profile since the temperature in the rock matrix is controlled by the heat transfer coefficient and the temperature gradient between rock matrix and fluid phase. Figure 7.26 compares the rock temperature and fluid phase temperature at different injection rates after 20 minutes of hot water injection. The difference in magnitude between both temperatures is negligible and thus the assumption of equal temperature i.e. LTE is sufficient.

In summary, Figures 7.24 to 7.26 predict similar temperature propagation trend with higher values close to the injection port. Further investigation reveals that the temperature profile predicted at any location within the porous medium increases as the injection rate increases. Clearly, this is attributed to the increasing effect of the magnitude of the convective flux in the energy equation. Thus, the increasing in fluid injection rate corresponds to an increase in fluid velocity in the porous medium and in turn an increase in heat transport due to convection. Moreover, the effect of heat conduction remains constant because the thermal conductivity of the fluid and rock is not allowed to vary with temperature in this study. Although the effective thermal conductivity varies as a result of porosity alteration in the porous medium. Hence we can suggest that he difference in temperature, may be partially attributed to fluid injection rate only.

The temperature history at the injection block and effluent block is presented in Figures 7.27 and 7.28. This is important for design analysis in that it gives us a feel of when the hot water front reaches the producer well (i.e. breakthrough time).

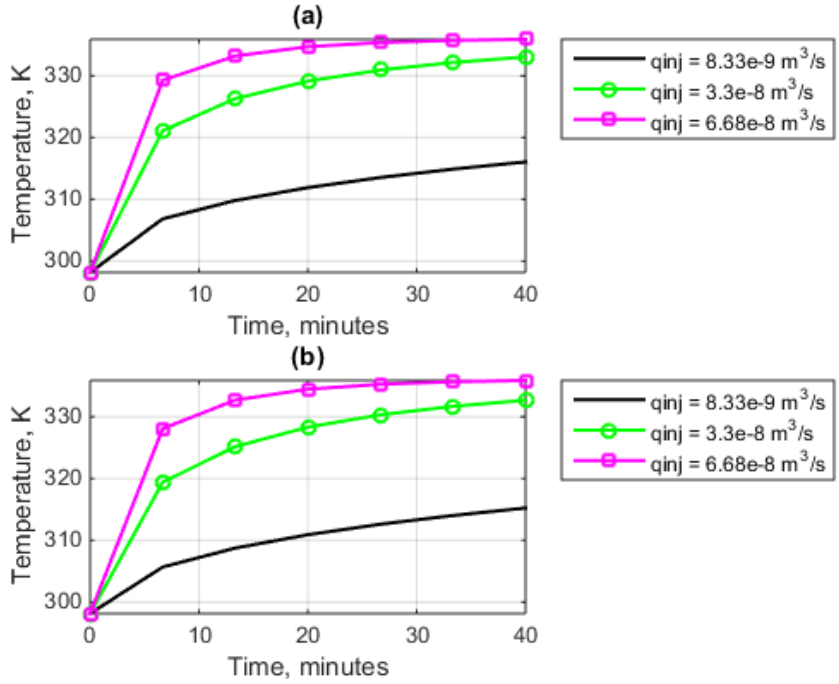


**Figure 7.27 Injection block temperature history for different injection rate when $\gamma = 1$ for (a) fluid (b) rock**

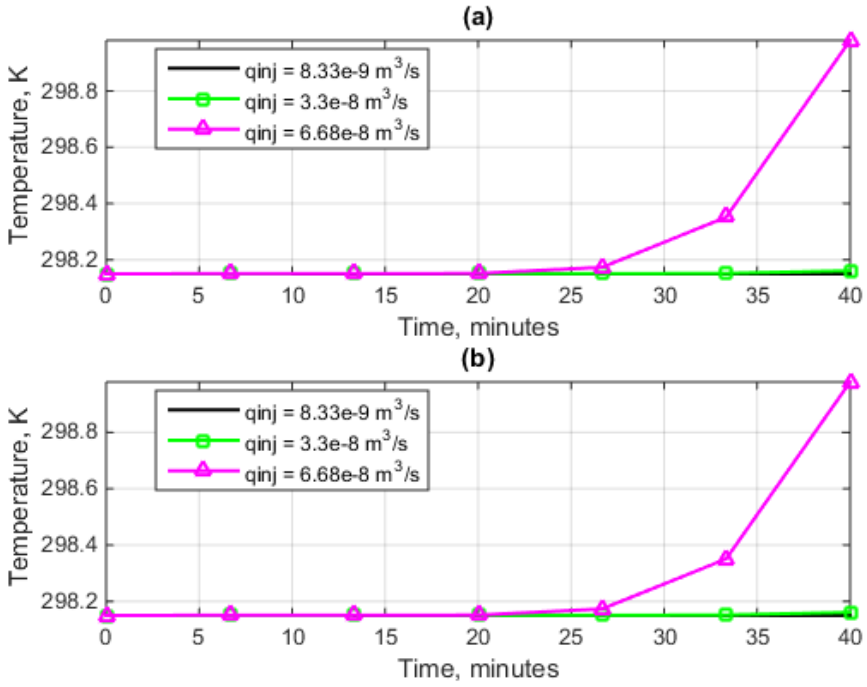


**Figure 7.28 Effluent block temperature history for different injection rate when $\gamma = 1$ for (a) fluid (b) rock**

It is obvious from Figure 7.27 that at higher injection rates usually translate to higher contribution of convective flux to the total energy flux in the porous medium. Figure 7.28 reveals that breakthrough time for the hot water front is dependent on the rate of hot water injection. The temperature history at the effluent block reveals that the larger the injection rate the faster the effect of temperature is felt at the effluent location. A closer look at Figure 7.28 suggest that the hot water front effect is noticed after 26 minutes after the start of injection for the injection rate of 4 ml/min. The effect of the hot water in not observed at the effluent block for the other injection rates throughout the simulation run.

**Case 2: Effect of fractional order derivative $\gamma$ on velocity and temperature**

The variation of fluid velocity at the control volume center, along the length of the core is presented in Figure 7.29 at different time intervals for $\gamma = 0.6$, 0.8 and 1 respectively. For all cases, the same velocity trend is predicted. Most importantly, the result shows that the fractional order has no effect on the velocity profile.

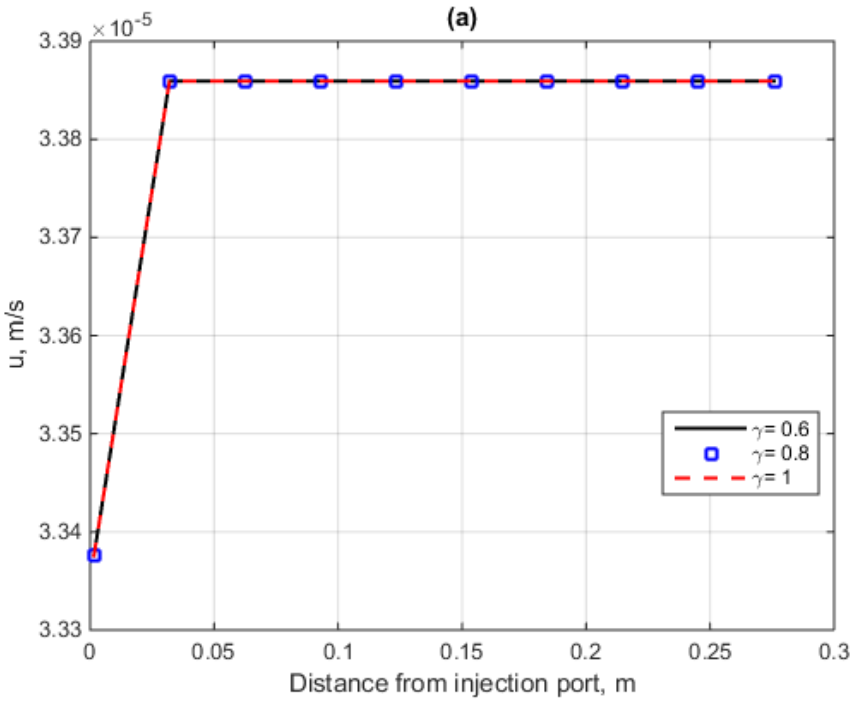


Figure 7.29: Velocity profile for different values of $\gamma$ after 15 minutes

The result from the velocity profile presented in Figure 7.29 suggest that the memory exponent would have little or no effect on the fluid and rock temperature profiles. However, to ascertain this hypothesis, Figure 7.30 and 7.31 presents the fluid and rock temperature profiles for different values of $\gamma$.

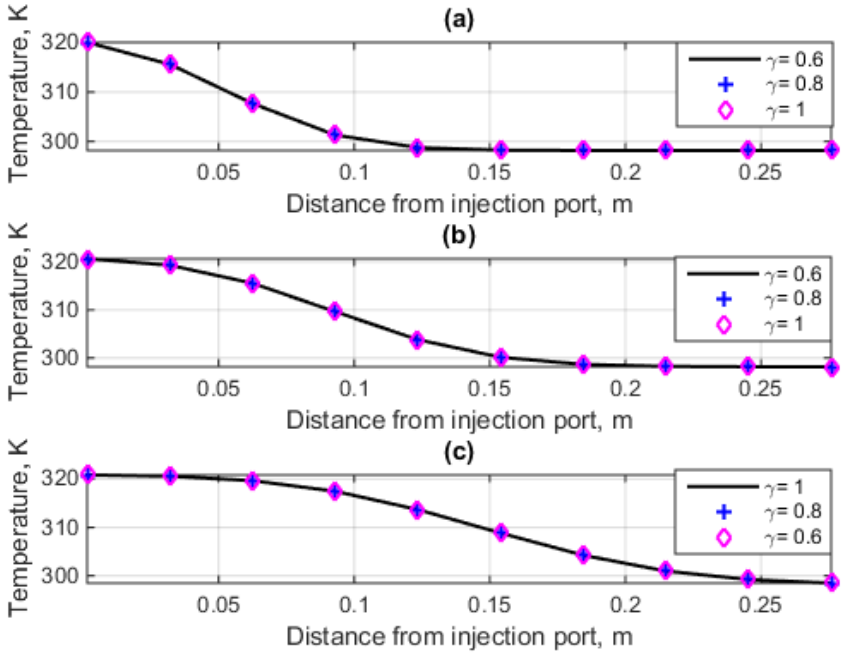


**Figure 7.30: Fluid temperature profile for different for different values of $\gamma$**

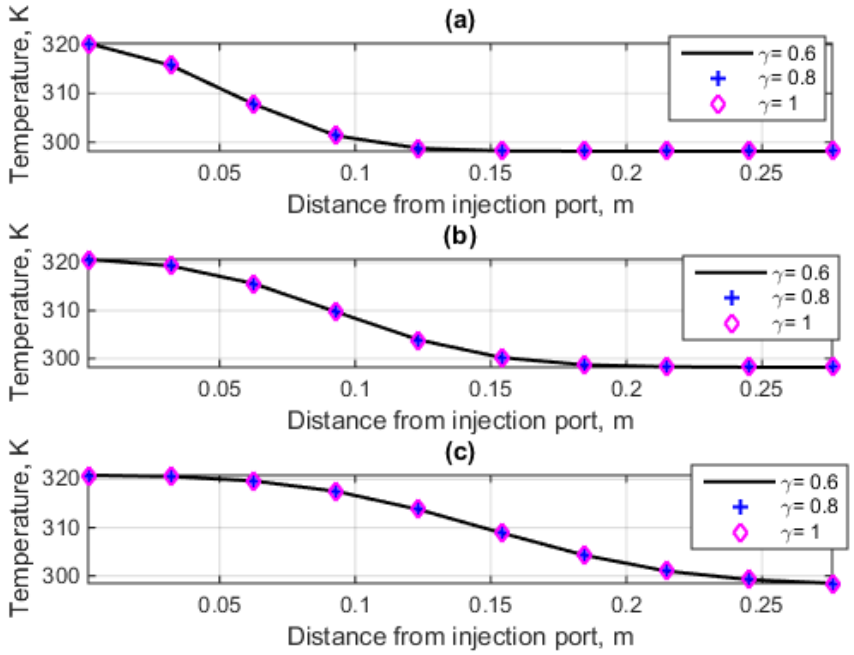


**Figure 7.31: Rock matrix temperature profile for different values of $\gamma$**

As can be seen clearly in the above figures, the memory exponent does not control the temperature propagation in the porous medium. In summary, the LTE assumption is adequate for predicting the temperature profile in a reservoir.

# CHAPTER 8

# CONCLUSIONS AND RECCOMENDATION

Conclusions from this research would be summarized next.

1. Increasing the injection rate of hot water leads to a larger alteration of rock porosity and permeability especially in the region close to the injection port.

2. Numerical runs reveal that the effect of memory on pressure to be very minimal, and also negligible in the temperature profile for the simulation run time considered.

3. A modified Boussinesq-based mathematical model describing hot fluid injection process in a porous medium was successfully developed considering alteration of rock physical properties due to temperature.

4. A more rigorous memory-based approach to modelling or describing fluid flow in a porous media was successfully developed where the time fractional derivative is interpreted with the Grunwald-Letnikov definition.

5. Two numerical simulators were successfully developed for predicting the temperature profile in a porous medium due to injection of a hot fluid considering cases of local thermal equilibrium and no-local thermal equilibrium. The numerical simulator was developed using a fully implicit discretization approach.

6. The corresponding non-algebraic obtained from discretization of the governing differential equations were handled using an iterative scheme (successive over relaxation method) with a specified tolerance of 1e-6.

7. The stability of the developed numerical scheme for solving the memory-based flow equation was derived in detail. Moreover, the numerical experiments established the stability of the numerical scheme.

8. The predictions from the non-isothermal simulators were verified with analytical solution of the simplified problem and experimental data published in literature. Subsequently, various case studies were presented to establish the accuracy and robustness of the simulators.

9. A new wellbore flow model was derived based on the memory based generalized constitutive equation to predict the wellbore flowing pressure.

10. The proposed modified-Boussinesq model was established to be valid within a temperature interval of 75 to 80 K. The region could be extended by disregarding the simplified linear equation of state relating the fluid density to temperature and pressure.

11. The memory formalism presented by Hossain et *al.*[127] implies that only the memory exponent term would be employed for minimizing the error between the experimental data and the results from the mathematical model.

Recommendations for future work are summarized below:

1. A critical analysis of the physical implication of the volumetric flux proposed by Hossain et *al.*[127].

2. Development of more efficient methods to address the high memory and computational requirements of the memory formulation.

# APPENDIX A: DISCRETIZATION OF PRESSURE EQUATION FOR MB MODEL UNDER LTE AND NOLTE (IMPIT SCHEME)

In order to solve Eq. (4.5) numerically, the fractional Equation need to be discretized and converted to an algebraic form. In order to approximate $D_t^{1-\gamma}$operator, Eq. (4.3) was employed. Starting with Eq. (4.5) and defining a function $\psi(\gamma,k)$,

$$\psi(\gamma,k)=\frac{(-1)^k\,\Gamma(2-\gamma)}{k!\,\Gamma(2-\gamma-k)} \tag{A1}$$

Noting that $\psi(\gamma,0)=1$, and $\psi(\gamma,k)=-\psi(\gamma,k-1)\frac{(2-\gamma-k)}{k}$, for $k>0$. (A2)

Due to the introduction of $\psi(\gamma,k)$, the final form of the memory flow model becomes

$$\lim_{\tau\to 0}\tau^{\gamma-1}\sum_{k=0}^{t/\tau}\psi(\gamma,k)\,\frac{\partial}{\partial x}\left(\frac{\eta}{B_w}\frac{\partial p(t-n\tau)}{\partial x}\right)=\frac{1}{\alpha_c\Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1}-\left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{A3}$$

Multiplying Eq. (A3) by $A_x\Delta x$ results in

$$\lim_{\tau\to 0}\tau^{\gamma-1}\sum_{k=0}^{t/\tau}\psi(\gamma,k)\,\frac{\partial}{\partial x}\left(\frac{\eta A_x}{B_w}\frac{\partial p(t-n\tau)}{\partial x}\right)\Delta x=\frac{V_b}{\alpha_c\Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1}-\left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{A4}$$

IMPIT Scheme

The next step is to discretize the spatial second derivative on the right hand side of the Eq. and approximating the coefficients at the Left Hand Side (LHS) with the values at time step *n*.

Taking the first term in bracket from LHS of Eq. (A4).

$$\lim_{\tau\to 0}(\Delta t)^{\gamma-1}\sum_{k=0}^{t/\tau}\psi(\gamma,k)\frac{\partial}{\partial x}\left[\frac{\eta A_x}{B_w}\frac{\partial p(t-n\tau)}{\partial x}\right]\Delta x=$$

$$\frac{1}{(\Delta t)^{1-\gamma}}\sum_{k=0}^{t/\Delta t}\psi(\gamma,k)\left[\left(\frac{\eta A_x}{B_w\Delta x}\right)_{i+\frac{1}{2}}^{n+1}\left(p_{i+1}^{n+1-k}-p_i^{n+1-k}\right)+\left(\frac{\eta A_x}{B_w\Delta x}\right)_{i-\frac{1}{2}}^{n+1}\left(p_{i-1}^{n+1-k}-p_i^{n+1-k}\right)\right] \tag{A5}$$

Introducing the fluid pseudo transmissibility as

$$T_p = \frac{\eta A_x}{B_w \Delta x}, \tag{A6}$$

Combining both sides of Eq. (A4), and approximating all coefficients at the next time step. We have as follows:

$$\frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] =$$

$$D\left[\psi(\gamma,0)\left\{T_{p,i-\frac{1}{2}}^{n+1} p_{i-1}^{n+1} - \left(T_{p,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1}\right) p_i^{n+1} + T_{p,i+\frac{1}{2}}^{n+1} p_{i+1}^{n+1}\right\}\right] +$$

$$D\sum_{k=1}^{t/\Delta t} \psi(\gamma,k)\left[T_{p,i-\frac{1}{2}}^{n+1} p_{i-1}^{n+1-k} - \left(T_{p,i-\frac{1}{2}}^{n+1} + T_{p,i+\frac{1}{2}}^{n+1}\right) p_i^{n+1-k} + T_{p,i+\frac{1}{2}}^{n+1} p_{i+1}^{n+1-k}\right] \tag{A7}$$

Where, $D = \frac{1}{(\Delta t)^{1-\gamma}}$, and the fluid transmissibility term is defined as;

$$T_x = \frac{\eta A_x D}{B_w \Delta x} = D T_p \tag{A8}$$

The transmissibility at the block interface $i \mp \frac{1}{2}$, is approximated using upstream weighting as explained in chapter 5.

Thus Eq. (A7) simplifies to

$$\frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] =$$

$$\left[\psi(\gamma,0)\left\{T_{x,i-\frac{1}{2}}^{n+1} p_{i-1}^{n+1} - \left(T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1}\right) p_i^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} p_{i+1}^{n+1}\right\}\right] +$$

$$\sum_{k=1}^{t/\Delta t} \psi(\gamma,k)\left[T_{p,i-\frac{1}{2}}^{n+1-k} p_{i-1}^{n+1-k} - \left(T_{p,i-\frac{1}{2}}^{n+1-k} + T_{p,i+\frac{1}{2}}^{n-k+1}\right) p_i^{n+1-k} + T_{p,i+\frac{1}{2}}^{n+1-k} p_{i+1}^{n+1-k}\right] \tag{A9}$$

Further simplification leads to;

$$T_{x,i-\frac{1}{2}}^{n+1} p_{i-1}^{n+1} - \left(T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1}\right) p_i^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} p_{i+1}^{n+1} =$$

$$\frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] -$$

$$\sum_{k=1}^{t/\Delta t} \psi(\gamma,k)\left[\left(T_{x,i-\frac{1}{2}}^{n+1-k} p_{i-1}^{n+1-k} - \left(T_{x,i-\frac{1}{2}}^{n+1-k} + T_{x,i+\frac{1}{2}}^{n+1-k}\right) p_i^{n+1-k} + T_{x,i+\frac{1}{2}}^{n+1-k} p_{i+1}^{n+1-k}\right)\right]$$

(A10)

Noting that $\psi(\gamma, 0) = 1$ was substituted in Eq. (A10).

The final form of Eq. (A10) is presented below:

$$\left(T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1}\right) p_i^{n+1} = T_{x,i-\frac{1}{2}}^{n+1} p_{i-1}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} p_{i+1}^{n+1} + \sum_{k=1}^{t/\Delta t} \psi(\gamma, k) \delta_i^{n+1-k} - \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right]$$

(A11)

Eq. (A11) is only applicable to the interior grid blocks.

Where $\delta_i^{n+1-k}$,the finite is difference kernel, and expressed as

$$\delta_i^{n+1-k} = \left(T_{x,i-\frac{1}{2}}^{n+1-k} p_{i-1}^{n+1-k} - \left(T_{x,i-\frac{1}{2}}^{n+1-k} + T_{x,i+\frac{1}{2}}^{n+1-k}\right) p_i^{n+1-k} + T_{x,i+\frac{1}{2}}^{n+1-k} p_{i+1}^{n+1-k}\right) \quad \text{(A11a)}$$

Therefore, Eq. (A11) is in the form

$$a_p \mathrm{p}_p = \sum_{nb} a_{nb} p_{nb} + b + a_p^0 \mathrm{p}_p^n \quad \text{(A12)}$$

Where,

$$a_E = T_{x,i+\frac{1}{2}}^{n+1} \quad \text{(A12a)}$$

$$a_w = T_{x,i-\frac{1}{2}}^{n+1} \quad \text{(A12b)}$$

$$a_p^0 = 0 \quad \text{(A12c)}$$

$$b = \sum_{k=1}^{t/\Delta t} \psi(\gamma, k) \delta_i^{n+1-k} - \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] \quad \text{(A12d)}$$

$$a_p = \left(T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1}\right) \quad \text{(A12e)}$$

In order to incorporate the boundary conditions, two imaginary grid blocks are introduced, a grid block ‘0’ which is assumed to be adjacent to the first grid block. Here the injection rate/constant pressure boundary condition is modeled. And the other imaginary grid block

is grid block '*nx+1*' which is adjacent to the last grid block. Here the constant pressure at the outlet is employed.

Constant Injection rate

According to Darcy's law at each time step at the inlet:

$$\frac{q_{HW}}{B_{w,i}^{n+1}} = -\frac{\eta A_x}{B_{w,i}^{n+1}} D_t^{1-\gamma} \left(\frac{\partial p}{\partial x}\right) \tag{A12}$$

From Eq. (A7) we have as follows

$$\frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] =$$

$$\frac{1}{(\Delta t)^{1-\gamma}} \sum_{k=0}^{t/\Delta t} \psi(\gamma, k)\left[T_{p,i-\frac{1}{2}}^{n+1-k}\left(p_{i-1}^{n+1-k} - p_i^{n+1-k}\right) + T_{p,i+\frac{1}{2}}^{n+1-k}\left(p_{i+1}^{n+1-k} - p_i^{n+1-k}\right)\right] \tag{A13}$$

Thus introducing the boundary condition into Eq. (A13) results in

$$\frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] = \frac{q_{HW}}{B_{w,i}^{n+1}} + T_{x,i+\frac{1}{2}}^{n+1}(p_{i+1}^{n+1} - p_i^{n+1}) +$$
$$\sum_{k=1}^{t/\Delta t} \psi(\gamma, k)\left[T_{x,i+\frac{1}{2}}^{n+1-k}\left(p_{i+1}^{n+1-k} - p_i^{n+1-k}\right)\right]$$

(A14)

Further simplifying Eq. (A14)

$$-\left(T_{x,i+\frac{1}{2}}^{n+1}\right) p_i^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} p_{i+1}^{n+1} = \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] - \frac{q_{HW}}{B_{w,i}^{n+1}} -$$

$$\sum_{k=1}^{t/\Delta t} \psi(\gamma, k)\left[T_{x,i+\frac{1}{2}}^{n+1-k}\left(p_{i+1}^{n+1-k} - p_i^{n+1-k}\right)\right] \tag{A15}$$

The final form of Eq. (A15) is presented below:

$$\left(T_{x,i+\frac{1}{2}}^{n+1}\right) p_i^{n+1} = T_{x,i+\frac{1}{2}}^{n+1} p_{i+1}^{n+1} + \frac{q_{HW}}{B_{w,i}^{n+1}} - \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] + \sum_{k=1}^{t/\Delta t} \psi(\gamma, k)\delta_i^{n+1-k} \tag{A16}$$

Where $\delta_i^{n+1-k}$,the finite is difference kernel, and expressed as

$$\delta_i^{n+1-k} = T_{x,i+\frac{1}{2}}^{n+1-k}\left(p_{i+1}^{n+1-k} - p_i^{n+1-k}\right) \tag{A16a}$$

Therefore, Eq. (A16) is in the form

$$a_p \mathrm{p}_p = \sum_{nb} a_{nb} p_{nb} + b + a_p^0 \mathrm{p}_p^n \tag{A17}$$

Where,

$$a_E = T_{x,i+\frac{1}{2}}^{n+1} \tag{A17a}$$

$$a_p^0 = 0 \tag{A17b}$$

$$b = \frac{q_{HW}}{B_{w,i}^{n+1}} - \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] + \sum_{k=1}^{t/\Delta t} \psi(\gamma, k)\delta_i^{n+1-k} \tag{A17c}$$

$$a_p = T_{x,i+\frac{1}{2}}^{n+1} \tag{A17d}$$

For the outlet boundary condition, similar manipulation is made as follows

From Eq. (A7) we have as follows

$$\frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] = \frac{1}{(\Delta t)^{1-\gamma}}\left[\psi(\gamma, 0)\left\{T_{i-\frac{1}{2}}^{n+1}\left(p_{i-1}^{n+1} - p_i^{n+1}\right) + T_{p,L}^{n+1}\left(p_{out} - p_i^{n+1}\right)\right\}\right] + \frac{1}{(\Delta t)^{1-\gamma}}\sum_{k=1}^{t/\Delta t} \psi(\gamma, k)\left[T_{p,i+\frac{1}{2}}^{n+1-k}\left(p_{i-1}^{n+1} - p_i^{n+1-k}\right) + T_{p,L}^{n+1-k}\left(p_{out} - p_i^{n+1-k}\right)\right] \tag{A18}$$

Simplifying Eq. (A18)

$$T_{x,i-\frac{1}{2}}^{n+1} p_{i-1}^{n+1} - \left(T_{x,i-\frac{1}{2}}^{n+1} + T_{xL}^{n+1}\right) p_i^{n+1} = \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] - T_{xL}^{n+1} p_{out} - \sum_{k=1}^{t/\Delta t} \psi(\gamma, k)\left[T_{x,i+\frac{1}{2}}^{n+1-k}\left(p_{i-1}^{n+1} - p_i^{n+1-k}\right) + T_{xL}^{n+1-k}\left(p_{out} - p_i^{n+1-k}\right)\right] \tag{A19}$$

The final form of Eq. (A19) is presented below:

$$\left(T_{x,i-\frac{1}{2}}^{n+1}+T_{xL}^{n+1}\right)p_i^{n+1}$$
$$= T_{x,i-\frac{1}{2}}^{n+1}\,p_{i-1}^{n+1}+T_{xL}^{n+1}p_{out}-\frac{V_b}{\alpha_c\Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1}-\left(\frac{\phi}{B_w}\right)_i^{n}\right]$$
$$+\sum_{k=1}^{t/\Delta t}\psi(\gamma,k)\delta_i^{n+1-k} \tag{A20}$$

Where $\delta_i^{n+1-k}$,the finite is difference kernel, and expressed as

$$\delta_i^{n+1-k}=T_{xL}^{n+1-k}\left(p_{out}-p_i^{n+1-k}\right)+T_{x,i-\frac{1}{2}}^{n+1-k}\left(p_{i-1}^{n+1-k}-p_i^{n+1-k}\right) \tag{A21}$$

Therefore, Eq. (A21) is in the form

$$a_p\mathrm{p}_p=\textstyle\sum_{nb}a_{nb}p_{nb}+b+a_p^0\mathrm{p}_p^n \tag{A22a}$$

Where,

$$a_w=\ T_{x,i-\frac{1}{2}}^{n+1} \tag{A22b}$$

$$a_p^0=0 \tag{A23c}$$

$$b=T_{xL}^{n+1}p_{out}-\tfrac{V_b}{\alpha_c\Delta t}\left[\left(\tfrac{\phi}{B_w}\right)_i^{n+1}-\left(\tfrac{\phi}{B_w}\right)_i^{n}\right]+\ \textstyle\sum_{k=1}^{t/\Delta t}\psi(\gamma,k)\delta_i^{n+1-k} \tag{A23d}$$

$$a_p=\left(T_{x,i-\frac{1}{2}}^{n+1}+T_{xL}^{n+1}\right) \tag{A23e}$$

# APPENDIX B: DISCRETIZATION OF PRESSURE EQUATION FOR OB MODEL UNDER LTE

In order to solve Eq. (4.13) numerically, the second order pressure derivative is approximated using central order differencing

$$\frac{\partial}{\partial x}\left(\frac{A_x K}{\mu B_w}\frac{\partial p}{\partial x}\right)\Delta x = \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{B1}$$

The next step is to discretize the spatial second derivative on the right hand side of the Equation and approximating the coefficients at the Left Hand Side (LHS) with the values at time step *n*.

$$T_{x,i-\frac{1}{2}}^{n+1}(p_{i-1}^{n+1} - p_i^{n+1}) + T_{x,i+\frac{1}{2}}^{n+1}(p_{i+1}^{n+1} - p_i^{n+1}) = \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{B2}$$

Noting that the fluid transmissibility is defined as;

$$T_x = \frac{KA_x}{\mu B_w \Delta x} \tag{B3}$$

Eq. (B2) can be further simplified as follows:

$$T_{x,i-\frac{1}{2}}^{n+1} p_{i-1}^{n+1} - \left(T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1}\right) p_i^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} p_{i+1}^{n+1} = \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{B4}$$

The final form of Eq. (B5) is presented below:

$$\left(T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1}\right) p_i^{n+1} = T_{x,i-\frac{1}{2}}^{n+1} p_{i-1}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1} p_{i+1}^{n+1} - \frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{B5}$$

Eq. (B5) is only applicable to the interior grid blocks. Therefore, Eq. (B5) is in the form

$$a_p \mathrm{p}_p = \sum_{nb} a_{nb} p_{nb} + b + a_p^0 \mathrm{p}_p^n \tag{B6a}$$

Where,

$$a_E = T_{x,i+\frac{1}{2}}^{n+1} \tag{B6b}$$

$$a_w = T_{x,i-\frac{1}{2}}^{n+1} \tag{B6c}$$

$$a_p^0 = 0 \tag{B6d}$$

$$b = -\frac{V_b}{\alpha_c \Delta t}\left[\left(\frac{\phi}{B_w}\right)_i^{n+1} - \left(\frac{\phi}{B_w}\right)_i^{n}\right] \tag{B6e}$$

$$a_p = \left[T_{x,i-\frac{1}{2}}^{n+1} + T_{x,i+\frac{1}{2}}^{n+1}\right] \tag{B6f}$$

# APPENDIX C: DISCRETIZATION OF ENERGY EQUATION FOR MB MODEL (IMPIT SCHEME) AND OB MODEL UNDER LTE

In what follows we derive the volume scheme to calculate the temperature profile. From Eq. (5.37) we have

$$\frac{\partial\left[\left(\rho C_p\right)_b T\right]}{\partial t} + \nabla.\bar{J} = \left(S_c + S_p T\right) \tag{C1}$$

Term 1

$$\frac{\partial\left[\left(\rho C_p\right)_b T\right]}{\partial t} = \frac{V_b\left[\left(\rho C_p T\right)_{b,i}^{n+1} - \left(\rho C_p T\right)_{b,i}^{n}\right]}{\Delta t} \tag{C2}$$

The flux term *J*, comprises of the convective flux and dispersive flux in the x direction. Thus we propose handling each direction separately using an upwind formulation for the convective flux.

Term 2

Integrating term 2 over control volume

$$\int \nabla.\bar{J}\, dV = \int S\, dV \tag{C3}$$

Applying the divergence theorem

$$\int \nabla.\bar{J}\, dA = \int S\, dV \tag{C4}$$

Thus we have

$$(J.A)_e + (J.A)_w \tag{C5}$$

Noting that the area vectors are defined as follows:

$$A_e = A_x\, i \tag{C6}$$

$$A_w = -A_x\, i \tag{C7}$$

Since, $J = \rho_f C_{\rho f} u_x T - K_{\ell,x} \nabla T$ (C8)

Noting that, $K_{\ell,x} = \lambda_e + \alpha_L u \rho_w C_{pw}$ (C9)

Therefore the total flux on the east phase is:

$$J_e.A_e = \left(\rho_f C_{pf} u_x T\right)_e A_x - K_{\ell,e} A_x \left(\frac{\partial T}{\partial x}\right)_e \quad \text{(C10)}$$

For convenience we introduce the following terms

$$F_e = \left(\rho_f C_{pf} u_x\right)_e A_x, \ and \ F_w = \left(\rho_f C_{pf} u_x\right)_w A_x \quad \text{(C11)}$$

$$D_e = \frac{K_{\ell,e} A_x}{\Delta x}, \ and \ D_w = \frac{K_{\ell,w} A_x}{\Delta x} \quad \text{(C12)}$$

Assuming a step-wise linear profile between adjacent grid blocks for the diffusive flux, the total flus in the east face can be expressed as:

$$J_e.A_e = F_e^{n+1} T_e^{n+1} - D_e^{n+1}(T_{i+1}^{n+1} - T_i^{n+1}) \quad \text{(C13)}$$

Similarly we approximate the temperature and its coefficient at the any face by the temperature at the upstream block. Therefore in the x direction, the total flux is:

$$(J.A)_x = (J.A)_e + (J.A)_w = F_i^{n+1} T_i^{n+1} - D_{i+\frac{1}{2}}^{n+1}(T_{i+1}^{n+1} - T_i^{n+1}) -$$

$$F_{i-1}^{n+1} T_{i-1}^{n+1} - D_{i-\frac{1}{2}}^{n+1}(T_{i-1}^{n+1} - T_i^{n+1}) \quad \text{(C14)}$$

Source term

$$\int S \, dV = \int \left(S_c + S_p T_p\right) dV = \left(S_{c,i} + S_{p,i} T_i^{n+1}\right) V_{b,i} \quad \text{(C15)}$$

Combining all terms:

$$\frac{V_{b.i}\left[\left(\rho C_p T\right)_{b,i}^{n+1} - \left(\rho C_p T\right)_{b,i}^{n}\right]}{\Delta t} + F_i^{n+1} T_i^{n+1} - D_{i+\frac{1}{2}}^{n+1}(T_{i+1}^{n+1} - T_i^{n+1}) - F_{i-1}^{n+1} T_{i-1}^{n+1} -$$

$$D_{i-\frac{1}{2}}^{n+1}(T_{i-1}^{n+1} - T_i^{n+1}) = \left(S_{c,i} + S_{p,i} T_i^{n+1}\right) V_{b,i} \quad \text{(C16)}$$

Rearranging and grouping like terms we have:

$$a_P T_i^{n+1} = a_E T_{i+1}^{n+1} + a_W T_{i-1}^{n+1} + b + a_P^0 T_i^n \quad \text{(C17a)}$$

Where,

$$a_P = \left[\frac{V_{b,i}(\rho C_p)_{b,i}^{n+1}}{\Delta t} + F_i^{n+1} + D_{i+\frac{1}{2}}^{n+1} + D_{i-\frac{1}{2}}^{n+1} - S_{p,i} V_{b,i}\right] \quad \text{(C17b)}$$

$$a_E = D_{i+\frac{1}{2}}^{n+1} \quad \text{(C17c)}$$

$$a_W = D_{i-\frac{1}{2}}^{n+1} + F_{i-1}^{n+1} \tag{C17d}$$

$$a_P^0 = \frac{V_{b,i}(\rho C_p)_{b,i}^n}{\Delta t} \tag{C17e}$$

$$b = S_{c,i} V_{b,i} \tag{C17f}$$

First Block

Introducing the boundary condition described by Eq. (5.7) into Eq. (C16) we have

$$\frac{V_{b.i}\left[(\rho C_p T)_{b,i}^{n+1} - (\rho C_p T)_{b,i}^n\right]}{\Delta t} + F_i^{n+1} T_i^{n+1} - D_{i+\frac{1}{2}}^{n+1}(T_{i+1}^{n+1} - T_i^{n+1}) -$$

$$(\rho_w C_{pw} u T)_{HW} A_x = \left(S_{c,i} + S_{p,i} T_i^{n+1}\right) V_{b,i} \tag{C18}$$

Rearranging and grouping like terms we have:

$$a_P T_i^{n+1} = a_E T_{i+1}^{n+1} + b_i + a_P^0 T_i^n \tag{C19a}$$

Where,

$$a_P = \left[\frac{V_{b,i}(\rho C_p)_{b,i}^{n+1}}{\Delta t} + F_i^{n+1} + D_{i+\frac{1}{2}}^{n+1} - S_{p,i} V_{b,i}\right] \tag{C19b}$$

$$a_E = D_{i+\frac{1}{2}}^{n+1} \tag{C19c}$$

$$a_P^0 = \frac{V_{b,i}(\rho C_p)_{b,i}^n}{\Delta t} \tag{C19d}$$

$$b_i = S_{c,i} V_{b,i} + (\rho_w C_{pw} u T)_{HW} A_x \tag{C19e}$$

Last block

Introducing the boundary condition described by Eq. (5.4) into Eq. (C16) we have

$$\frac{V_{b.i}\left[(\rho C_p T)_{b,i}^{n+1} - (\rho C_p T)_{b,i}^n\right]}{\Delta t} + F_i^{n+1} T_i^{n+1} - F_{i-1}^{n+1} T_{i-1}^{n+1} -$$

$$D_{i-\frac{1}{2}}^{n+1}(T_{i-1}^{n+1} - T_i^{n+1}) = \left(S_{c,i} + S_{p,i} T_i^{n+1}\right) V_{b,i} \tag{C20}$$

Rearranging and grouping like terms we have:

$$a_P T_i^{n+1} = a_w T_{i-1}^{n+1} + b_i + a_P^0 T_i^n \tag{C21a}$$

Where,

$$a_P = \left[\frac{V_{b,i}(\rho C_p)_{b,i}^{n+1}}{\Delta t} + F_i^{n+1} + D_{i-\frac{1}{2}}^{n+1} - S_{p,i}V_{b,i}\right] \quad \text{(C21b)}$$

$$a_w = D_{i-\frac{1}{2}}^{n+1} + F_{i-1}^{n+1} \quad \text{(C21c)}$$

$$a_P^0 = \frac{V_{b,i}(\rho C_p)_{b,i}^{n}}{\Delta t} \quad \text{(C21d)}$$

$$b = S_{c,i}V_{b,i} \quad \text{(C21e)}$$

# APPENDIX D: DISCRETIZATION OF ENERGY EQUATION FOR MB MODEL (IMPIT SCHEME) AND OB MODEL UNDER NOLTE

The energy balance for the rock matrix and fluid phase are discretised using an implicit finite volume scheme. The convective term is approximated using its upstream value.

$$\frac{\partial[(1-\phi)\rho_s C_{ps} T_s]}{\partial t} = \nabla.[(1-\phi)\lambda_s \nabla T_s] + (1-\phi)q_s^{"} + h(T_w - T_s) \quad \text{(D1)}$$

$$\frac{\partial(\phi\rho_w C_{pw} T_w)}{\partial t} + \nabla.\left(\vec{u}_m \rho_w C_{pw} T_w\right) = \nabla.(\phi\lambda_w \nabla T_w) + \phi q_w^{"} + h(T_s - T_w) \quad \text{(D2)}$$

For the one dimensional flow problem and assuming no heat generation in both phases considered, Equations. (D1) and (D2) are written as follows:

$$\frac{\partial}{\partial x}\left[(1-\phi)\lambda_s \frac{\partial T_s}{\partial x}\right] = \frac{\partial[(1-\phi)\rho_s C_{ps} T_s]}{\partial t} + h(T_w - T_s) \quad \text{(D3)}$$

$$\frac{\partial(\phi\rho_w C_{pw} T_w)}{\partial t} + \frac{\partial}{\partial x}\left(u_m \rho_w C_{pw} T_w\right) = \frac{\partial}{\partial x}\left(\phi\lambda_w \frac{\partial T_w}{\partial x}\right) + h(T_s - T_w) \quad \text{(D4)}$$

Eq. (D3) is discretized as follows:

$$\frac{\partial[(1-\phi)\rho_s C_{ps} T_s]}{\partial t} = \frac{\partial}{\partial x}\left[(1-\phi)\lambda_s \frac{\partial T_s}{\partial x}\right] + h(T_w - T_s) \quad \text{(D5)}$$

A closer look at Eq. (D5) shows that the rock matrix energy Eq. is a diffusion Eq. with a source term describing the heat transfer between both phases. Thus Eq. (D5) is of the form:

$$\frac{\partial[(1-\phi)\rho_s C_{ps} T_s]}{\partial t} + \nabla.J_{diff} = (S_c + S_P T_s) \quad \text{(D6)}$$

Integrating Eq. (D5) over a control volume we have for each term the following:

<u>Term1: Transient term</u>

$$\left[\{(1-\phi)\rho_s C_{ps} T_s\}_i^{n+1} - \{(1-\phi)\rho_s C_{ps} T_s\}_i^{n}\right] V_b \quad \text{(D7)}$$

<u>Term 2: Diffusive term</u>

$$A_x\left[\left\{\left\{\frac{\lambda_s(1-\phi)}{\Delta x}\right\}_{i+\frac{1}{2}}^{n+1}(T_{i+1}^{n+1} - T_i^{n+1})\right\} - \left\{\left\{\frac{\lambda_s(1-\phi)}{\Delta x}\right\}_{i-\frac{1}{2}}^{n+1}(T_i^{n+1} - T_{i-1}^{n+1})\right\}\right]\Delta t \quad \text{(D8)}$$

<u>Term 3: Heat transfer terms</u>

$$h_i^{n+1}\Delta t\left(T_{w,i}^{n+1} - T_{s,i}^{n+1}\right)V_b \quad \text{(D9)}$$

Combining all terms together we have

$$\frac{\left[\{(1-\phi)\rho_s C_{ps} T_s\}_i^{n+1} - \{(1-\phi)\rho_s C_{ps} T_s\}_i^{n}\right] V_b}{\Delta t} = h_i^{n+1}\left(T_{w,i}^{n+1} - T_{s,i}^{n+1}\right) V_b +$$

$$\left[D_{s,i+\frac{1}{2}}\left(T_{s,i+1}^{n+1} - T_{s,i}^{n+1}\right) + D_{s,i-\frac{1}{2}}\left(T_{s,i-1}^{n+1} - T_{s,i}^{n+1}\right)\right] \quad \text{(D10)}$$

Where,

$$D_{s,i+\frac{1}{2}} = A_x\left\{\frac{\lambda_s(1-\phi)}{\Delta x}\right\}_{i+\frac{1}{2}}^{n+1}, \text{ and } D_{s,i-\frac{1}{2}} = A_x\left\{\frac{\lambda_s(1-\phi)}{\Delta x}\right\}_{i-\frac{1}{2}}^{n+1} \quad \text{(D11)}$$

Taking like terms we have

$$a_p T_{s,i}^{n+1} = a_w T_{s,i-1}^{n+1} + a_e T_{s,i+1}^{n+1} + a_p^o T_{s,i}^{n} + b \quad \text{(D12)}$$

Where the coefficients are defined below

$$a_w = D_{s,i-\frac{1}{2}} \quad \text{(D13a)}$$

$$a_e = D_{s,i+\frac{1}{2}} \quad \text{(D13b)}$$

$$a_p^o = \frac{V_b}{\Delta t}\left\{(1-\phi)\rho_s C_{ps}\right\}_i^n \quad \text{(D13c)}$$

$$a_p = a_e + a_w + \frac{V_b}{\Delta t}\left\{(1-\phi)\rho_s C_{ps}\right\}_i^{n+1} + h_i^{n+1} V_b \quad \text{(D13d)}$$

$$b = h_i^{n+1} V_b T_{w,i}^{n+1} \quad \text{(D13e)}$$

**Boundary conditions**

First block

$$\frac{\partial T_s}{\partial x} = 0, x = 0, t > 0 \quad \text{(D14)}$$

Therefore we have as follows:

<u>Term1: Transient term</u>

$$\left[\left\{(1-\phi)\rho_s C_{ps} T_s\right\}_i^{n+1} - \left\{(1-\phi)\rho_s C_{ps} T_s\right\}_i^{n}\right] V_b \quad \text{(D15)}$$

<u>Term 2: Diffusive term</u>

$$A_x\left[\left\{\frac{k_s(1-\phi)}{\Delta x}\right\}_{i+\frac{1}{2}}^{n+1}\left(T_{s,i+1}^{n+1}-T_{s,i}^{n+1}\right)\right]\Delta t \tag{D16}$$

Term 3: Heat transfer terms

$$h_i^{n+1}\Delta t\left(T_{w,i}^{n+1}-T_{s,i}^{n+1}\right)V_b \tag{D17}$$

Combining all terms together we have

$$\frac{\left[\left\{(1-\phi)\rho_s C_{ps}T_s\right\}_i^{n+1}-\left\{(1-\phi)\rho_s C_{ps}T_s\right\}_i^{n}\right]V_b}{\Delta t}=D_{s,i+\frac{1}{2}}\left(T_{s,i+1}^{n+1}-T_{s,i}^{n+1}\right)+$$

$$h_i^{n+1}\left(T_{w,i}^{n+1}-T_{s,i}^{n+1}\right)V_b \tag{D18}$$

Where,

$$D_{s,i+\frac{1}{2}}=A_x\left\{\frac{\lambda_s(1-\phi)}{\Delta x}\right\}_{i+\frac{1}{2}}^{n+1} \tag{D19}$$

Taking like terms we have

$$a_pT_{s,i}^{n+1}=a_eT_{s,i+1}^{n+1}+a_p^oT_{s,i}^{n}+b \tag{D20}$$

with the coefficients defined as below

$$a_e=D_{s,i+\frac{1}{2}} \tag{D21a}$$

$$a_p^o=\frac{V_b}{\Delta t}\left\{(1-\phi)\rho_s C_{ps}T_s\right\}_i^n \tag{D21b}$$

$$a_p=a_e+\frac{V_b}{\Delta t}\left\{(1-\phi)\rho_s C_{ps}\right\}_i^{n+1}+h_i^{n+1}V_b \tag{D21c}$$

$$b=h_i^{n+1}V_bT_{w,i}^{n+1} \tag{D21d}$$

Last Block

$$\frac{\partial T}{\partial x}=0,\ x=L,\ t>0 \tag{D22}$$

Term1: Transient term

$$\left[\left\{(1-\phi)\rho_s C_{ps}T_s\right\}_i^{n+1}-\left\{(1-\phi)\rho_s C_{ps}T_s\right\}_i^{n}\right]V_b \tag{D23}$$

Term 2: Diffusive term

$$A_x\left\{\frac{\lambda_s(1-\phi)}{\Delta x/2}\right\}_{i-\frac{1}{2}}^{n+1}\left(T_{s,i}^{n+1}-T_{s,i-1}^{n+1}\right)\Delta t \tag{D24}$$

Term 3: Heat transfer terms

$$h_i^{n+1}\Delta t\left(T_{w,i}^{n+1}-T_{s,i}^{n+1}\right)V_b \tag{D25}$$

Combining all terms together we have

$$\frac{\left[\left\{(1-\phi)\rho_s C_{ps}T_s\right\}_i^{n+1}-\left\{(1-\phi)\rho_s C_{ps}T_s\right\}_i^{n}\right]V_b}{\Delta t}=D_{s,i-\frac{1}{2}}\left(T_{s,i-1}^{n+1}-T_{s,i}^{n+1}\right)+$$

$$h_i^{n+1}\left(T_{w,i}^{n+1}-T_{s,i}^{n+1}\right)V_b \tag{D26}$$

Taking like terms we have

$$a_p T_s^{n+1}=a_w T_{s,i-1}^{n+1}+a_p^o T_{s,i}^{n}+b \tag{D27}$$

Where the coefficients are defined below

$$a_w=D_{s,i-\frac{1}{2}} \tag{D28a}$$

$$a_p^o=\frac{V_b}{\Delta t}\left[(1-\phi)\rho_s C_{ps}\right]_i^n \tag{D28b}$$

$$a_p=a_w+\frac{V_b}{\Delta t}\left[(1-\phi)\rho_s C_{ps}\right]_i^{n+1}+h_i^{n+1}V_b \tag{D28c}$$

$$b=h_i^{n+1}V_b T_{w,i}^{n+1} \tag{D28d}$$

Now, the fluid phase energy Eq. is discretised, a closer look at Eq. B4 describes a convection-dispersion Eq. with a source term.

$$\frac{\partial[\phi\rho_w C_{pw}T_w]}{\partial t}+\nabla.J_{adv+diff}=(S_c+S_P T_w) \tag{D29}$$

Now, integrating Eq. (D29) over a control volume we have as follows:

Term 1: Transient term

$$\left[\left(\phi\rho_w C_{pw}T_w\right)_i^{n+1}-\left(\phi\rho_w C_{pw}T_w\right)_i^{n}\right]V_b \tag{D30}$$

Term2: Convective term

$$\left(u_m\rho_w C_{pw}T_w\right)_{i+\frac{1}{2}}^{n+1}-\left(u_m\rho_w C_{pw}T_w\right)_{i-\frac{1}{2}}^{n+1} \tag{D31}$$

Using the upstream weighting to evaluate coefficients at the face of control volume in Eq. (D31) simplifies

$$F_e T_{w,i}^{n+1}-F_w T_{w,i-1}^{n+1} \tag{D32}$$

Where the coefficients defined above are as shown below,

$$F_e = \left(u_m \rho_w C_{pw}\right)_i^{n+1}, \text{ and } F_w = \left(u_m \rho_w C_{pw}\right)_{i-1}^{n+1} \tag{D33}$$

Term 3: Diffusive term

$$A_x \left[\left\{\left(\frac{\lambda_w \phi}{\Delta x}\right)_{i+\frac{1}{2}}^{n+1}\left(T_{w,i+1}^{n+1} - T_{w,i}^{n+1}\right)\right\} - \left\{\left(\frac{\lambda_w \phi}{\Delta x}\right)_{i-\frac{1}{2}}^{n+1}\left(T_{w,i}^{n+1} - T_{w,i-1}^{n+1}\right)\right\}\right]\Delta t \tag{D34}$$

Term 4: Heat transfer terms

$$h_i^{n+1}\Delta t\left(T_{s,i}^{n+1} - T_{w,i}^{n+1}\right)V_b \tag{D35}$$

Combining all terms together we have

$$\frac{\left[\left(\phi \rho_w C_{pw} T_w\right)_i^{n+1} - \left(\phi \rho_w C_{pw} T_w\right)_i^{n}\right]V_b}{\Delta t} + \left(F_e T_{w,i}^{n+1} - F_w T_{w,i-1}^{n+1}\right) =$$

$$\left[D_{w,i+\frac{1}{2}}\left(T_{w,i+1}^{n+1} - T_{w,i}^{n+1}\right) + D_{w,i-\frac{1}{2}}\left(T_{w,i-1}^{n+1} - T_{w,i}^{n+1}\right)\right] + h_i^{n+1}\left(T_{w,i}^{n+1} - T_{w,i}^{n+1}\right)V_b \tag{D36}$$

Where,

$$D_{w,i\mp\frac{1}{2}} = A_x\left(\frac{\lambda_f \phi}{\Delta x}\right)_{i\mp\frac{1}{2}}^{n+1} \tag{D37}$$

Taking like terms we have

$$a_p T_w^{n+1} = a_w T_{w,i-1}^{n+1} + a_e T_{w,i+1}^{n+1} + a_p^o T_{f,i}^{n} + b \tag{D38}$$

Where the coefficients are defined below

$$a_w = D_{w,i-\frac{1}{2}} + F_w \tag{D39a}$$

$$a_e = D_{w,i+\frac{1}{2}} \tag{D39b}$$

$$a_p^o = \frac{V_b}{\Delta t}\left(\phi \rho_w C_{pw}\right)_i^{n} \tag{D39c}$$

$$a_p = D_{w,i+\frac{1}{2}} + D_{w,i-\frac{1}{2}} + F_e + \frac{V_b}{\Delta t}\left(\phi \rho_w C_{pw}\right)_i^{n+1} + h_i^{n+1}V_b \tag{D39d}$$

$$b = h_i^{n+1} V_b T_{s,i}^{n+1} \tag{D39e}$$

**Boundary Conditions**

First Block

$$\left(\rho_w C_{pw} u_m T_w\right)_{inlet} = \rho_w u_m C_{pw} T_w - \phi\lambda_w \frac{\partial T_w}{\partial x}, x = 0, \;\; t > 0 \tag{D40}$$

Now, integrating Eq. (D29) over the control volume at the first block results in:

Term 1: Transient term

$$\left[\left(\phi\rho_w C_{pw} T_w\right)_i^{n+1} - \left(\phi\rho_w C_{pw} T_w\right)_i^n\right] V_b \tag{D41}$$

Term2: Convective term

$$\left(u_m \rho_m C_{pw} T_w\right)_{i+\frac{1}{2}}^{n+1} - \left(u_m \rho_w C_{pw} T_w\right)_{inlet}^{n+1} \tag{D42}$$

Using the upstream weighting to evaluate coefficients at the face of control volume in Eq. (D42) simplifies

$$\left(F_e T_{w,i}^{n+1} - F_{inlet} T_{inlet}\right) \tag{D43}$$

Where,

$$F_e = \left(u_m \rho_w C_{pw}\right)_i^{n+1}, \text{ and } F_{inlet} = \left(u_m \rho_w C_{pw}\right)_{inlet}^{n+1} \tag{D44}$$

Term 3: Diffusive term

$$A_x\left[\left\{\left(\frac{\lambda_w \phi}{\Delta x}\right)_{i+\frac{1}{2}}^{n+1}\left(T_{w,i+1}^{n+1} - T_{w,i}^{n+1}\right)\right\} - \left\{\left(\frac{\lambda_w \phi}{\Delta x}\right)_{i-\frac{1}{2}}^{n+1}\left(T_{w,i}^{n+1} - T_{inlet}\right)\right\}\right]\Delta t \tag{D45}$$

Term 4: Heat transfer terms

$$h_i^{n+1}\Delta t\left(T_{s,i}^{n+1} - T_{w,i}^{n+1}\right)V_b \tag{D46}$$

Combining all terms together we have

$$\frac{\left[\left(\phi\rho_w C_{pw} T_w\right)_i^{n+1} - \left(\phi\rho_w C_{pw} T_w\right)_i^n\right] V_b}{\Delta t} + \left(F_e T_{w,i}^{n+1} - D_{w,i+\frac{1}{2}}\right) =$$

$$A_x\left(\rho_w C_{pw} u_m T_w\right)_{inlet} + \; h_i^{n+1}\left(T_{w,i}^{n+1} - T_{w,i}^{n+1}\right)V_b \tag{D47}$$

Where,

$$D_{w,i+\frac{1}{2}} = A_x\left(\frac{\lambda_w \phi}{\Delta x}\right)_{i+\frac{1}{2}}^{n+1} \tag{D48}$$

$$D_{w,i-\frac{1}{2}} = A_x \left(\frac{\lambda_w \phi}{\Delta x}\right)_{i-\frac{1}{2}}^{n+1} \tag{D49}$$

Taking like terms we have

$$a_p T_{w,i}^{n+1} = a_e T_{w,i+1}^{n+1} + a_p^o T_{w,i}^n + b \tag{D50}$$

Where the coefficients are defined below

$$a_e = D_{w,i+\frac{1}{2}} \tag{D51a}$$

$$a_p^o = \frac{V_b}{\Delta t}\left(\phi \rho_w C_{pw}\right)_i^n \tag{D51b}$$

$$a_p = D_{w,i+\frac{1}{2}} + F_e + \frac{V_b}{\Delta t}\left(\rho_w C_{pw} \phi\right)_i^{n+1} + h_i^{n+1} V_b \tag{D51c}$$

$$b = h_i^{n+1} V_b T_{s,i}^{n+1} + A_x \left(\rho_w C_{pw} u_m T_w\right)_{inlet} \tag{D51d}$$

Last Block

$$\frac{\partial T_w}{\partial x} = 0, \quad x = L, \ t > 0 \tag{D52}$$

Now, integrating Eq. (D29) over the last control volume and applying Eq. (D52) we have as follows:

Term 1: Transient term

$$\left[\left(\phi \rho_w C_{pw} T_w\right)_i^{n+1} - \left(\phi \rho_w C_{pw} T_w\right)_i^n\right] V_b \tag{D53}$$

Term2: Convective term

$$\left(u_m \rho_w C_{pw} T_w\right)_{i+\frac{1}{2}}^{n+1} - \left(u_m \rho_w C_{pw} T_w\right)_{i-\frac{1}{2}}^{n+1} \tag{D54}$$

Using the upstream weighting to evaluate coefficients at the face of control volume in Eq. (D54) simplifies to:

$$F_e T_{w,i}^{n+1} - F_w T_{w,i-1}^{n+1} \tag{D55}$$

Where,

$$F_e = \left(u_m \rho_w C_{pw}\right)_i^{n+1}, \text{ and } F_w = \left(u_m \rho_w C_{pw}\right)_{i-1}^{n+1} \tag{D56}$$

Term 3: Diffusive term

$$A_x\left[\left\{\left(\frac{\lambda_w\phi}{\Delta x}\right)_{i+\frac{1}{2}}^{n+1}\left(T_{w,i-1}^{n+1}-T_{w,i}^{n+1}\right)\right\}\right]\Delta t \tag{D57}$$

Term 4: Heat transfer terms

$$h_i^{n+1}\Delta t\left(T_{s,i}^{n+1}-T_{w,i}^{n+1}\right)V_b \tag{D58}$$

Combining all terms together we have

$$\frac{\left[\left(\phi\rho_w C_{pw}T_w\right)_i^{n+1}-\left(\phi\rho_w C_{pw}T_w\right)_i^{n}\right]V_b}{\Delta t}+F_eT_{w,i}^{n+1}-F_wT_{w,i-1}^{n+1}=$$

$$D_{w,i-\frac{1}{2}}\left(T_{w,i-1}^{n+1}-T_{w,i}^{n+1}\right)+h_i^{n+1}\left(T_{s,i}^{n+1}-T_{w,i}^{n+1}\right)V_b \tag{D59}$$

Where,

$$D_{w,i-\frac{1}{2}}=A_x\left(\frac{\lambda_w\phi}{\Delta x}\right)_{i-\frac{1}{2}}^{n+1} \tag{D60}$$

Taking like terms we have

$$a_pT_{w,i}^{n+1}=a_wT_{w,i-1}^{n+1}+a_p^oT_{w,i}^{n}+b \tag{D61}$$

Where the coefficients are defined below

$$a_w=D_{w,i-\frac{1}{2}}+F_w \tag{D62a}$$

$$a_p^o=\frac{V_b}{\Delta t}\left(\phi\rho_w C_{pw}\right)_i^n \tag{D62b}$$

$$a_p=D_{w,i-\frac{1}{2}}+F_e+\frac{V_b}{\Delta t}\left(\phi\rho_w C_{pw}\right)_i^{n+1}+h_i^{n+1}V_b \tag{D62c}$$

$$b=h_i^{n+1}V_bT_{s,i}^{n+1} \tag{D62d}$$

# APPENDIX E: MATLAB PROGRAMMING CODE FOR MEMORY BASED MODEL UNDER LTE

```
%function Memory_Model_LTE
% Written by: Obembe Abiola David
% Date       : 13-03-2016
% Scheme     : IMPIT with constant q Pressure BC
%=========================================================================
% solves the memory-based non-isothermal fluid flow problem
clear all
close all
clc
Format long
global    K_ref Phi_ref P_ref P_out rho_Rock T_ref  c_R a b ...
c m_factor
data       = load('CoreData.txt');
L          = data(1);
w          = data(2);
H          = data(3);
Nx         = data(4);
Ny         = data(5);
Nz         = data(6);
P_ref      =  data(7)*0.068046;
P_out      = data(9);
```

```
Kxi         =  data(10);
Phi_ref     = data(12);
q           = data(14);
T_ref       = data(15)+273.15;
T_ini       = data(16)+273.15;
Tinlet1     = data(17)+273.15;
Tt          = data(18)*60;
dt          = data(19);
c_R         = data(26);
Cpr         = data(28);
kr          = data(31);
rho_Rock    = data(36)*1000;
aDL         = data(37);
betacp      = data(49);
alphacp     = data(53);
alpha1      = data(54);
m_factor    = data(56);
a           = data(57);
b           = data(58);
c           = data(59);
Tinfinity   = data(61)+273.15;
Uoverall    = data(62);
%% lab units
Lp    = 2.54*L;
wp    = 2.54*w;
```

```
Hp    = 2.54*H;
qp1   = 0.017*q;
K_ref = Kxi *1e-3;
Poutp = P_out*0.068046;
% spe metric units
Lc    = 0.0254*L;
wc    = 0.0254*w;
Hc    = 0.0254*H;
qc1   = 1.7e-8*q;
%% Grid size
NyNz  = Ny*Nz;
M     = Nx*Ny*Nz;
Mp1   = M+1;
Mp2   = M+2;
Nt    = Tt/dt;
%% More variables
dxp   = Lp/Nx;
Axp   = (pi*wp^2)/4;
Vbp   = dxp*Axp;
dxc   = Lc/Nx;
Axc   = (pi*wc^2)/4;
Vbc   = dxc*Axc;
PVt1m = Axc*Lc*Phi_ref;
%% Generate Initial porosity and permeability
X     = 0.5*dxp:dxp:Lp-0.5*dxp;
```

```
%% Generates Initial Pressure and Temperature
P_pi      = Poutp*ones(M,1);
T_pi      = T_ini*ones(M,1);
Pcomp     = [P_pi; Poutp;Poutp];
%% Iteration Options
omega    = 0.85;
err_tol  = 1e-6;
%% Pre computaion of wk and coeffiecient
D     = 1/(dt^(1-alpha1));
gtabl = ones(1,Nt);
for r         = 2:Nt+1
   nn         = r -1;
   gtabl(1,r)= -gtabl(1,r-1)*(2-alpha1-nn)/nn;
end
Shmemory = 65*60;
%% Rock and fluid Initial Properties
Phii      = Civan_PORO(P_pi,T_pi);                    pressure and temperature
Kxi       = Civan_PERM(T_pi);
rhofi     = XSteam('rho_pT',P_pi(1)*1.01325,T_pi(1)-273.15);
muii      = XSteam('my_pT',P_pi(1)*1.01325,T_pi(1)-273.15)*1e3;
rhosc     = XSteam('rho_pT',P_ref*1.01325,T_ref-273.15);
%% Model Initialization
Ue1m          = zeros(M,Nt+1);
Uw1m          = zeros(M,Nt+1);
Upp1m         = zeros(M,Nt+1);
```

```
qinj          = zeros(Nt+1,1);
pvinj1m       = zeros(Nt+1,1);
PoroT1m       = zeros(M,Nt+1);
Perm1m        = zeros(M,Nt+1);
Muuw1m        = zeros(M,Nt+1);
Rhoow1m       = zeros(M,Nt+1);
Kimp1m        = zeros(M,Nt+1);
Eta1m         = zeros(M,Nt+1);
Ue1m(:,1)     = 0;
Uw1m(:,1)     = 0;
Upp1m(:,1)    = 0;
Kimp1m(:,1)   = 1;
Muuw1m(:,1)   = muii;
Rhoow1m(:,1)  = rhofi;
P1m           = P_pi;
T1m           = T_pi;
PoroT1m(:,1)  = Phii;
Perm1m(:,1)   = Kxi*1e3;
Eta1m(:,1)    = Kxi./muii ;
Bwater        = (rhosc./rhofi)*ones(M,1);
Txx           = ((betacp*(Kxi./muii)*Axp*D)./(Bwater*dxp));
t             = 0;
Pinjj1m       = P1m(1);
%% Solve Pressure Equation
tic
```

```
for n     = 1:Nt  % time loop
    np1  = n+1;
    err  = 2*err_tol;
    iter = 0;
    timer= n*dt ;
    Pn   = P1m(:,n);
    Piter= Pn;                Pnew_iter = Pn;
    Tn   = T1m(:,n);          Tnew_iter = Tn;
    Titer= Tn;
   while (iter < 2) || (err>err_tol)
            Poron   = Civan_PORO(Pn,Tn);
            Poronp1 = Civan_PORO(Piter,Tn);
            KKx     = Civan_PERM(Tn);
            for  nn = 1: M
               rhowsu(nn) = XSteam('rho_pT',Pn(nn)*1.01325, Tn(nn)-273.15);
               rhows(nn)  = XSteam('rho_pT',Pn(nn)*1.01325, Tn(nn)-273.15);
               bwu(nn)    = rhosc/rhowsu(nn);
               bwn(nn)    = rhosc/rhows(nn);
               wmui(nn)   = XSteam('my_pT',Pn(nn)*1.01325, Tn(nn)-273.15)*1e3;
            end
            Bonp1   = bwu(:);
            Bon     = bwn(:);
            miunp1  = wmui(:);
            AccP    = (Vbp/(alphacp*dt))*((Poronp1./Bonp1)-((Poron./Bon)));
            eeta    = (KKx./miunp1)*(timer).^(1-alpha1);
```

```
Tx      =  ( betacp*eeta*Axp*D)./(Bon*dxp);
for m = 1:M
    mp1   = m+1;  mm1   = m-1;
    Txmmh = 0;    Txmph = 0;
            if m == 1
               Sum(m)   = 0;
               for k     = 1:1:min(Shmemory/dt,n);          % past history effect
                  wk    = gtabl(1,k+1);                    % wk term
                  Sum(m)  =  Sum(m)  +  (wk*(Txx(m,n-k+1).*(P1m(mp1,n-k+1)-
P1m(m,n-k+1))));
               end
            Txmph    = Tx(m);
            % coeffcients of discretized equation
            ae      = Txmph;
            ap      = (Txmph+Txmmh) ;
            ap0     = 0;
            bb      =  (qp1/Bonp1(m))+Sum(m)-AccP(m);

           %Pressure SOR
            Pnew_iter(m) = (ae*Piter(mp1)+(ap0*Pn(m))+ bb)/ap;
            Pnew_iter(m) = (1-omega)*Piter(m)+omega*Pnew_iter(m);
        elseif m > 1 && m < M

            Sum(m) = 0;
            for k  = 1:1:min(Shmemory/dt,n);           % past history effect
```

```
                wk = gtabl(1,k+1);                  % wk term
                Sum(m) = Sum(m) + (wk*((Txx(mm1,n-k+1).*(P1m(mm1,n-k+1)-
P1m(m,n-k+1)))+ ...
                        (Txx(m,n-k+1).*(P1m(mp1,n-k+1)-P1m(m,n-k+1)))));
            end
            Txmph    = Tx(m);
            Txmmh    = Tx(mm1);
            % discretized coeffcients
            aw      = Txmmh;
            ae      = Txmph;
            ap      = (Txmph+Txmmh);
            ap0     = 0 ;
            bb      =  Sum(m)-AccP(m);
            Pnew_iter(m)  = (aw*Pnew_iter(mm1)+ae*Piter(mp1)+(ap0*Pn(m))+
bb)/ap;
            Pnew_iter(m)  = (1-omega)*Piter(m)+omega*Pnew_iter(m);
        else
            Sum(m) = 0;
            for k  = 1:1:min(Shmemory/dt,n);          % past history effect
                wk = gtabl(1,k+1);                  % wk term
                Sum(m) = Sum(m) + (wk*((Txx(mm1,n-k+1).*(P1m(mm1,n-k+1)-
P1m(m,n-k+1)))+ ...
                        (2*Txx(m,n-k+1).*(Poutp-P1m(m,n-k+1)))));
            end
            Txmph    = 2*Tx(m);
            Txmmh    = Tx(mm1);
```

```
            % discretized coeffcients
            aw      = Txmmh;
            ap      = (Txmph+ Txmmh);
            ap0     = 0;
            bb      =  (Txmph*Poutp)+Sum(m)-AccP(m);
            Pnew_iter(m)  = (aw*Pnew_iter(mm1)+(ap0*Pn(m))+ bb)/ap;
            Pnew_iter(m)  = (1-omega)*Piter(m)+omega*Pnew_iter(m);
             end
 end
Pnew   = Pnew_iter;
Sums   = 0;
for k  = 1:1:min(Shmemory/dt,n);          % past history effect
      wk = gtabl(1,k+1);                  % wk term
      Sums = Sums + wk*(2*Txx(1,n-k+1)*(Pinjj1m(1,n-k+1)-P1m(1,n-k+1)));
   end
Pinjp  = Pnew(1)+((qp1-Sums)/(2*Tx(1)));
% Parameters for energy calculation in metric unit
Phinp1u   = Civan_PORO(Pn,Tn);
for j = 1:M
   wmuiu(j)   =  XSteam('my_pT',Pn(j)*1.01325, Tn(j)-273.15)*1e3;
   rhowsuu(j) = XSteam('rho_pT',Pn(j)*1.01325, Tn(j)-273.15);
   kwater(j)  = XSteam('tc_pT',Pn(j)*1.01325, Tn(j)-273.15);
   Cpwatu(j)  = XSteam('Cp_pT',Pn(j)*1.01325, Tn(j)-273.15)*1e3;
   Cpwatnu(j) = XSteam('Cp_pT',Pn(j)*1.01325, Tn(j)-273.15)*1e3;
   Bwwu(j)    = rhosc/rhowsuu(j);
```

```
    end
    visup     = wmuiu(:);
    rhowup    = rhowsuu(:);
    kwu       = kwater(:);
    Cpwnp1    = Cpwatu(:);
    Cpwn      = Cpwatnu(:);
    rhownotup = rhows(:);
    Bonp1u    = Bwwu(:);
    Keffec    = (kwu.* Phinp1u) + (kr*(1-Phinp1u));
     eetau    = (KKx./visup)*(timer).^(1-alpha1);
     Txu      =  ( betacp*eetau*Axp*D)./(Bonp1u*dxp);
     Pnewcomp   = [ Pnew; Poutp; Pinjp];
     Pmemory    = [ Pcomp Pnewcomp];
     Pdot       = fractional_der_press(Pmemory,alpha1);
     rhowcpbn   = ((rhownotup.*Cpwn.*Poron)+(rho_Rock*Cpr*(1-Poron)));
     rhowcpbnp1 = ((rhowup.*Cpwnp1.*Phinp1u)+(rho_Rock*Cpr*(1-Phinp1u)));
     AccTn      = (Vbc*rhowcpbn)/dt;
     AccTnp1    = (Vbc*rhowcpbnp1)/dt;
for m = 1:M
        mp1  = m+1;        mm1  = m-1;

        Scc  = 0;          Spp  = 0;
        De   = 0;          Dw   = 0;
        bbT  = 0;
      if m == 1
```

```
% Obtain flow velocity in m/s
ue(m)    = -0.01*(Txu(m).*(Pdot(mp1)-Pdot(m)))/(Axp);
uw(m)      = -0.01*(Txu(m).*(Pdot(m)-Pdot(Mp2)))/(0.5*Axp);
up(m)    = 0.5*(ue(m)+uw(m));
% Diffusion term
lamdaef  = harmmean([Keffec(m),Keffec(mp1)]);
Disperse = aDL*abs(ue(m))*rhowup(m)*Cpwnp1(m);                    % Dispersiivty in east face
De       =  ((lamdaef+Disperse)*Axc)/dxc;
Fe       = Axc*ue(m)*rhowup(m)*Cpwnp1(m);
% Implement Boundary condition
rhowCpin                =           XSteam('rho_pT',Pnew(1)*1.01325,Tinlet1-273.15)*XSteam('Cp_pT',Pnew(1)*1.01325,Tinlet1-273.15)*1e3;
sourceT   = rhowCpin*Axc*uw(m)*Tinlet1;
% Heat loss terms
Scc   = (2*pi*(Hc/2)*dxc*Uoverall*(Tinfinity+273.15))/Vbc;
Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;
% Coeffiecient
aE  = De;
aP  = ( AccTnp1(m)+Fe+De+Dw-(Spp*Vbc));
aP0 =  AccTn(m);
bbT = (Scc*Vbc) +(sourceT) ;
%%  Temperature SOR
 Tnew_iter(m)  = (aE*Titer(mp1)+(aP0*Tn(m))+bbT)/aP;
 Tnew_iter(m)  = (1-omega)*Titer(m)+omega*Tnew_iter(m);
 elseif m > 1 && m < M
```

```
ue(m)    = -0.01*(Txu(m).*(Pdot(mp1)-Pdot(m)))/(Axp);
uw(m)    = -0.01*(Txu(mm1).*(Pdot(m)-Pdot(mm1)))/(Axp);
up(m)    = 0.5*(ue(m)+uw(m));
% Diffusion term
lamdaef  = harmmean([Keffec(m),Keffec(mp1)]);
lamdawf  = harmmean([Keffec(m),Keffec(mm1)]);
Disperse =  aDL*abs(ue(m))*rhowup(m)*Cpwnp1(m);
Dispersw =  aDL*abs(uw(m))*rhowup(m)*Cpwnp1(mm1);
De       =  ((lamdaef+Disperse)*Axc)/dxc;
Dw       = ((lamdawf+Dispersw)*Axc)/dxc;
% Convection term
Fe     =  Axc*ue(m)*rhowup(m)*Cpwnp1(m);                    %
Fw       =  Axc*uw(m)*rhowup(m)*Cpwnp1(mm1);
 % Heat loss terms
 Scc   = (2*pi*(Hc/2)*dxc*Uoverall*(Tinfinity+273.15))/Vbc;
 Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;
% Coeffiecients
aW = Dw+Fw;
aE = De;
aP = (AccTnp1(m)+Fe+De+Dw-(Spp*Vbc));
aP0= AccTn(m);
bbT = Scc*Vbc;
Tnew_iter(m)                                                =
(aW*Tnew_iter(mm1)+aE*Titer(mp1)+(aP0*Tn(m))+bbT)/aP;
Tnew_iter(m)  = (1-omega)*Titer(m)+omega*Tnew_iter(m);
```

```
    else
            ue(m)    = -0.01*(Txu(m).*(Pdot(mp1)-Pdot(m)))/(0.5*Axp);
            uw(m)    = -0.01*(Txu(mm1).*(Pdot(m)-Pdot(mm1)))/(Axp);
            up(m)    = 0.5*(ue(m)+uw(m));
            lamdawf  =  harmmean([Keffec(m),Keffec(mm1)]);
            Dispersw =  aDL*abs(uw(m))*rhowup(mm1)*Cpwnp1(mm1);
            Dw       = ((lamdawf+Dispersw)*Axc)/dxc;
            Fe       =  Axc*ue(m)*rhowup(m)*Cpwnp1(m);
            Fw       =  Axc*uw(m)*rhowup(mm1)*Cpwnp1(mm1);
             Scc   = (2*pi*(Hc/2)*dxc*Uoverall*(Tinfinity+273.15))/Vbc;
             Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;
            aW = Dw+Fw;
            aP = (AccTnp1(m)+De+Fe+Dw-(Spp*Vbc));
            aP0=  AccTn(m);
            bbT = Scc*Vbc;
             Tnew_iter(m)  = (aW*Tnew_iter(mm1)+(aP0*Tn(m))+bbT)/aP;
             Tnew_iter(m)  = (1-omega)*Titer(m)+omega*Tnew_iter(m);
    end
end
  normP  =  norm(((Pnew_iter-Piter)/Piter),inf)
  Piter  = Pnew_iter;
  normT =  norm(((Tnew_iter-Titer)/Titer),inf)
  Titer = Tnew_iter;
  disp('error is :')
  err    = max([normP,normT])
```

```
        iter = iter+1
    end
    T1m      = [T1m Titer];
    P1m      = [ P1m Piter];   Pinjj1m = [Pinjj1m Pinjp]  ;
    KKu      = Civan_PERM(Titer);
     for i= 1:M
       rhownew(i) = XSteam('rho_pT',Piter(i)*1.01325,Titer(i)-273.15);
       Bwnew(i)   = rhosc./rhownew(i);
       muinew(i)  = XSteam('my_pT',Piter(i)*1.01325,Titer(i)-273.15)*1e3;
       eetanew(i) = (KKu(i)/muinew(i))*(timer).^(1-alpha1);
       Tx(i)      = ( betacp*eetanew(i)*Axp*D)./(Bwnew(i)*dxp);
    end
    Bwater   = [ Bwater Bwnew'];
    Txx      = [ Txx  Tx];
    Eta1m(:,np1)= eetanew';
    PoroT1m(:,np1) = Civan_PORO(Piter,Titer);
    Perm1m(:,np1)  = KKu*1e3;
    Muuw1m(:,np1)  = muinew';
    Rhoow1m(:,np1) = rhownew';
    Ue1m(:,np1)    = ue(:);
    Uw1m(:,np1)    = uw(:);
    Upp1m(:,np1)   = up(:);
    Kimp1m(:,np1)  = KKu./Kxi;
    qinj(np1)      = qc1*timer;
    pvinj1m(np1)   = (qc1*timer)/PVt1m;
```

```
   disp('Volume of water injected, (m^3)')
   Q           = qc1*(timer)
   disp('Simulation time in sec')
   t           = (t+dt)
end% time loop ends
toc
  tplot   = (0:dt:Tt)/60;
```

# APPENDIX F: MATLAB PROGRAMMING CODE FOR MODIFIED OB MODEL UNDER LTE

```
%function Modified_Boussinesq_Model
% Written by: Obembe Abiola David
% Date       : 13-03-2016
% Scheme     : IMPIT with constant flowrate BC
%=========================================================================
% Solves the memory-based non-isothermal fluid flow problem
clear all
close all
clc
format long
global  K_ref Phi_ref P_ref P_out rhow_ref rho_Rock T_ref c_R  a b ...
       c m_factor gamma_0 beta_0 miu_ref a_0 b_0 c_0 d_0 e_0 f_0 kw Cpw
data       = load('DataforCore.txt');
L          = data(1);                % core length (inches)
w          = data(2);                % core width(inches)
H          = data(3);                % core thickness(inches)
Nx         = data(4);                % Nx grid number
Ny         = data(5);                % Ny grid number
Nz         = data(6);                % Nz grid number
P_ref      = data(7)*0.0689476;      % Reference pressure in atm
P_out      = data(9);                % Out let pressure in psia
```

```
Kxi        = data(10);                % Initial measured Core Permeability at ref cond in mD
miu_ref    = data(11);                % viscosity of water at ref conditions in cp
Phi_ref    = data(12);                % Initial mean porosity at ref pressure and temperature
delphi     = data(13);                % heterogeity factor s.d of porosity
q          = data(14);                % flowrate in ml/min
T_ref      = data(15)+273.15;         % Reference Temperature degree K
T_ini      = data(16)+273.15;         % Initial Temperature degree C
Tinlet3    = data(17)+273.15;         % Hot water temperature degree C
Tt         = data(18)*60;             % Simulation time in secs
dt         = data(19);                % time step in secs
rhow_ref   = data(20);                % Reference density if water in g/cm3
gamma_0    = data(23)*1013250;        % Water density isothermal compressiblity 1/atm
beta_0     = data(24);                % Water density thermal expansion coefficient 1/K
g          = data(25)*100;            % Acceleration due to gravity cm/s^2
c_R        = data(26);                % Rock isothermal compressibility coefficient in 1/atm
Cpr        = data(28);                % Specific heat Capacity of rock (J/Kg/K)
Cpw        = data(29);                % Reference Specific heat Capacity of water (J/Kg/K)
kw         = data(30);                % Reference thermal Conductivity of water (W/m/K)
kr         = data(31);                % Reference thermal Conductivity of rock(W/m/K)
f_0        = data(32)*1013250;        % Water Thermal conductivity isothermal expansion
e_0        = data(33);                % Water thermal conductivity thermal expansion
b_0        = data(34)*1013250;        % Water  Specific heat  isothermal expansion
a_0        = data(35);                % Water Specific heat thermal expansion coefficient
rho_Rock   = data(36)*1000;           % Reference density if solid matrix in kg/m3
aDL        = data(37);                % Coefficient of Long. Thermal dispersion in m
```

```
betacc     = data(48);                % betac for metric unit
betacp     = data(49);                % betac for lab unit
c_0        = data(50);                % Thermal part of viscosity in 1/K
d_0        = data(51)*1013250;        % Isothermal part of water viscosity in atm
alphacc    = data(52);                % alphacc
alphacp    = data(53);                % alphacp
alpha3     = data(54);                % Fractional order derivative
P_conf     = data(55);                % Confining pressure in psia
m_factor   = data(56);                % Cementation factor
a          = data(57);                % Empirical constant a for Modified Kozeny-Carman
b          = data(58);                % Empirical constant b for Modified Kozeny-Carman
c          = data(59);                % Empirical constant c for Modified Kozeny-Carman
Pinlet     = data(60);                % Injection pressure in psia
Tinfinity  = data(61);                % Ambient temperature in degree C
Uoverall   = data(62);
%% lab units
Lp    = 2.54*L;                       % core length (centimeters)
wp    = 2.54*w;                       % core width(centimeters)
Hp    = 2.54*H;                       % core thickness(centimeters)
qp3   = 0.017*q;                      % flow rate (cc/s)
K_ref = Kxi *1e-3;                    % Initial Measuerd Core Permeability in Darcy
Poutp  = P_out*0.068046;              % Backpressure/Outlet pressure in atm
% spe metric units
Lc   = 0.0254*L;                      % core length (meters)
wc   = 0.0254*w;                      % core width(meters)
```

```
Hc   = 0.0254*H;                % core thickness(meters)
qc3  = 1.7e-8*q;                % flow rate (cubic meters/s)
%% Grid size
NyNz= Ny*Nz;
M   = Nx*Ny*Nz;
Mp1 = M+1; mP2  = M+2;
Nt  = Tt/dt;
%% More variables
dxp = Lp/Nx;
Axp = pi*(wp)^2/4;
Vbp =  Axp*dxp;
PVp = Axp*Lp*Phi_ref;
dxc = Lc/Nx;
Axc =  pi*(wc)^2/4;
Vbc =  Axc*dxc;
PVpc= Axc*Lc*Phi_ref;
%% correlations for rock and water properties
rho      = @(P,T) rhow_ref.*((1+ gamma_0*(P-P_ref)-beta_0*(T-T_ref)));
Bw       = @(P,T) rhow_ref./rho(P,T);
mui      = @(P,T) miu_ref*((1+d_0*(P-P_ref)+c_0*(T-T_ref)));
kwat     = @(P,T) kw*(1+e_0*(T-T_ref)+f_0*(P-P_ref));
Cpwat    = @(P,T) Cpw*(1+a_0*(T-T_ref)+b_0*(P-P_ref));
%% Generate Initial porosity and permeability
X        = 0.5*dxp:dxp:Lp-0.5*dxp;
%% Generates Initial Pressure and Temperature
```

```
P_pi      = Poutp;
T_pi      = T_ini ;
%% Iteration Options
omega    = 0.85;
err_tol  = 1e-6;
%% Rock and fluid Initial Properties
Tavge     = 0.5*(T_ini+Tinlet3);
Phii      = Civan_PORO(P_pi,T_pi);
Kxi       = Civan_PERM(T_pi);
rhosc     = rho(P_ref,T_ref);
Cpwii     = Cpwat(P_pi,T_pi);
rhofi     = rho(P_pi,T_pi);
muii      = XSteam('my_pT',P_pi*1.01325,T_pi-273.15)*1e3;
%% Model Initialization
Ue3          = zeros(M,Nt+1);
Uw3          = zeros(M,Nt+1);
Upp3         = zeros(M,Nt+1);
qinj         = zeros(Nt+1,1);
PoroT3       = zeros(M,Nt+1);
Perm3        = zeros(M,Nt+1);
Muuw3        = zeros(M,Nt+1);
Rhoow3       = zeros(M,Nt+1);
Ue3(:,1)     = 0;
Uw3(:,1)     = 0;
Upp3(:,1)    = 0;
```

```
Muuw3(:,1)   = muii;
Rhoow3(:,1)  = rhofi;
P3           = Poutp*ones(M,1);
T3           = T_pi*ones(M,1);
PoroT3(:,1)  = Phii;
Perm3(:,1)   = Kxi*1e3;
Bwater       = Bw(P_pi,T_pi);
Txx          =(( betacp*(Kxi./muii)*Axp)./(Bwater*dxp));
t            = 0;
Pinjj3(1)    = P3(1);
tic
for n     = 1:Nt  % time loop
    np1  = n+1;
    err  = 2*err_tol;
    iter = 0;
    timer= n*dt ;
    Pn   = P3(:,n);
    Piter= Pn;                Pnew_iter = Pn;
    Tn   = T3(:,n);           Tnew_iter = Tn;
    Titer= Tn;
   while (iter < 2) || (err>err_tol)    %(err>err_tol) && (iter<max_iter);
            Poron   = Civan_PORO(Pn,Tn);
            Poronp1 = Civan_PORO(Piter,Tn);
            KKx     = Civan_PERM(Tn);
            Bonp1   = Bw(Piter,Tn);
```

```
Bon     = Bw(Pn,Tn);
for i = 1:M
   wmui(i) = XSteam('my_pT',Piter(i)*1.01325,Tn(i)-273.15)*1e3;
end
miunp1  = wmui(:);
AccP = (Vbp/(alphacp*dt))*((Poronp1./Bonp1)-((Poron./Bon)));
Tx   =  ( betacp*KKx*Axp)./(Bonp1.*miunp1*dxp);
for m = 1:M
    mp1   = m+1;  mm1   = m-1;
    Txmmh = 0;    Txmph = 0;
    if m==1
            Txmph    = Tx(m);
            % coeffcients of discretized equation
            ae      = Txmph;
            ap      = (Txmph+Txmmh) ;
            ap0     = 0;
            bb      =  (qp3/Bonp1(m))-AccP(m);
           %Pressure SOR
            Pnew_iter(m) = (ae*Piter(mp1)+(ap0*Pn(m))+ bb)/ap;
            Pnew_iter(m) = (1-omega)*Piter(m)+omega*Pnew_iter(m);
 elseif m > 1 && m < M
            Txmph    = Tx(m);
            Txmmh    = Tx(mm1);
            % discretized coeffcients
            aw      = Txmmh;
```

```
                    ae      = Txmph;
                    ap      = (Txmph+Txmmh);
                    ap0     = 0 ;
                    bb      =  -AccP(m);
                    Pnew_iter(m)  = (aw*Pnew_iter(mm1)+ae*Piter(mp1)+(ap0*Pn(m))+
bb)/ap;
                    Pnew_iter(m)  = (1-omega)*Piter(m)+omega*Pnew_iter(m);
          else
                    Txmph    = 2*Tx(m);
                    Txmmh    = Tx(mm1);
                    % discretized coeffcients
                    aw      = Txmmh;
                    ap      = (Txmph+ Txmmh);
                    ap0     = 0;
                    bb      =  (Txmph*Poutp)-AccP(m);
                    Pnew_iter(m)  = (aw*Pnew_iter(mm1)+(ap0*Pn(m))+ bb)/ap;
                    Pnew_iter(m)  = (1-omega)*Piter(m)+omega*Pnew_iter(m);
                     end
           end
          Pnew   = Pnew_iter;
          Pinjp  = Pnew(1)+(qp3/(2*Tx(1)));
         % Parametrs for enrgy calculation in metric unit

         Phinp1u   = Civan_PORO(Pnew,Tn);          % Instantaneous porosity at time
level n+1
         for i = 1:M
```

```
        wmuiu(i) = XSteam('my_pT',Pnew(i)*1.01325,Tn(i)-273.15)*1e3;
      end
    visup     = wmuiu(:);
    rhowup    = rho(Pnew,Tn);
    kwu       = kwat(Pnew,Tn);
    Cpwnp1    = Cpwat(Pnew,Tn);
    Cpwn      = Cpwat(Pn,Tn);
    rhownotup = rho(Pn,Tn);
    Bonp1u    = Bw(Pnew,Tn);
     Keffec = (kwu.* Phinp1u) + (kr*(1-Phinp1u));
     Txu    =  ( betacp*KKx*Axp)./(visup.*Bonp1u*dxp);
     rhowcpbn   = ((rhownotup.*Cpwn.*Poron)+(rho_Rock*Cpr*(1-Poron)));
     rhowcpbnp1 = ((rhowup.*Cpwnp1.*Phinp1u)+(rho_Rock*Cpr*(1-Phinp1u)));
     AccTn      = (Vbc*rhowcpbn)/dt;
     AccTnp1    = (Vbc*rhowcpbnp1)/dt;
for m = 1:M
          mp1  = m+1;        mm1  = m-1;
          Scc  = 0;          Spp  = 0;
          De   = 0;          Dw   = 0;
          bbT  = 0;
       if m == 1
          % Obtain flow velocity in m/s
          ue(m)    = -0.01*(Txu(m).*(Pnew(mp1)-Pnew(m)))/(Axp);
          %uw(m)    =  (qc5/Axc);
          uw(m)    = -0.01*(Txu(m).*(Pnew(m)-Pinjp))/(0.5*Axp);
```

```
up(m)    = 0.5*(ue(m)+uw(m));

% Diffusion term

lamdaef  = harmmean([Keffec(m),Keffec(mp1)]);

Disperse = aDL*abs(ue(m))*rhowup(m)*Cpwnp1(m);
De       = ((lamdaef+Disperse)*Axc)/dxc;

% Convection term

Fe       = Axc*ue(m)*rhowup(m)*Cpwnp1(m);

% Implement Boundary condition

rhowCpin  = rho(Pnew(1),Tinlet3)*Cpwat(Pnew(1),Tinlet3);

sourceT   = rhowCpin*Axc*uw(m)*(Tinlet3);

% Heat loss terms

Scc   = (2*pi*(Hc/2)*dxc*Uoverall*(Tinfinity))/Vbc;

Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;

% Coeffiecient

aE  = De;

aP  = ( AccTnp1(m)+Fe+De+Dw-(Spp*Vbc));

aP0 =  AccTn(m);

bbT = (Scc*Vbc) +(sourceT) ;

%  Temperature SOR

 Tnew_iter(m)  = (aE*Titer(mp1)+(aP0*Tn(m))+bbT)/aP;

 Tnew_iter(m)  = (1-omega)*Titer(m)+omega*Tnew_iter(m);

 elseif m > 1 && m < M

    % Obtain flow velocity in m/s

    ue(m)    = -0.01*(Txu(m).*(Pnew(mp1)-Pnew(m)))/(Axp);

    uw(m)    = -0.01*(Txu(mm1).*(Pnew(m)-Pnew(mm1)))/(Axp);
```

```
                up(m)    = 0.5*(ue(m)+uw(m));
               % Diffusion term
                lamdaef  = harmmean([Keffec(m),Keffec(mp1)]);
                lamdawf  = harmmean([Keffec(m),Keffec(mm1)]);
                Disperse =  aDL*abs(ue(m))*rhowup(m)*Cpwnp1(m);
                Dispersw =  aDL*abs(uw(m))*rhowup(m)*Cpwnp1(mm1);
                De       =  ((lamdaef+Disperse)*Axc)/dxc;
                Dw       = ((lamdawf+Dispersw)*Axc)/dxc;
               % Convection term
                Fe       =  Axc*ue(m)*rhowup(m)*Cpwnp1(m);
                Fw       =  Axc*uw(m)*rhowup(m)*Cpwnp1(mm1);
                 % Heat loss terms
                Scc   = (2*pi*(Hc/2)*dxc*Uoverall*(Tinfinity))/Vbc;
                Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;
               % Coeffiecients
                aW = Dw+Fw;
                aE = De;
                aP = (AccTnp1(m)+Fe+De+Dw-(Spp*Vbc));
                aP0= AccTn(m);
                bbT = Scc*Vbc;
               %%
                Tnew_iter(m)  =
(aW*Tnew_iter(mm1)+aE*Titer(mp1)+(aP0*Tn(m))+bbT)/aP;
                Tnew_iter(m)  = (1-omega)*Titer(m)+omega*Tnew_iter(m);
          else
```

```
                % Obtain flow velocity in m/s
                ue(m)    = -0.01*(Txu(m).*(Poutp-Pnew(m)))/(0.5*Axp);
                uw(m)    = -0.01*(Txu(mm1).*(Pnew(m)-Pnew(mm1)))/(Axp);
                up(m)    = 0.5*(ue(m)+uw(m));
                % Diffusion term
                lamdawf  =  harmmean([Keffec(m),Keffec(mm1)]);
                Dispersw =  aDL*abs(uw(m))*rhowup(mm1)*Cpwnp1(mm1);
                Dw       = ((lamdawf+Dispersw)*Axc)/dxc;
                % Convection term
                Fe       =  Axc*ue(m)*rhowup(m)*Cpwnp1(m);
% Convective flux in east face
                Fw       =  Axc*uw(m)*rhowup(mm1)*Cpwnp1(mm1);
% Convective flux west face
                % Heat loss terms
                 Scc   = (2*pi*(Hc/2)*dxc*Uoverall*(Tinfinity))/Vbc;
                 Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;
                % Coeffiecients
                aW = Dw+Fw;
                aP = (AccTnp1(m)+De+Fe+Dw-(Spp*Vbc));
                aP0=  AccTn(m);
                bbT = Scc*Vbc;
                %%  Temp
                 Tnew_iter(m)  = (aW*Tnew_iter(mm1)+(aP0*Tn(m))+bbT)/aP;
                 Tnew_iter(m)  = (1-omega)*Titer(m)+omega*Tnew_iter(m);
            end
        end
```

```
      normP  =  norm(((Pnew_iter-Piter)/Piter),inf)
      Piter  = Pnew_iter;
      normT =  norm(((Tnew_iter-Titer)/Titer),inf)
      Titer = Tnew_iter;
      disp('error is :')
      err    = max([normP,normT])
      iter = iter+1
end
T3     = [T3 Titer];
P3     = [ P3 Piter];   Pinjj3 = [Pinjj3 Pinjp]  ;
KK      = Civan_PERM(Titer);
rhownew  = rho(Piter,Titer);
Bwnew    =  Bw(Piter,Titer);
for i = 1:M
   wmuif(i) = XSteam('my_pT',Piter(i)*1.01325,Titer(i)-273.15)*1e3;
 end
muinew   = wmuif(:);
  PoroT3(:,np1) = Civan_PORO(Piter,Titer);
  Perm3(:,np1)  = KK*1e3;
  Muuw3(:,np1)  = muinew;
  Rhoow3(:,np1) = rhownew;
  Ue3(:,np1)    = ue(:);
  Uw3(:,np1)    = uw(:);
  Upp3(:,np1)   = up(:);
  qinj(np1)    = qp3*(timer);
```

```
    disp('Volume of water injected, (cc)')
    Q           = qp3*(timer)
    disp('Simulation time in sec')
    t           = (t+dt)
end% time loop ends
toc
  tplot    = (0:dt:Tt)/60;
```

# APPENDIX G: MATLAB PROGRAMMING CODE FOR MEMORY BASED MODEL UNDER NOLTE

```
%function  [ P, T] = MB_ModelNOLTE(gamma,Pi,Pout,qinj,Tini,Tinlet,Phii,Kxi)
% Written by : Obembe Abiola David
% Date       : 13-03-2016
% Scheme     : IMPIT with constant q Pressure BC
%=========================================================================
% Solves the memory-based non-siothermal fluid flow problem
clear all
close all
clc
format long
global  K_ref Phi_ref P_ref P_out  rho_Rock T_ref  c_R  a b ...
       c m_factor alpha  hcc asf Cpr
data       = load('CoreData.txt');
L          = data(1);                  % core length (inches)
w          = data(2);                 % core width(inches)
H          = data(3);                   % core thickness(inches)
Nx         = data(4);                   % Nx grid number
Ny         = data(5);                   % Ny grid number
Nz         = data(6);                   % Nz grid number
P_ref      = data(7)*0.068046;     % Reference pressuer in atm
```

```
P_out      = data(9);                 % Out let pressure in psia
Kx         = data(10);                % Initial Measuerd Core Permeability at ref cond in mD
Phi_ref    = data(12);                % Iniital mean porosity at ref pressure and temperature
q          = data(14);                % flowrate in ml/min
T_ref      = data(15)+273.15;     % Reference Temperature K
T_ini      = data(16)+273.15;       % Initial Temperature K
Tinlet1n   = data(17)+273.15;     % Hot water temperature K
Tt         = data(18)*60;             % Simulation time in secs
dt         = data(19);                % time step in secs
c_R        = data(26);                % Rock isothermal compressibility coefficient in 1/atm
Cpr        = data(28);                % Specific heat Capacity of rock (J/Kg/K)
kr         = data(31);                % Reference thermal Conductivity of rock(W/m/K)
rho_Rock   = data(36)*1000;       % Reference density if solid matrix in kg/m3 (SiO2)
aDL        = data(37);                % Coefficient of Long. Thermal dispersion in m
betacc     = data(48);                % betac for metric unit
betacp     = data(49);                % betac for lab unit
alphacc    = data(52);                % alphacc
alphacp    = data(53);                % alphacp
alpha      = data(54);                % Fractional order derivative
m_factor   = data(56);                % Cementation factor
a          = data(57);                % Empirical constant a for Modified Kozeny-Carman
b          = data(58);                % Empirical constant b for Modified Kozeny-Carman
c          = data(59);                % Emprical constant c for Modified Kozeny-Carman
Tinfinity  = data(61)+273.15;         % Ambient temperature in degree C
Uoverall   = data(62);
```

```
dp        = data(65);
%% lab units
Lp              = 2.54*L;              % core length (centimeters)
wp              = 2.54*w;              % core width(centimeters)
Hp              = 2.54*H;              % core thickness (centimeters)
qp1n             = 0.0166667*q;         % flow rate (cc/s)
K_ref           = Kx *1e-3;            % Initial measured Core Permeability in Darcy
Poutp           = P_out*0.068046;      % Backpressure/Outlet pressure in atm
% spe metric units
Lc              = 0.0254*L;            % core length (meters)
wc              = 0.0254*w;              % core width(meters)
Hc              = 0.0254*H;                % core thickness (meters)
qc1n             = 1.66666667e-8*q;     % flow rate (cubic meters/s)
%% Grid size
NyNz       = Ny*Nz;
M          = Nx*Ny*Nz;
Mp1        = M+1;
Nt         = Tt/dt;
%% More variables
dxp        = Lp/Nx;
Axp        = pi*(wp)^2/4;
Vbp        =  Axp*dxp;
PVp        = Axp*Lp*Phi_ref;
dxc        = Lc/Nx;
Axc        =  pi*(wc)^2/4;
```

```
Vbc        =  Axc*dxc;
PVt        = Axc*Lc*Phi_ref;
%% Generate Initial porosity and permeability
X          = 0.5*dxp:dxp:Lp-0.5*dxp;
%% Generates Initial Pressure and Temperature
MP2= M+2;
P_pi       = Poutp*ones(M,1);                    % Initial pressure in atm (lab units)
T_pi       =  T_ini*ones(M,1);                   % Initial temperature in each block K
Pcomp      = [P_pi; Poutp;Poutp];
%% Iteration Options
omega      = 0.5;
err_tol    = 1e-6;
%% Pre computaion of wk and coeffiecient
D      = 1/(dt^(1-alpha));                       % Coefficient of memory term
gtabl = ones(1,Nt);
for r          = 2:Nt+1
    nn         = r -1;
    gtabl(1,r)= -gtabl(1,r-1)*(2-alpha-nn)/nn;
end
Shmemory = 65*60;                                % arbitrary chosen by user
%% Rock and fluid Initial Properties
Phii       = Civan_PORO(P_pi,T_pi);
Kxi        = Civan_PERM(T_pi);
rhosc      = XSteam('rho_pT',P_ref*1.01325, T_ref-273.15);
muii       = XSteam('my_pT',P_pi(1)*1.01325,T_pi(1)-273.15)*1000;
```

```
rhofi     = XSteam('rho_pT',(P_pi(1)*1.01325),(T_pi(1)-273.15));
%% Model Initialization
Ue1n          = zeros(M,Nt+1);
Uw1n          = zeros(M,Nt+1);
Upp1n         = zeros(M,Nt+1);
qinj          = zeros(Nt+1,1);
PoroT1n       = zeros(M,Nt+1);
Perm1n        = zeros(M,Nt+1);
Muuw1n        = zeros(M,Nt+1);
Rhoow1n       = zeros(M,Nt+1);
Kimp1n        = zeros(M,Nt+1);
Eta1n         = zeros(M,Nt+1);
Pinjj1n       = zeros(Nt+1,1);
pvinj1n       = zeros(Nt+1,1);
Ue1n(:,1)     = 0;
Uw1n(:,1)     = 0;
Upp1n(:,1)    = 0;
Kimp1n(:,1)   = 1;
pvinj1n(1)    = 0;
Muuw1n(:,1)   = muii;
Rhoow1n(:,1)  = rhofi;
P1n           = P_pi;
Tfluid1n      = T_pi;
Trock1n       = T_pi;
PoroT1n(:,1)  = Phii;
```

```
Perm1n(:,1)   = Kxi*1e3;
Eta1n(:,1)    = Kxi./muii ;
Bwater        = (rhosc/rhofi)*ones(M,1);
Txx           =((betacp*(Kxi./muii)*Axp*D)./(Bwater*dxp));
t             = 0;
Pinjj1n(1)    = P1n(1);
%%%% Solve Pressure Equation
tic
for n     = 1:Nt  % time loop
    np1   = n+1;
    err   = 2*err_tol;
    iter  = 0;
    timer = n*dt ;
    Pn    = P1n(:,n);
    Piter = Pn;                    Pnew_iter = Pn;
    Trn   = Trock1n(:,n);          Trnew_iter = Trn;
    Triter= Trn;
    Tfn   = Tfluid1n(:,n);         Tfnew_iter = Tfn;
    Tfiter= Tfn;
    while (iter < 2) || (err>err_tol)    %(err>err_tol) && (iter<max_iter);
            Tavgn   = 0.5*(Trn+Tfn);
            Tavgnp1 =  0.5*(Triter+Tfiter);
            Poron   = Civan_PORO(Pn,Tavgn);
            Poronp1 = Civan_PORO(Piter,Tavgn);
            KKx     = Civan_PERM(Tavgn);
```

```
for i = 1:M
   miu(i)   = XSteam('my_pT',Pn(i)*1.01325,Tfn(i)-273.15)*1e3;
   rhon(i)  = XSteam('rho_pT',Pn(i)*1.01325,Tfn(i)-273.15);
   rhonp1(i)= XSteam('rho_pT',Piter(i)*1.01325,Tfn(i)-273.15);
   Bon(i)   = rhosc/rhon(i);
   Bonp1(i) = rhosc/rhonp1(i);
end
Bwnp1   = Bonp1(:);
Bwn     = Bon(:);
viswnp1 = miu(:);
AccP    = (Vbp/(alphacp*dt))*((Poronp1./Bwnp1)-((Poron./Bwn)));
eeta    = (KKx./viswnp1)*(timer).^(1-alpha);
Tx      =  ( betacp*eeta*Axp*D)./(Bwnp1*dxp);
for m = 1:M
    mp1   = m+1;  mm1   = m-1;
    Txmmh = 0;    Txmph = 0;
            if m == 1
               Sum(m)   = 0;
               for k     = 1:1:min(Shmemory/dt,n);          % past history effect
                  wk    = gtabl(1,k+1);                    % wk term
                  Sum(m) = Sum(m) + (wk*(Txx(m,n-k+1).*(P1n(mp1,n-k+1)-
P1n(m,n-k+1))));
               end
            % for mph
             Kxmph    = KKx(m);
```

```
                miumph   = viswnp1(m);
                Bwmph    = Bwnp1(m);
                eetaxmph = (Kxmph/miumph)*(timer).^(1-alpha);
                Txmph    = ( betacp*eetaxmph*Axp*D)./(Bwmph*dxp);
                % coeffcients of discretized equation
                ae      = Txmph;
                ap      = (Txmph+Txmmh) ;
                ap0     = 0;
                bb      =  (qp1n/Bwmph)+Sum(m)-AccP(m);
               %Pressure SOR
                Pnew_iter(m) = (ae*Piter(mp1)+(ap0*Pn(m))+ bb)/ap;
                Pnew_iter(m) = (1-omega)*Piter(m)+omega*Pnew_iter(m);
        elseif m > 1 && m < M
                Sum(m) = 0;
                for k  = 1:1:min(Shmemory/dt,n);          % past history effect
                   wk = gtabl(1,k+1);                  % wk term
                   Sum(m) = Sum(m) + (wk*((Txx(mm1,n-k+1).*(P1n(mm1,n-k+1)-
P1n(m,n-k+1)))+ ...
                        (Txx(m,n-k+1).*(P1n(mp1,n-k+1)-P1n(m,n-k+1)))));
                end
                % for mph
                Kxmph    = KKx(m);
                miumph   = viswnp1(m);
                Bwmph    = Bwnp1(m);
                eetaxmph = (Kxmph/miumph)*(timer).^(1-alpha);
```

```
                    Txmph    = ( betacp*eetaxmph*Axp*D)./(Bwmph*dxp);
                    % for mmh
                    Kxmmh    = KKx(mm1);
                    miummh   = viswnp1(mm1);
                    Bwmmh    = Bwnp1(mm1);
                    eetaxmmh = (Kxmmh/miummh)*(timer).^(1-alpha);
                    Txmmh    = ( betacp*eetaxmmh*Axp*D)./(Bwmmh*dxp);
                    % discretized coeffcients
                    aw      = Txmmh;
                    ae      = Txmph;
                    ap      = (Txmph + Txmmh);
                    ap0     = 0 ;
                    bb      =  Sum(m)-AccP(m);
                    Pnew_iter(m)  = (aw*Pnew_iter(mm1)+ae*Piter(mp1)+(ap0*Pn(m))+
bb)/ap;
                    Pnew_iter(m)  = (1-omega)*Piter(m)+omega*Pnew_iter(m);
          else
                   Sum(m) = 0;
                    for k  = 1:1:min(Shmemory/dt,n);          % past history effect
                       wk = gtabl(1,k+1);                  % wk term
                       Sum(m) = Sum(m) + (wk*((Txx(mm1,n-k+1).*(P1n(mm1,n-k+1)-
P1n(m,n-k+1)))+ ...
                            (2*Txx(m,n-k+1).*(Poutp-P1n(m,n-k+1)))));
                    end
                     % for mph
                    Kxmph    = KKx(m);
```

```
            miumph   = miu(m);
            Bwmph    = Bonp1(m);
            eetaxmph = (Kxmph/miumph)*(timer).^(1-alpha);
            Txmph    = ( betacp*eetaxmph*Axp*D)./(Bwmph*dxp*0.5);
            % for mmh
            Kxmmh    = KKx(mm1);
            miummh   = miu(mm1);
            Bwmmh    = Bonp1(mm1);
            eetaxmmh = (Kxmmh/miummh)*(timer).^(1-alpha);
            Txmmh    = ( betacp*eetaxmmh*Axp*D)./(Bwmmh*dxp);
            % discretized coeffcients
            aw      = Txmmh;
            ap      = (Txmph + Txmmh);
            ap0     = 0;
            bb      =  (Txmph*Poutp)+Sum(m)-AccP(m);

            Pnew_iter(m)  = (aw*Pnew_iter(mm1)+(ap0*Pn(m))+ bb)/ap;
            Pnew_iter(m)  = (1-omega)*Piter(m)+omega*Pnew_iter(m);
             end
 end
Pnew   = Pnew_iter;
 Sums   = 0;
for k  = 1:1:min(Shmemory/dt,n);          % past history effect
      wk = gtabl(1,k+1);                  % wk term
      Sums = Sums + wk*(2*Txx(1,n-k+1)*(Pinjj1n(1,n-k+1)-P1n(1,n-k+1)));
```

```
    end
 Pinjp  = Pnew(1)+((qp1n-Sums)/(2*Tx(1)));
% Parametrs for enrgy calculation in metric unit
Phinp1u    = Civan_PORO(Pnew,Tavgn);          % Instantaneous porosity at time level n+1
for r = 1: M
   visupp(r)      = XSteam('my_pT',Pnew(r)*1.01325,Tfn(r)-273.15)*1e3;
   rhowupp(r)     = XSteam('rho_pT',Pnew(r)*1.01325,Tfn(r)-273.15);
   kwuu(r)        = XSteam('tc_pT',Pnew(r)*1.01325,Tfn(r)-273.15);
   Cpwnp1u(r)     = XSteam('Cp_pT',Pnew(r)*1.01325,Tfn(r)-273.15)*1e3;
   Cpwnu(r)       = XSteam('Cp_pT',Pnew(r)*1.01325,Tfn(r)-273.15)*1e3;
   rhownotupp(r)  = XSteam('rho_pT',Pnew(r)*1.01325,Tfn(r)-273.15);
   Bwnp1up(r)     = rhosc/rhowupp(r);
end
 visup     = visupp(:);
rhowup    = rhowupp(:);
kwu       = kwuu(:);
Cpwnp1    = Cpwnp1u(:);
Cpwn      = Cpwnu(:);
rhownotup = rhownotupp(:);
Bwwu      = Bwnp1up(:);
Kreff     = kr*(1-Phinp1u);
Kweff     = kwu .* Phinp1u ;
Txu       =  ( betacp*KKx*Axp)./(visup.*Bwwu*dxp);
AccTrn    = (rho_Rock*Cpr.*(1-Poron))*(Vbc/dt);
```

```
AccTrnp1  = (rho_Rock*Cpr.*(1-Phinp1u))*(Vbc/dt);

AccTfn    = (rhownotup.*Cpwn.*Poron)*(Vbc/dt);

AccTfnp1  =(rhowup.*Cpwnp1.*Phinp1u)*(Vbc/dt);

Prr       =   (visup.*Cpwnp1*1e-3)./kwu;

asf       = 6*(1-Phinp1u)/dp;

Pnewcomp   = [ Pnew; Poutp; Pinjp];

Pmemory    = [ Pcomp Pnewcomp];

Pdot       = fractional_der_press(Pmemory,alpha);

for m = 1:M

      mp1     = m+1;        mm1  = m-1;

      De      = 0;          Dw   = 0;

      Dre     = 0;          Drw  = 0;

      bbR     = 0;          bbF  = 0;

      if m == 1

     ue(m)    = -0.01*(Txu(m).*(Pdot(mp1)-Pdot(m)))/(Axp);

  uw(m)    = -0.01*(Txu(m).*(Pdot(m)-Pdot(MP2)))/(0.5*Axp);
 up(m)    = 0.5*(ue(m)+uw(m));

      hcc(m) = (kwu(m)*(2+
1.1*(Prr(m)^(1/3))*((rhowup(m)*up(m)*dp)/(visup(m)*1e-3))^0.6))/dp;

      %  for rock temperature

      kreffe =  harmmean([ Kreff(m),Kreff(mp1)]);

      Dre    =  kreffe*(Axc/dxc);

     % Coeffiecient

      aEr    =  Dre;

      aPr    =  (AccTrnp1(m)+Dre+Drw+(hcc(m)*asf(m)*Vbc));

      aP0r   = AccTrn(m);
```

```
bbTs   =  (hcc(m)*asf(m))*Vbc*Tfiter(m);
% For  Fluid Temperature
rhowin   = XSteam('rho_pT',Pnew(1)*1.01325,Tinlet1n-273.15);
Fe       = ue(m)*Axc*rhowup(m)*Cpwnp1(m);
kfeffe   = harmmean([Kweff(m),Kweff(mp1)]);
De       = kfeffe*(Axc/dxc);
% Heat loss terms
Scc   = (2*pi*(Hc/2)*dxc*Uoverall*Tinfinity)/Vbc;
Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;
% Boundary condition
sourceF     = XSteam('Cp_pT',Pnew(1)*1.01325,Tinlet1n-
273.15)*rhowin*uw(m)*Axc*Tinlet1n*1e3;
% Coeffcients
aEf      = De;
aPf      = (AccTfnp1(m)+(De+Fe)+Dw+(hcc(m)*asf(m)*Vbc));
aP0f     = AccTfn(m);
bbTf      =  sourceF +(hcc(m)*asf(m)*Vbc*Triter(m));
% Fluid Temperature SOR
Tfnew_iter(m)  = (aEf*Tfiter(mp1)+(aP0f*Tfn(m))+bbTf)/aPf;
Tfnew_iter(m)  = (1-omega)*Tfiter(m)+omega*Tfnew_iter(m);
%  Rock Temperature SOR
Trnew_iter(m)  = (aEr*Triter(mp1)+(aP0r*Trn(m))+bbTs)/aPr;
Trnew_iter(m)  = (1-omega)*Triter(m)+omega*Trnew_iter(m);
elseif m > 1 && m < M
        % Obtain flow velocity in m/s
```

```
                ue(m)    = -0.01*(Txu(m).*(Pdot(mp1)-Pdot(m)))/(Axp);
uw(m)    = -0.01*(Txu(mm1).*(Pdot(m)-Pdot(mm1)))/(Axp);

                up(m)    = 0.5*(ue(m)+uw(m));

                hcc(m) = (kwu(m)*(2+
1.1*(Prr(m)^(1/3))*((rhowup(m)*up(m)*dp)/(visup(m)*1e-3))^0.6))/dp;

                %  for rock temperature

                 kreffe =  harmmean([ Kreff(m),Kreff(mp1)]);

                 kreffw =  harmmean([ Kreff(m),Kreff(mm1)]);

                 Dre    =  kreffe*(Axc/dxc);

                 Drw    =  kreffw*(Axc/(dxc));

              % Coeffiecient

                aEr    =  Dre;

                aWr    =  Drw;

                aPr    =  (AccTrnp1(m)+Dre+Drw+(hcc(m)*asf(m)*Vbc));

                aP0r   = AccTrn(m);

                bbTs    =   (hcc(m)*asf(m))*Vbc*Tfiter(m);

              % For Fluid Temperature

                Fe     = ue(m)*Axc*rhowup(m)*Cpwnp1(m);

                Fw     = uw(m)*Axc*rhowup(mm1)*Cpwnp1(mm1);

                kfeffe = mean([Kweff(m),Kweff(mp1)]);

                kfeffw = mean([Kweff(m),Kweff(mm1)]);

                Dw     = kfeffw*(Axc/dxc);

                De     = kfeffe*(Axc/dxc);

              % Heat loss terms

                Scc   = (2*pi*(Hc/2)*dxc*Uoverall*Tinfinity)/Vbc;

                Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc ;
```

```
                % Coeffcients
                   aWf      = Dw+Fw;
                   aEf      = De;
                   aPf      = (AccTfnp1(m)+(De+Fe)+Dw+(hcc(m)*asf(m)*Vbc));
                   aP0f     = AccTfn(m);
                   bbTf     =    (hcc(m)*asf(m))*Vbc*Triter(m);
                %  Fluid Temperature SOR
                   Tfnew_iter(m)  =
(aWf*Tfnew_iter(mm1)+aEf*Tfiter(mp1)+(aP0f*Tfn(m))+bbTf)/aPf;
                   Tfnew_iter(m)  = (1-omega)*Tfiter(m)+omega*Tfnew_iter(m);
                %  Rock Temperature SOR
                   Trnew_iter(m)  =
(aWr*Trnew_iter(mm1)+aEr*Triter(mp1)+(aP0r*Trn(m))+bbTs)/aPr;
                   Trnew_iter(m)  = (1-omega)*Triter(m)+omega*Trnew_iter(m);
             else
                % Obtain flow velocity in m/s
                   ue(m)    = -0.01*(Txu(m).*(Pdot(mp1)-Pdot(m)))/(0.5*Axp);
uw(m)    = -0.01*(Txu(mm1).*(Pdot(m)-Pdot(mm1)))/(Axp);
                   up(m)    = 0.5*(ue(m)+uw(m));
                   hcc(m) = (kwu(m)*(2+
1.1*(Prr(m)^(1/3))*((rhowup(m)*up(m)*dp)/(visup(m)*1e-3))^0.6))/dp;
                %  for rock temperature
                   kreffw =  harmmean([ Kreff(m),Kreff(mm1)]);
                   Drw    =  kreffw*(Axc/(dxc));
                % Coeffiecient
                    aWr    =  Drw;
                    aPr    =  (AccTrnp1(m)+Dre+Drw+(hcc(m)*asf(m)*Vbc));
```

```
            aP0r   = AccTrn(m);
            bbR    =  hcc(m)*asf(m)*Vbc*Tfiter(m);
          % For  Fluid Temperature
            Fe     = ue(m)*Axc*rhowup(m)*Cpwnp1(m);
            Fw     = uw(m)*Axc*rhowup(mm1)*Cpwnp1(mm1);
            kfeffw   =  harmmean([Kweff(m),Kweff(mm1)]);
            Dw     = kfeffw*(Axc/dxc);
          % Heat loss terms
           Scc   = (2*pi*(Hc/2)*dxc*Uoverall*Tinfinity)/Vbc;
           Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;
          % Coeffcients
            aWf    = Dw+Fw;
            aPf    = (AccTfnp1(m)+(De+Fe)+Dw+(hcc(m)*asf(m)*Vbc));
            aP0f   = AccTfn(m);
            bbF    =  hcc(m)*asf(m)*Vbc*Triter(m);
          % Fluid Temp
           Tfnew_iter(m)  = (aWf*Tfnew_iter(mm1)+(aP0f*Tfn(m))+ bbF)/aPf;
           Tfnew_iter(m)  = (1-omega)*Tfiter(m)+omega*Tfnew_iter(m);
          % Rock Temp
           Trnew_iter(m)  = (aWr*Trnew_iter(mm1)+(aP0r*Trn(m))+bbR)/aPr;
           Trnew_iter(m)  = (1-omega)*Triter(m)+omega*Trnew_iter(m);
      end
end
 normP  =  sum(abs((Pnew_iter-Piter)./Piter));
 Piter  = Pnew_iter;
```

```
        normTf =  sum(abs((Tfnew_iter-Tfiter)./Tfiter));
        Tfiter = Tfnew_iter;
        normTr =  sum(abs((Trnew_iter-Triter)./Triter));
        Triter = Trnew_iter;
        disp('error is :')
        err    = max([normP,normTf,normTr])
        iter   = iter+1
 end
 Tavgup   = 0.5*(Triter+Tfiter);
 Tfluid1n  = [Tfluid1n Tfiter];
 P1n       = [ P1n Piter];
 Trock1n   = [Trock1n Triter];
 %Pinjj1n(np1)   =  Pinjp;
 KKu      = Civan_PERM(Tavgup);
for i= 1:M
    rhownew(i) = XSteam('rho_pT',Piter(i)*1.01325,Tfiter(i)-273.15);
    Bwnew(i)   = rhosc./rhownew(i);
    muinew(i)  = XSteam('my_pT',Piter(i)*1.01325,Tfiter(i)-273.15)*1e3;
    eetanew(i) = (KKu(i)/muinew(i))*(timer).^(1-alpha);
    Tx(i)      = ( betacp*eetanew(i)*Axp*D)./(Bwnew(i)*dxp);
end
 Bwater   = [ Bwater Bwnew'];
 Txx     = [ Txx  Tx];
 Eta1n(:,np1)= eetanew';
PoroT1n(:,np1) = Civan_PORO(Piter,Tavgup);
```

```
   Perm1n(:,np1)  = KKu;
   Muuw1n(:,np1)  = muinew';
   Rhoow1n(:,np1) = rhownew';
   Ue1n(:,np1)    = ue(:);
   Uw1n(:,np1)    = uw(:);
   Upp1n(:,np1)   = up(:);
   Kimp1n(:,np1)  = KKu./Kxi;
   qinj(np1)      = qp1n*t;
   pvinj1n(np1)   = (qc1n*timer)/PVt;
   disp('Volume of water injected, (m^3)')
   Q           = qc1n*t
   t           = t+dt;
end% time loop ends
 toc
   tplot    = (0:dt:Tt)/60;
```

# APPENDIX H: MATLAB PROGRAMMING CODE FOR MODIFIED OB MODEL UNDER NOLTE

```
%function  [ P, T] = OB_ModelNOLTE(Pi,Pout,qinj,Tini,Tinlet,Phii,Kxi)
% Written by : Obembe Abiola David
% Date       : 13-03-2016
% Scheme     : IMPIT with constant flowrate BC
%=========================================================================
% Solves the memory-based non-isothermal fluid flow problem
clear all
close all
clc
format long
global  K_ref Phi_ref P_ref P_out rhow_ref rho_Rock T_ref c_R  a b ...
       c m_factor gamma_0 beta_0 a_0 b_0 c_0 d_0 e_0 f_0 kw Cpw

data       = load('CoreData.txt');
L          = data(1);                % core length (inches)
w          = data(2);                % core width(inches)
H          = data(3);                % core thickness(inches)
Nx         = data(4);                % Nx grid number
Ny         = data(5);                % Ny grid number
Nz         = data(6);                % Nz grid number
P_ref      = data(7)*0.068046;     % Reference pressure in atm
```

```
P_out      = data(9)*0.068046;     % Out let pressure in atm
Kx         = data(10);             % Initial measured Core Permeability at ref cond in mD
Phi_ref    = data(12);             % Initial mean porosity at ref pressure and temperature
q          = data(14);             % flowrate in ml/min
T_ref      = data(15)+273.15;      % Reference Temperature K
T_ini      = data(16)+273.15;      % Initial Temperature K
Tinlet3    = data(17)+273.15;      % Hot water temperature K
Tt         = data(18)*60;          % Simulation time in secs
dt         = data(19);             % time step in secs
rhow_ref   = data(20)*1e3;         % Reference density if water in g/cm3
gamma_0    = data(23)*1013250;     % Water density isothermal compressibility 1/atm
beta_0     = data(24);             % Water density thermal expansion coefficient 1/K
c_R        = data(26);             % Rock isothermal compressibility coefficient in 1/atm
Cpr        = data(28);             % Specific heat Capacity of rock (J/Kg/K)
Cpw        = data(29);             % Reference Specific heat Capacity of water (J/Kg/K)
kw         = data(30);             % Reference thermal Conductivity of water (W/m/K)
kr         = data(31);             % Reference thermal Conductivity of rock(W/m/K)
f_0        = data(32)*1013250;     % Water Thermal conductivity isothermal expansion
e_0        = data(33);             % Water thermal conductivity thermal expansion
b_0        = data(34)*1013250;     % Water  Specific heat  isothermal expansion
a_0        = data(35);             % Water Specific heat thermal expansion coefficient 1/K
rho_Rock   = data(36)*1000;        % Reference density if solid matrix in kg/m3
aDL        = data(37);             % Coefficient of Long. Thermal dispersion in m
betacc     = data(48);             % betac for metric unit
betacp     = data(49);             % betac for lab unit
```

```
c_0        = data(50);                % Thermal part of viscosity in 1/K
d_0        = data(51)*1013250;     % Isothermal part of water viscosity in atm
alphacp    = data(53);                % alphacp
m_factor   = data(56);                % Cementation factor
a          = data(57);                % Empirical constant a for Modified Kozeny-Carman
b          = data(58);                % Empirical constant b for Modified Kozeny-Carman
c          = data(59);                % Empirical constant c for Modified Kozeny-Carman
Tinfinity  = data(61)+273.15;    % Ambient temperature in degree C
Uoverall   = data(62);
dp         = data(65);
%% lab units
Lp    = 2.54*L;              % core length (centimeters)
wp    = 2.54*w;              % core width(centimeters)
Hp    = 2.54*H;              % core thickness(centimeters)
qp3    = 0.0166667*q;         % flow rate (cc/s)
K_ref = Kx *1e-3;            % Initial Measuerd Core Permeability in Darcy
Poutp = P_out;               % Backpressure/Outlet pressure in atm
% spe metric units
Lc    = 0.0254*L;            % core length (meters)
wc    = 0.0254*w;            % core width(meters)
Hc    = 0.0254*H;            % core thickness(meters)
qc3    = 1.66666667e-8*q;      % flow rate (cubic meters/s)
%% Grid size
NyNz  = Ny*Nz;
M     = Nx*Ny*Nz;
```

```
Mp1   = M+1;

Nt    = Tt/dt;

%% More variables

dxp   = Lp/Nx;

Axp   = pi*(wp)^2/4;

Vbp   =  Axp*dxp;

PVp   = Axp*Lp*Phi_ref;

dxc   = Lc/Nx;

Axc   =  pi*(wc)^2/4;

Vbc   =  Axc*dxc;

PVt3   = Axc*Lc*Phi_ref;

%% correlations for rock and water properties

rho       = @(P,T) rhow_ref.*((1+ gamma_0*(P-P_ref)-beta_0*(T-T_ref));

Bw        = @(P,T) rhow_ref./rho(P,T);

kwat      = @(P,T) kw*(1+e_0*(T-T_ref)+f_0*(P-P_ref));

Cpwat     = @(P,T) Cpw*(1+a_0*(T-T_ref)+b_0*(P-P_ref));

%% Generate Initial porosity and permeability

X         = 0.5*dxp:dxp:Lp-0.5*dxp;

%% Generates Initial Pressure and Temperature

P_pi      = Poutp*ones(M,1);

T_pi      = T_ini*ones(M,1) ;

%% Iteration Options

omega     = 0.75;

err_tol   = 1e-6;                              % arbitrary chosen by user

%% Rock and fluid Initial Properties
```

```
Phii      = Civan_PORO(P_pi,T_pi);
Kxi       = Civan_PERM(T_pi);
rhosc     = rho(P_ref, T_ref);
muii      = XSteam('my_pT',P_pi(1)*1.01325,T_pi(1)-273.15)*1000;
rhofi     = rho(P_pi,T_pi);
%% Model Initialization
Ue3          = zeros(M,Nt+1);
Uw3          = zeros(M,Nt+1);
Upp3         = zeros(M,Nt+1);
qinj         = zeros(Nt+1,1);
pvinj3       = zeros(Nt+1,1);
PoroT3       = zeros(M,Nt+1);
Perm3        = zeros(M,Nt+1);
Muuw3        = zeros(M,Nt+1);
Rhoow3       = zeros(M,Nt+1);
Kimp3        = zeros(M,Nt+1);
Pinjj3       = zeros(Nt+1,1);
Ue3(:,1)     = 0;
Uw3(:,1)     = 0;
Upp3(:,1)    = 0;
Kimp3(:,1)   = 1;
pvinj3(1)    = 0;
Muuw3(:,1)   = muii;
Rhoow3(:,1)  = rhofi;
P3           = P_pi;
```

```
Tr3         = T_pi;
Tf3         = T_pi;
PoroT3(:,1)  = Phii;
Perm3(:,1)   = Kxi*1e3;
t           = 0;
Pinjj3(1)    = P3(1);
 %%% Solve Pressure Equation
tic
for n     = 1:Nt  % time loop
    np1   = n+1;
    err   = 2*err_tol;
    iter  = 0;
    timer = n*dt ;
    Pn    = P3(:,n);
    Piter = Pn;          Pnew_iter = Pn;
    Trn   = Tr3(:,n);     Trnew_iter= Trn;
    Triter= Trn;         Tfn       = Tf3(:,n);
    Tfiter= Tfn;         Tfnew_iter = Tfn;
   while (iter <2) || (err>err_tol)    %(err>err_tol) && (iter<max_iter);
            Tavgn   = 0.5*(Trn+Tfn);
            Poron   = Civan_PORO(Pn,Tavgn);
            Poronp1 = Civan_PORO(Piter,Tavgn);
            KKx     = Civan_PERM(Tavgn);
            Bonp1   = Bw(Pn,Tfn);
            Bon     = Bw(Pn,Tfn);
```

```
for i = 1:M
   miu(i)    = XSteam('my_pT',Pn(i)*1.01325,Tfn(i)-273.15)*1e3;
end
visnp1 = miu(:);
AccP = (Vbp/(alphacp*dt))*((Poronp1./Bonp1)-((Poron./Bon;
Tx   =  ( betacp*KKx*Axp)./(visnp1.*Bon*dxp);
 for m = 1:M
    mp1   = m+1;  mm1   = m-1;
    Txmmh = 0;    Txmph = 0;
    % Constant Injection rate BC at inlet
 if m == 1
            % for mph
             Bwmph    = Bonp1(m);
             Txmph   = Tx(m);
             % coeffcients of discretized equation
             ae      = Txmph;
             ap      = (Txmph+Txmmh);
             ap0     = 0;
             bb      =  (qp3/Bwmph)-AccP(m);
            %Pressure SOR
             Pnew_iter(m) = (ae*Piter(mp1)+(ap0*Pn(m))+ bb)/ap;
             Pnew_iter(m) = (1-omega)*Piter(m)+omega*Pnew_iter(m);
 elseif m > 1 && m < M
             Txmph    = Tx(m);
             Txmmh    = Tx(mm1);
```

```
                % discretized coeffcients
                aw      = Txmmh;
                ae      = Txmph;
                ap      = (Txmph+Txmmh);
                ap0     = 0;
                bb      =  -AccP(m);
                Pnew_iter(m)  = (aw*Pnew_iter(mm1)+ae*Piter(mp1)+(ap0*Pn(m))+
bb)/ap;
                Pnew_iter(m)  = (1-omega)*Piter(m)+omega*Pnew_iter(m);
          else
                Txmph    = 2*Tx(m);
                Txmmh    = Tx(mm1);
                % discretized coeffcients
                aw      = Txmmh;
                ap      = (Txmph+Txmmh);
                ap0     = 0;
                bb      = (Txmph*Poutp)-AccP(m);
                Pnew_iter(m)  = (aw*Pnew_iter(mm1)+(ap0*Pn(m))+ bb)/ap;
                Pnew_iter(m)  = (1-omega)*Piter(m)+omega*Pnew_iter(m);
           end
           end
          Pnew   = Pnew_iter;
          Pinjp  = Pnew(1)+(qp3/(2*Tx(1)));
         % Parametrs for enrgy calculation in metric unit
         Phinp1u    = Civan_PORO(Pnew,Tavgn);
```

```
for r = 1: M
        muuw(r)      = XSteam('my_pT',Pnew(r)*1.01325,Tavgn(r)-273.15)*1e3;
    end
    visup     = muuw(:);
    rhowup    = rho(Pnew,Tfn);
    kwu       = kwat(Pnew,Tfn);
    Cpwnp1    = Cpwat(Pnew,Tfn);
    Cpwn      = Cpwat(Pn,Tfn);
    rhownotup = rho(Pn,Tfn);
    Bonp1u    = Bw(Pnew,Tfn);
    Kreff     = kr*(1-Phinp1u);
    Kweff     = kwu .* Phinp1u ;
    Txu       =  ( betacp*KKx*Axp)./(visup.*Bonp1u*dxp);
    AccTrn    = (rho_Rock*Cpr.*(1-Poron))*(Vbc/dt);
    AccTrnp1  = (rho_Rock*Cpr.*(1-Phinp1u))*(Vbc/dt);
    AccTfn    = (rhownotup.*Cpwn.*Poron)*(Vbc/dt);
    AccTfnp1  =(rhowup.*Cpwnp1.*Phinp1u)*(Vbc/dt);
    Prr       =   (visup.*Cpwnp1*1e-3)./kwu;
    asf       = 6*(1-Phinp1u)/dp;
  for m = 1:M
        mp1  = m+1;        mm1  = m-1;
        Scs  = 0;          Sps  = 0;
        Scf  = 0;          Spf  = 0;
        Dre  = 0;          Drw  = 0;
        De   = 0;          Dw   = 0;
```

```
bbTs = 0;          bbTf = 0;

Scc  = 0;          Spp  = 0;

if m == 1

% Obtain flow velocity in m/s

ue(m)    = -0.01*(Txu(m).*(Pnew(mp1)-Pnew(m)))/(Axp);

uw(m)    =  (0.01*qp3/Axp);

%uw(m)    = -0.01*(Txu(m).*(Pnew(m)-Pinjp))/(0.5*Axp);

up(m)    = 0.5*(ue(m)+uw(m));

% Heat transfer between rock and fluid

hcc(m) = (kwu(m)*(2+
1.1*(Prr(m)^(1/3))*((rhowup(m)*up(m)*dp)/(visup(m)*1e-3))^0.6))/dp;

% for rock temperature

kreffe =  harmmean([ Kreff(m),Kreff(mp1)]);

Dre    =  kreffe*(Axc/dxc);

% Coefficients

aEr    =  Dre;

aPr    =  (AccTrnp1(m)+Dre+Drw+(hcc(m)*asf(m)*Vbc));

aP0r   = AccTrn(m);

bbTs   =  (hcc(m)*asf(m))*Vbc*Tfiter(m);

% For Fluid Temperature

rhowin   = rho(Pnew(1),Tinlet3);

Fe       = ue(m)*Axc*rhowup(m)*Cpwnp1(m);

kfeffe   = harmmean([Kweff(m),Kweff(mp1)]);

De       = kfeffe*(Axc/dxc);

%% Heat loss terms
```

```
Scc   = (2*pi*(Hc/2)*dxc*Uoverall*Tinfinity)/Vbc;
Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;
%% Boundary condition
sourceF     = Cpwat(Pnew(1),Tinlet3)*rhowin*uw(m)*Axc*Tinlet3;
% Coefficients
aEf      = De;
aPf      = (AccTfnp1(m)+(De+Fe)+Dw+(hcc(m)*asf(m)*Vbc));
aP0f     = AccTfn(m);
bbTf      =  sourceF +(hcc(m)*asf(m)*Vbc*Triter(m));
% Fluid Temperature SOR
Tfnew_iter(m)  = (aEf*Tfiter(mp1)+(aP0f*Tfn(m))+bbTf)/aPf;
Tfnew_iter(m)  = (1-omega)*Tfiter(m)+omega*Tfnew_iter(m);
%  Rock Temperature SOR
Trnew_iter(m)  = (aEr*Triter(mp1)+(aP0r*Trn(m))+bbTs)/aPr;
Trnew_iter(m)  = (1-omega)*Triter(m)+omega*Trnew_iter(m);

elseif m > 1 && m < M
    % Obtain flow velocity in m/s
    ue(m)    = -0.01*(Txu(m).*(Pnew(mp1)-Pnew(m)))/(Axp);
    uw(m)    = -0.01*(Txu(mm1).*(Pnew(m)-Pnew(mm1)))/(Axp);
    up(m)    = 0.5*(ue(m)+uw(m));
   % Heat transfer between rock and fluid
    hcc(m) = (kwu(m)*(2+
1.1*(Prr(m)^(1/3))*((rhowup(m)*up(m)*dp)/(visup(m)*1e-3))^0.6))/dp;
    %  for rock temperature
```

```
    kreffe =  harmmean([ Kreff(m),Kreff(mp1)]);
    kreffw =  harmmean([ Kreff(m),Kreff(mm1)]);
    Dre    =  kreffe*(Axc/dxc);
    Drw    =  kreffw*(Axc/(dxc));
% Coeffiecient
   aEr    =  Dre;
   aWr    =  Drw;
   aPr    =  (AccTrnp1(m)+Dre+Drw+(hcc(m)*asf(m)*Vbc));
   aP0r   = AccTrn(m);
   bbTs   =   (hcc(m)*asf(m))*Vbc*Tfiter(m);
% For Fluid Temperature
   Fe     = ue(m)*Axc*rhowup(m)*Cpwnp1(m);
   Fw     = uw(m)*Axc*rhowup(mm1)*Cpwnp1(mm1);
   kfeffe = mean([Kweff(m),Kweff(mp1)]);
   kfeffw = mean([Kweff(m),Kweff(mm1)]);
   Dw     = kfeffw*(Axc/dxc);
   De     = kfeffe*(Axc/dxc);
% Heat loss terms
   Scc   = (2*pi*(Hc/2)*dxc*Uoverall*Tinfinity)/Vbc;
   Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc ;
% Coeffcients
   aWf      = Dw+Fw;
   aEf      = De;
   aPf      = (AccTfnp1(m)+(De+Fe)+Dw+(hcc(m)*asf(m)*Vbc));
   aP0f     = AccTfn(m);
```

```
                bbTf    =   (hcc(m)*asf(m))*Vbc*Triter(m);

            %  Fluid Temperature SOR

                Tfnew_iter(m)  =
(aWf*Tfnew_iter(mm1)+aEf*Tfiter(mp1)+(aP0f*Tfn(m))+bbTf)/aPf;

                Tfnew_iter(m)  = (1-omega)*Tfiter(m)+omega*Tfnew_iter(m);

            %  Rock Temperature SOR

                Trnew_iter(m)  =
(aWr*Trnew_iter(mm1)+aEr*Triter(mp1)+(aP0r*Trn(m))+bbTs)/aPr;

                Trnew_iter(m)  = (1-omega)*Triter(m)+omega*Trnew_iter(m);

        else

            % Obtain flow velocity in m/s

                ue(m)   = -0.01*(Txu(m).*(Poutp-Pnew(m)))/(0.5*Axp);

                uw(m)   = -0.01*(Txu(mm1).*(Pnew(m)-Pnew(mm1)))/(Axp);

                up(m)   = 0.5*(ue(m)+uw(m));

 % Heat transfer between rock and fluid

                hcc(m) = (kwu(m)*(2+
1.1*(Prr(m)^(1/3))*((rhowup(m)*up(m)*dp)/(visup(m)*1e-3))^0.6))/dp;

            %  for rock temperature

                kreffw =  harmmean([ Kreff(m),Kreff(mm1)]);

                Drw    =  kreffw*(Axc/(dxc));

            % Coeffiecient

                aWr    =  Drw;

                aPr    =  (AccTrnp1(m)+Dre+Drw+(hcc(m)*asf(m)*Vbc));

                aP0r   = AccTrn(m);

                bbR    =  hcc(m)*asf(m)*Vbc*Tfiter(m);

            % For  Fluid Temperature
```

```
            Fe      = ue(m)*Axc*rhowup(m)*Cpwnp1(m);
            Fw      = uw(m)*Axc*rhowup(mm1)*Cpwnp1(mm1);
            kfeffw  =  harmmean([Kweff(m),Kweff(mm1)]);
            Dw      = kfeffw*(Axc/dxc);
          % Heat loss terms
           Scc   = (2*pi*(Hc/2)*dxc*Uoverall*Tinfinity)/Vbc;
           Spp   = -(2*pi*(Hc/2)*dxc*Uoverall)/Vbc;
          % Coeffcients
            aWf     = Dw+Fw;
            aPf     = (AccTfnp1(m)+(De+Fe)+Dw+(hcc(m)*asf(m)*Vbc));
            aP0f    = AccTfn(m);
            bbF     =  hcc(m)*asf(m)*Vbc*Triter(m);
          % Fluid Temp
           Tfnew_iter(m)  = (aWf*Tfnew_iter(mm1)+(aP0f*Tfn(m))+ bbF)/aPf;
           Tfnew_iter(m)  = (1-omega)*Tfiter(m)+omega*Tfnew_iter(m);
          % Rock Temp
           Trnew_iter(m)  = (aWr*Trnew_iter(mm1)+(aP0r*Trn(m))+bbR)/aPr;
           Trnew_iter(m)  = (1-omega)*Triter(m)+omega*Trnew_iter(m);
      end
end
 normP  =  sum(abs((Pnew_iter-Piter)/Piter));
 Piter  = Pnew_iter;
 normTr =  sum(abs((Trnew_iter-Triter)/Triter));
 Triter = Trnew_iter;
 normTf =  sum(abs((Tfnew_iter-Tfiter)/Tfiter));
```

```
        Tfiter = Tfnew_iter;
        disp('error is :')
        err    = max([normP,normTr, normTf])
        iter = iter+1
 end
 Tf3      = [Tf3 Tfiter];
 Tr3      = [Tr3 Triter];
 P3       = [ P3 Piter];  Tavgu = 0.5*(Triter+Tfiter);
 KKu      = Civan_PERM(Tavgu);
for i= 1:M
    muinew(i)  = XSteam('my_pT',Piter(i)*1.01325,Tfiter(i)-273.15)*1e3;
end
    Pinjj3(np1)    = Pinjp;
    rhownew       = rho(Piter,Tfiter);
    PoroT3(:,np1)  = Civan_PORO(Piter,Tavgu);
    Perm3(:,np1)   = KKu*1e3;
    Muuw3(:,np1)   = muinew';
    Rhoow3(:,np1)  = rhownew;
    Ue3(:,np1)     = ue(:);
    Uw3(:,np1)     = uw(:);
    Upp3(:,np1)    = up(:);
    Kimp3(:,np1)   = KKu./Kxi;
    qinj(np1)     = qp3*(timer);
    pvinj3         = (qc3*timer)/PVt3;
    disp('Volume of water injected, (cc)')
```

```
    Q            = qp3*(timer)
    disp('Simulation time in sec')
    t            = (t+dt)
end% time loop ends
toc
  tplot    = (0:dt:Tt)/60;
```

# References


[1] I.S. Nashawi, A. Malallah, M.J. Al Bisharah, Modeling World Oil Supply: Its Peak Production Rate and Time - Model Validation, in: SPE Asia Pacific Oil Gas Conf. Exhib., Society of Petroleum Engineers, 2010. doi:10.2118/133208-MS.

[2] S.A. Shedid, A.-A.A. Abbas, Comparison of Chemical Steam Floods through Vertical and Horizontal Wells, in: SPE/CIM Int. Conf. Horiz. Well Technol., Society of Petroleum Engineers, 2000. doi:10.2118/65482-MS.

[3] M.E. Hossain, S.A. Abu-khamsin, Development of dimensionless numbers for heat transfer in porous media using a memory concept., J. Porous Media. 15 (2012) 18.

[4] M.E. Hossain, S.A. Abu-khamsin, Utilization of memory concept to develop heat transfer dimensionless numbers for porous media undergoing thermal flooding with equal rock and fluid temperatures., J. Porous Media. 15 (2011) 18. doi:10.1615/JPorMedia.v15.i10.50.

[5] M.E. Hossain, S.A. Abu-Khamsin, A.-A. Al-Helali, Use of the Memory Concept to Investigate the Temperature Profile during a Thermal EOR Process:, in: SPE/DGS Saudi Arab. Sect. Tech. Symp. Exhib., Society of Petroleum Engineers, 2011. doi:http://dx.doi.org/10.2118/149094-MS.

[6] Y.T. Chan, S. Banerjee, Analysis of Transient Three-Dimensional Natural Convection in Porous Media, J. Heat Transfer. 103 (1981) 242. doi:10.1115/1.3244448.

[7] M. Kaviany, Principles of Heat Transfer in Porous Media, Springer New York, New York, NY, 1995. doi:10.1007/978-1-4612-4254-3.

[8] Z. Chen, Formulations and Numerical Methods of the Black Oil Model in Porous Media, SIAM J. Numer. Anal. 38 (2000) 489–514.

[9] H.A.H. Alahmadi, Triple Porosity Model for Frcactured Horizontal Wells, Texas A& M University, 2010.

[10] R.W. Zimmerman, G. Chen, T. Hadgu, G.S. Bodvarsson, A numerical dual-porosity model with semianalytical treatment of fracture/matrix flow, Water Resour. Res. 29 (1993) 2127–2137.

[11] A. de Swaan O, Analytic solutions for determining naturally fractured reservoir properties by well testing, Soc. Pet. Eng. J. 16 (1976) 117–122.

[12] H. Kazemi, Pressure transient analysis of naturally fractured reservoirs with uniform fracture distribution, Soc. Pet. Eng. J. 9 (1969) 451–462.

[13] H. Kazemi, L.S. Merrill Jr, K.L. Porterfield, P.R. Zeman, Numerical simulation of

water-oil flow in naturally fractured reservoirs, Soc. Pet. Eng. J. 16 (1976) 317–326.

[14] G.I. Barenblatt, I.P. Zheltov, I.N. Kochina, Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks [strata], J. Appl. Math. Mech. 24 (1960) 1286–1303.

[15] A.C. Gringarten, P.A. Witherspoon, A method of analyzing pump test data from fractured aquifers, Percolation through Fiss. Rock, Dtsch. Gesellschaft Fur Red Grundbau, Stuttgart. (1972) T3B1–T3B8.

[16] M.J. Mavor, H. Cinco-Ley, Transient pressure behavior of naturally fractured reservoirs, in: SPE Calif. Reg. Meet., Society of Petroleum Engineers, 1979.

[17] A.C. Gringarten, Flow-test evaluation of fractured reservoirs, Geol. Soc. Am. Spec. Pap. 189 (1982) 237–264.

[18] N.S. Boulton, T.D. Streltsova, Unsteady flow to a pumped well in a fissured water-bearing formation, J. Hydrol. 35 (1977) 257–270.

[19] H.L. Najurieta, A theory for pressure transient analysis in naturally fractured reservoirs, J. Pet. Technol. 32 (1980) 1–241.

[20] A.F. Moench, Double-porosity models for a fissured groundwater reservoir with fracture skin, Water Resour. Res. 20 (1984) 831–846.

[21] T. Arbogast, J. Douglas Jim, U. Hornung, Derivation of the double porosity model of single phase flow via homogenization theory, SIAM J. Math. Anal. 21 (1990) 823–836.

[22] J. Douglas Jr, T. Arbogast, Dual porosity models for flow in naturally fractured reservoirs, Dyn. Fluids Hierarchical Porous Media. (1990) 177–221.

[23] J. Douglas, J.L. Hensley, T. Arbogast, A dual-porosity model for waterflooding in naturally fractured reservoirs, Comput. Methods Appl. Mech. Eng. 87 (1991) 157–174.

[24] F. Civan, Porous media transport phenomena, John Wiley & Sons, 2011.

[25] H.H. Hardy, R.A. Beier, Fractals in reservoir engineering, World Scientific, 1994.

[26] M. Sahimi, Y.C. Yortsos, Applications of fractal geometry to porous media: a review, in: Annu. Fall Meet. Soc. Pet. Eng. New Orleans, LA, 1990.

[27] F.J. Molz, G.J. Fix, S. Lu, A physical interpretation for the fractional derivative in Levy diffusion, Appl. Math. Lett. 15 (2002) 907–911.

[28] H.T. Fomin SE, Chugunov V, Mathematical modeling of anomalous diffusion in porous media, Fract. Differ. Calc. 109 (2011) 187–205. doi:dx.doi.org/10.7153/fdc-01-01.

[29] J. Klafter, G. Zumofen, M.F. Shlesinger, Long-tailed distributions and non-Brownian transport in complex systems in: F. Mallamace, HE Stanley Eds., The Physics of Complex Systems, (1997).

[30] D.A. Benson, M.M. Meerschaert, J. Revielle, Fractional calculus in hydrologic modeling: A numerical perspective, Adv. Water Resour. 51 (2013) 479–497.

[31] E. Major, D.A. Benson, J. Revielle, H. Ibrahim, A. Dean, R.M. Maxwell, et al., Comparison of Fickian and temporally nonlocal transport theories over many scales in an exhaustively sampled sandstone slab, Water Resour. Res. 47 (2011).

[32] M.M. Meerschaert, Fractional calculus, anomalous diffusion, and probability, Fract. Dyn. (2012) 265–284.

[33] K. Razminia, A. Razminia, J.J. Trujilo, Analysis of radial composite systems based on fractal theory and fractional calculus, Signal Processing. 107 (2015) 378–388.

[34] R. Camacho-Velazquez, A. de Swaan-Oliva, M. Vasquez-Cruz, Interference Tests Analysis in Fractured Formations with a Time Fractional Equation, (2011).

[35] M. Sahimi, Y.C. Yortsos, Applications of Fractal Geometry to Porous Media: A Review, paper SPE 20476, in: SPE Annu. Tech. Conf. Exhib. New Orleans, LA, 1990.

[36] O. Ozcan, H. Sarak, E. Ozkan, R.S. Raghavan, A trilinear flow model for a fractured horizontal well in a fractal unconventional reservoir, in: SPE Annu. Tech. Conf. Exhib., Society of Petroleum Engineers, 2014.

[37] R. Raghavan, Fractional diffusion: performance of fractured wells, J. Pet. Sci. Eng. 92 (2012) 167–173.

[38] R. Raghavan, Fractional derivatives: Application to transient flow, J. Pet. Sci. Eng. 80 (2011) 7–13.

[39] R. Raghavan, C. Chen, Fractional diffusion in rocks produced by horizontal wells with multiple, transverse hydraulic fractures of finite conductivity, J. Pet. Sci. Eng. 109 (2013) 133–143. doi:http://dx.doi.org/10.1016/j.petrol.2013.08.027.

[40] B.B. Mandelbrot, The fractal geometry of nature/Revised and enlarged edition, New York, WH Free. Co., 1983, 495 P. 1 (1983).

[41] K. Falconer, Fractal geometry: mathematical foundations and applications, John Wiley & Sons, 2004.

[42] D. Avnir, O. Biham, D. Lidar, O. Malcai, Is the geometry of nature fractal, Science (80-. ). 279 (1998) 39–40.

[43] R.W. Holy, E. Ozkan, A Practical and Rigorous Approach for Production Data Analysis in Unconventional Wells, in: SPE Low Perm Symp., Society of Petroleum Engineers, 2016.

[44] R.R. Nigmatullin, To the theoretical explanation of the “universal response,” Phys. Status Solidi. 123 (1984) 739–745.

[45] Y. Dassas, P. Duby, Diffusion toward Fractal Interfaces Potentiostatic, Galvanostatic, and Linear Sweep Voltammetric Techniques, J. Electrochem. Soc. 142 (1995) 4175–4180.

[46] M. Caputo, Diffusion of fluids in porous media with memory, Geothermics. 28 (1998) 2–19. doi:http://dx.doi.org/10.1016/S0375-6505(98)00047-9.

[47] M. Caputo, C. Cametti, Diffusion with memory in two cases of biological interest., J. Theor. Biol. 254 (2008) 697–703. doi:10.1016/j.jtbi.2008.06.021.

[48] C. Chen, R. Raghavan, Transient flow in a linear reservoir for space–time fractional diffusion, J. Pet. Sci. Eng. 128 (2015) 194–202.

[49] K. Razminia, A. Razminia, J.A.T. Machado, Analysis of diffusion process in fractured reservoirs using fractional derivative approach, Commun. Nonlinear Sci. Numer. Simul. 19 (2014) 3161–3170.

[50] A. Cloot, J.F. Botha, A generalised groundwater flow equation using the concept of non-integer order derivatives, Water SA. 32 (2006) 1–7.

[51] A. Atangana, J.F. Botha, A generalized groundwater flow equation using the concept of variable-order derivative, Bound. Value Probl. 2013 (2013) 1–11.

[52] A. Atangana, N. Bildik, The use of fractional order derivative to predict the groundwater flow, Math. Probl. Eng. 2013 (2013).

[53] A. Atangana, J.F. Botha, Analytical solution of the groundwater flow equation obtained via homotopy decomposition method, J. Earth Sci. Clim. Change. 2012 (2012).

[54] J.A. Ochoa-Tapia, F.J. Valdes-Parada, J. Alvarez-Ramirez, A fractional-order Darcy’s law, Phys. A Stat. Mech. Its Appl. 374 (2007) 1–14.

[55] K. Razminia, A. Razminia, D. Baleanu, Investigation of the fractional diffusion equation based on generalized integral quadrature technique, Appl. Math. Model. 39 (2015) 86–98.

[56] A.D. Obembe, S.A. Abu-khamsin, M.E. Hossain, A Review of Modelling Thermal Displacement Processes in Porous media, Arab. J. Sci. Eng. In press (2016). doi:10.1007/s13369-016-2265-5.

[57] F. Medeiros, B. Kurtoglu, E. Ozkan, H. Kazemi, Analysis of Production Data From Hydraulically Fractured Horizontal Wells in Shale Reservoirs, SPE Reserv. Eval. Eng. 13 (2010).

[58] A. Albinali, E. Ozkan, Analytical Modeling of Flow in Highly Disordered, Fractured Nano-Porous Reservoirs, in: SPE West. Reg. Meet., Society of

Petroleum Engineers, 2016.

[59] D. Ben-Avraham, S. Havlin, Diffusion and reactions in fractals and disordered systems, Cambridge University Press, 2000.

[60] J.E. Warren, P.J. Root, The Behavior of Naturally Fractured Reservoirs, Soc. Pet. Eng. J. 3 (1963) 245–255. doi:10.2118/426-PA.

[61] R.O. Bello, Rate transient analysis in shale gas reservoirs with transient linear behavior rate transient analysis in shale gas reservoirs, 2009.

[62] E. Ozkan, R. Raghavan, P. Petroleum, C. Retd, O.G. Apaydin, SPE 134830 Modeling of Fluid Transfer from Shale Matrix to Fracture Network, SPE Annu. Tech. Conf. Exhib. (2010) 18. https://secure.mines.edu/mslib/app/,DanaInfo=www.onepetro.org+Preview.do?paperNumber=SPE-134830-MS&societyCode=SPE.

[63] H.A. Al Ahmadi, A.M. Almarzooq, R.A. Wattenbarger, Application of linear flow analysis to shale gas wells-field cases, in: SPE Unconv. Gas Conf., Society of Petroleum Engineers, 2010.

[64] R.O. Bello, R.A. Wattenbarger, Multi-stage hydraulically fractured horizontal shale gas well rate transient analysis, in: North Africa Tech. Conf. Exhib., Society of Petroleum Engineers, 2010.

[65] C.M. Du, X. Zhang, L. Zhan, H. Gu, B. Hay, K. Tushingham, et al., Modeling hydraulic fracturing induced fracture networks in shale gas reservoirs as a dual porosity system, in: Int. Oil Gas Conf. Exhib. China, Society of Petroleum Engineers, 2010.

[66] O. Samandarli, H.A. Al Ahmadi, R.A. Wattenbarger, A new method for history matching and forecasting shale gas reservoir production performance with a dual porosity model, in: North Am. Unconv. Gas Conf. Exhib., Society of Petroleum Engineers, 2011.

[67] J. Li, C. Du, X. Zhang, Critical evaluations of shale gas reservoir simulation approaches: single porosity and dual porosity modeling, in: SPE Middle East Unconv. Gas Conf. Exhib., Society of Petroleum Engineers, 2011.

[68] I.G. Brohi, M. Pooladi-Darvish, R. Aguilera, Modeling fractured horizontal wells as dual porosity composite reservoirs-application to tight gas, shale gas and tight oil cases, in: SPE West. North Am. Reg. Meet., Society of Petroleum Engineers, 2011.

[69] Y. Zhao, L. Zhang, J. Zhao, J. Luo, B. Zhang, “Triple porosity” modeling of transient well test and rate decline analysis for multi-fractured horizontal well in shale gas reservoirs, J. Pet. Sci. Eng. 110 (2013) 253–262.

[70] H.A. Al-Ahmadi, R.A. Wattenbarger, Triple-porosity models: one further step towards capturing fractured reservoirs heterogeneity, in: SPE/DGS Saudi Arab.

Sect. Tech. Symp. Exhib., Society of Petroleum Engineers, 2011.

[71] V. Tivayanonda, S. Apiwathanasorn, C. Economides, R. Wattenbarger, Alternative interpretations of shale gas/oil rate behavior using a triple porosity model, in: SPE Annu. Tech. Conf. Exhib., Society of Petroleum Engineers, 2012.

[72] A.D. Obembe, M.E. Hossain, A New Pseudosteady Triple-Porosity Model For Naturally Fractured Shale Gas Reservoir, in: SPE Annu. Tech. Conf. Exhib., Society of Petroleum Engineers, 2015. doi:http://dx.doi.org/10.2118/178726-STU.

[73] F. Kuchuk, D. Biryukov, Pressure-transient behavior of continuously and discretely fractured reservoirs, SPE Reserv. Eval. Eng. 17 (2014) 82–97.

[74] F. Kuchuk, D. Biryukov, Pressure-transient tests and flow regimes in fractured reservoirs, SPE Reserv. Eval. Eng. 18 (2015) 187–204.

[75] O.O. Thomas, R.S. Raghavan, T.N. Dixon, Effect of scaleup and aggregation on the use of well tests to identify geological properties, SPE Reserv. Eval. Eng. 8 (2005) 248–254.

[76] C. V Theis, The relation between the lowering of the Piezometric surface and the rate and duration of discharge of a well using ground-water storage, Eos, Trans. Am. Geophys. Union. 16 (1935) 519–524.

[77] B. Baeumer, S. Kurita, M. Meerschaert, Inhomogeneous fractional diffusion equations, Fract. Calc. Appl. Anal. 8 (2005) 371–386.

[78] Q. Huang, G. Huang, H. Zhan, A finite element solution for the fractional advection–dispersion equation, Adv. Water Resour. 31 (2008) 1578–1589.

[79] B.P. Sprouse, Computational efficiency of fractional diffusion using adaptive time step memory and the potential application to neural glial networks, (2010).

[80] H. Darcy, Les fontaines publiques de la ville de Dijon: exposition et application ... (Google eBook), Victor Dalmont, 1856.

[81] P.M. Adler, A.E. Malevich, V. V Mityushev, Nonlinear correction to Darcy's law for channels with wavy walls, Acta Mech. 224 (2013) 1823–1848.

[82] J. Kim, J. Lee, K.-C. Lee, Nonlinear correction to Darcy's law for a flow through periodic arrays of elliptic cylinders, Phys. A Stat. Mech. Its Appl. 293 (2001) 13–20.

[83] J.-L. Auriault, C. Boutin, C. Geindreau, Homogenization of coupled phenomena in heterogenous media, John Wiley & Sons, 2010.

[84] J. Bear, Dynamics of fluids in porous media, Courier Corporation, 2013.

[85] G. Sposito, General Criteria for the Validity of the Buckingham-Darcy Flow Law1, Soil Sci. Soc. Am. J. 44 (1980) 1159. doi:10.2136/sssaj1980.03615995004400060006x.

[86] H.C. Brinkman, A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles, Appl. Sci. Res. 1 (1949) 27–34. doi:10.1007/BF02120313.

[87] S. Whitaker, Flow in porous media I: A theoretical derivation of Darcy's law, Transp. Porous Media. 1 (1986) 3–25. doi:10.1007/BF01036523.

[88] S. Martino, M. Caputo, G. Iaffaldano, Experimental and theoretical memory diffusion of water in sand, Hydrol. Earth Syst. Sci. Discuss. 10 (2006) 93–100.

[89] E. Di Giuseppe, M. Moroni, M. Caputo, Flux in porous media with memory: models and experiments, Transp. Porous Media. 83 (2010) 479–500.

[90] M. Caputo, W. Plastino, Diffusion in porous layers with memory, Geophys. J. Int. 158 (2004) 385–396. doi:10.1111/j.1365-246X.2004.02290.x.

[91] G. Iaffaldano, M. Caputo, S. Martino, Experimental and theoretical memory diffusion of water in sand, Hydrol. Earth Syst. Sci. Discuss. 2 (2005) 1329–1357. doi:10.5194/hessd-2-1329-2005.

[92] J. Bear, Dynamics of fluids in porous media, Am. Else-Vier, New York. (1972).

[93] M. Caputo, Models of flux in porous media with memory, Water Resour. Res. 36 (2000) 693–705. doi:10.1029/1999WR900299.

[94] M. Caputo, Models of flux in porous media with memory for a pressure plane, 36 (2000) 693–705.

[95] M. Caputo, W. Plastino, Diffusion with space memory, in: Geod. Chall. 3rd Millenn., Springer, 2003: pp. 429–435.

[96] E. Di Giuseppe, M. Moroni, M. Caputo, Flux in porous media with memory : Models and experiments, (n.d.) 3–4.

[97] F.J. Fayers, Some theoretical results concerning the displacement of a viscous oil by a hot fluid in a porous medium, J. Fluid Mech. 13 (1962) 65–76.

[98] G.S. Bodvarsson, S.M. Benson, P.A. Witherspoon, Theory of the development of geothermal systems charged by vertical faults, J. Geophys. Res. Solid Earth. 87 (1982) 9317–9328.

[99] L.A. Wilson, P.J. Root, Cost Comparison of Reservoir Heating Using Steam or Air: SPE-1116-PA, J. Pet. Technol. 18 (1966) 233–239. doi:10.2118/1116-PA.

[100] J. Ferrer, A three-phase, two-dimensional compositional thermal simulator for steam injection processes, J. Can. Pet. Technol. 16 (1977) 78–90. http://dx.doi.org/10.2118/77-01-07.

[101] J.H. Abou-Kassem, Investigation of grid orientation in a two-dimensional, compositional, three-phase steam model, Chemical and Petroleum Engineering, University of Calgary, 1981. http://hdl.handle.net/1880/22121.

[102] K. Ishimoto, G.A. Pope, K. Sepchrnoori, An equation-of-state steam simulator, In Situ;(United States). 11 (1987) 1–37.

[103] R.L. Spillette, A. G., & Nielsen, Two-Dimensional Method For Predicting Hot Waterflood Recovery Behavior, J. Pet. Technol. 20 (1967) 627–638. doi:10.2118/1895-PA.

[104] J.K. Jordon, J.R. Rayne, S.W. Marshall, A Calculation Procedure for Estimating the Production History During Hot Water Injection in Linear Reservoirs, in: 20th Tech. Conf. Pet. Prod. Pennsylvania State U., Univ. Park. Pa.(May 9-10, 1957), 1957.

[105] P.K.W. Vinsome, A Numerical Description of Hot-Water and Steam Drives By The Finite-Difference Method, in: Fall Meet. Soc. Pet. Eng. AIME, Society of Petroleum Engineers, 1974. doi:10.2118/5248-MS.

[106] H. Class, R. Helmig, P. Bastian, Numerical simulation of non-isothermal multiphase multicomponent processes in porous media.: 1. An efficient solution technique, Adv. Water Resour. 25 (2002) 533–550.

[107] T.B. Jensen, M.P. Sharma, H.G. Harris, D.L. Whitman, Numerical investigations of steam and hot-water flooding in fractured porous media, in: SPE/DOE Enhanc. Oil Recover. Symp., Society of Petroleum Engineers, 1992.

[108] K.D. Dreher, D.E. Kenyon, F.O. Iwere, Heat Flow during Steam Injection into a Fractured Carbonate Reservoir, paper SPE 14902 presented at the 1986 SPE, in: DOE Fifth Symp. Enhanc. Oil Recover. Tulsa, Oklahoma, April, 1986: pp. 20–23.

[109] B.C. Sahuquet, J.J. Ferrier, Steam-drive pilot in a fractured carbonated reservoir: Lacq Superieur field, J. Pet. Technol. 34 (1982) 873–880.

[110] J.C. Reis, Oil recovery mechanisms in fractured reservoirs during steam injection, in: SPE/DOE Enhanc. Oil Recover. Symp., Society of Petroleum Engineers, 1990.

[111] M.D. Sumnu, W.E. Brigham, K. Aziz, L.M. Castanier, An experimental and numerical study on steam injection in fractured systems, in: SPE/DOE Improv. Oil Recover. Symp., Society of Petroleum Engineers, 1996.

[112] Lord Rayleigh, LIX. On convection currents in a horizontal layer of fluid, when the higher temperature is on the under side, Philos. Mag. Ser. 6. 32 (1916) 529–546. doi:10.1080/14786441608635602.

[113] P.S. Karra, K. Aziz, A Numerical Study of Transient Natural Convection in Porous Media, in: Proc. 17th Annu. Conf. Can. Soc. Chem. Eng. Ontario, 1967.

[114] P. Sarathi, Thermal numerical simulator for laboratory evaluation of steamflood oil recovery, National Inst. for Petroleum and Energy Research, Bartlesville, OK (USA), 1991. doi:http://dx.doi.org/10.2172/5746567.

[115] N.D. Shutler, Numerical, Three-Phase Simulation of the Linear Steamflood

Process: SPE-2233-PA, Soc. Pet. Eng. J. 9 (1969) 232–246. doi:10.2118/2233-PA.

[116] S. Abdus, D.R. Parrish, A Two-Dimensional Analysis of Reservoir Heating by Steam Injection, Soc. Pet. Eng. J. 11 (1971) 185–197. doi:10.2118/2515-PA.

[117] E.E. Gomaa, Correlations for Predicting Oil Recovery by Steamflood, J. Pet. Technol. 32 (1980) 325–332. doi:10.2118/6169-PA.

[118] A. Abdalla, K.H. Coats, A three-phase, experimental and numerical simulation study of the steam flood process, in: Fall Meet. Soc. Pet. Eng. AIME, Society of Petroleum Engineers, 1971.

[119] K.H. Coats, Simulation of steamflooding with distillation and solution gas, Soc. Pet. Eng. J. 16 (1976) 235–247. http://dx.doi.org/10.2118/5015-PA.

[120] R.W. Falta, K. Pruess, I. Javandel, P.A. Witherspoon, Numerical modeling of steam injection for the removal of nonaqueous phase liquids from the subsurface: 1. Numerical formulation, Water Resour. Res. 28 (1992) 433–449.

[121] R.W. Falta, K. Pruess, I. Javandel, P.A. Witherspoon, Numerical modeling of steam injection for the removal of nonaqueous phase liquids from the subsurface: 2. Code validation and application, Water Resour. Res. 28 (1992) 451–465.

[122] M.E. Hossain, S.H. Mousavizadegan, M.R. Islam, Rock and fluid temperature changes during thermal operation in EOR process., J. Nat. Sci. Sustain. Technol. 2 (2007) 347–378.

[123] M.E. Hossain, S.H. Mousavizadegan, M.R. Islam, Variation of Rock and Fluid Temperature During Thermal Operation in Porous Media, Pet. Sci. Technol. 27 (2009) 597–611. doi:10.1080/10916460802105526.

[124] O. Cicek, Numerical simulation of steam displacement of oil in naturally fractured reservoirs using fully implicit compositional formulation: A comparative analysis of the effects of capillary and gravitational forces in matrix/fracture exchange term, in: SPE Annu. Tech. Conf. Exhib., Society of Petroleum Engineers, 2005.

[125] O. Cicek, A parametric study of the effects of reservoir and operational properties on the performance of steam displacement of heavy oil in naturally fractured reservoirs, in: SPE Int. Therm. Oper. Heavy Oil Symp., Society of Petroleum Engineers, 2005.

[126] K. Wu, X. Li, Y. Zhai, The Model for Predicting Stream Breakthrough Timing during Steam Drive Development of Heavy Oil Reservoirs: SPE-150504-MS, in: SPE Heavy Oil Conf. Exhib., Society of Petroleum Engineers, 2011. doi:10.2118/150504-MS.

[127] M.E. Hossain, S.H. Mousavizadegan, M.R. Islam, A new porous media diffusivity equation with the inclusion of rock and fluid memories: SPE-114287-MS, Society of Petroleum Engineers, 2008.

[128] M.E. Hossain, Comprehensive modelling of complex petroleum phenomena with an engineering approach, J. Porous Media. 15 (2012) 173–186.

[129] M.E. Hossain, S.H. Mousavizadegan, M.R. Islam, The Effects of Thermal Alterations on Formation Permeability and Porosity, Pet. Sci. Technol. 26 (2008) 1282–1302. doi:10.1080/10916460701834028.

[130] F. Civan, Non-isothermal Permeability Impairment by Fines Migration and Deposition in Porous Media including Dispersive Transport, Transp. Porous Media. 85 (2010) 233–258. doi:10.1007/s11242-010-9557-0.

[131] N. Yoshida, D. Zhu, A.D. Hill, Temperature Prediction Model For A Horizontal Well With Multiple Fractures In A Shale Reservoir, in: SPE Annu. Tech. Conf. Exhib., Society of Petroleum Engineers, 2013. doi:10.2118/166241-MS.

[132] N. Yoshida, D. Zhu, A.D. Hill, Temperature-Prediction Model for a Horizontal Well With Multiple Fractures in a Shale Reservoir, SPE Prod. Oper. 29 (2014) 261–273. doi:10.2118/166241-PA.

[133] S. Mozaffari, M. Nikookar, M.R. Ehsani, L. Sahranavard, E. Roayaie, A.H. Mohammadi, Numerical modeling of steam injection in heavy oil reservoirs, Fuel. 112 (2013) 185–192.

[134] S. Irawan, M. Bathaee, Numerical Modeling of Flow and Temperature Distribution in Heterogeneous Wellbore Medium : SPE-172568-MS, in: SPE Middle East Oil Gas Show Conf., Society of Petroleum Engineers, 2015. doi:10.2118/172568-MS.

[135] J. Boussinesq, Théorie analytique de la chaleur: mise en harmonie avec la thermodynamique et avec la théorie mécanique de la lumière, Gauthier-Villars, 1903.

[136] A. Oberbeck, Ueber die Wärmeleitung der Flüssigkeiten bei Berücksichtigung der Strömungen infolge von Temperaturdifferenzen, Ann. Phys. 243 (1879) 271–292.

[137] D. Gartling, C. Hickox, A numerical study of the applicability of the Boussinesq approximation for a fluid saturated porous medium, J. Numer. Methods Fluids. 5 (1985) 19.

[138] A.J. Landman, R.J. Schotting, Heat and brine transport in porous media: The Oberbeck-Boussinesq approximation revisited, Transp. Porous Media. 70 (2007) 355–373.

[139] T.J. McDougall, R.J. Greatbatch, Y. Lu, On Conservation Equations in Oceanography: How Accurate Are Boussinesq Ocean Models?, J. Phys. Oceanogr. 32 (2002) 1574–1584. doi:10.1175/1520-0485(2002)032<1574:OCEIOH>2.0.CO;2.

[140] M.B. Peirotti, M.D. Giavedoni, J.A. Deiber, Natural convective heat transfer in a rectangular porous cavity with variable fluid properties—validity of the Boussinesq approximation, Int. J. Heat Mass Transf. 30 (1987) 2571–2581.

[141] D. Marpu, V. Satyamurty, Investigations on the validity of Boussinesq approximation on free convection in vertical porous annulus, Wärme-Und Stoffübertragung. 147 (1991) 141–147.

[142] K. Johannsen, On the validity of the Boussinesq approximation for the Elder problem, Comput. Geosci. 7 (2003) 169–182.

[143] A.J. Landman, Analysis of Physical Mechanisms Underlying Density-Dependent Transport in Porous Media, Technische Universiteit Delft, 2005.

[144] K.R. Rajagopal, G. Saccomandi, L. Vergori, A systematic approximation for the equations governing convection–diffusion in a porous medium, Nonlinear Anal. Real World Appl. 11 (2010) 2366–2375. doi:10.1016/j.nonrwa.2009.07.010.

[145] D.L. Turcotte, A.T. Hsui, K.E. Torrance, G. Schubert, Influence of viscous dissipation on Bénard convection, J. Fluid Mech. 64 (2006) 369. doi:10.1017/S0022112074002448.

[146] J.M. Hewitt, D.P. Mckenzie, N.O. Weiss, Dissipative heating in convective flows, J. Fluid Mech. 68 (2006) 721. doi:10.1017/S002211207500119X.

[147] D.P. Mckenzie, J.M. Roberts, N.O. Weiss, Convection in the earth's mantle: towards a numerical simulation, J. Fluid Mech. 62 (1974) 465.

[148] M.G. Velarde, R. Perez Cordon, On the (non-linear) foundations of boussinesq approximation applicable to a thin layer of fluid. (II). viscous dissipation and large cell gap effects, J. Phys. 37 (1976) 177–182. doi:10.1051/jphys:01976003703017700.

[149] B.J. Bayly, C.D. Levermore, T. Passot, Density variations in weakly compressible flows, Phys. Fluids A Fluid Dyn. 4 (1992) 945. doi:10.1063/1.858275.

[150] J. Rodrigues, J. Urbano, On the stationary Boussinesq-Stefan problem with constitutive power-laws, Int. J. Non. Linear. Mech. 33 (1998) 555–566. doi:10.1016/S0020-7462(97)00041-3.

[151] P.A. Litsek, A. Bejan, Convection in the Cavity Formed Between Two Cylindrical Rollers, J. Heat Transfer. 112 (1990) 625. doi:10.1115/1.2910433.

[152] N. and S.T.N. Rudraiah, Natural Convection Through Vertical Porous Media, Int. J. Engng Sci. 15 (1977) 589–600.

[153] H.C. Brinkman, On the permeability of media consisting of closely packed porous particles, Appl. Sci. Res. 1 (1949) 81–86. doi:10.1007/BF02120318.

[154] M.V. and M.S.M. N. Rudraiah, Oberbeck convection through vertical porous stratum.pdf, (1982) 21.

[155] V. Prasad, F.A. Kulacki, Convective heat transfer in a rectangular porous cavity—effect of aspect ratio on flow structure and heat transfer, J. Heat Transfer. 106

(1984) 158–165. doi:http://dx.doi.org/10.1115/1.3246629.

[156] G. V Hadjisophocleous, A.C.M. Sousa, Three-dimensional numerical predictions of internally heated free convective flows, Wärme-Und Stoffübertragung. 21 (1987) 283–290.

[157] J.P. Van Doormaal, G.D. Raithby, Enhancements of the simple method for predicting incompressible fluid flows., Numer. Heat Transf. 7 (1984) 147–163. doi:10.1080/01495728408961817.

[158] P.H. Oosthuizen, Mixed convective heat transfer from a heated horizontal plate in a porous medium near an impermeable surface, J. Heat Transfer. 110 (1988) 390–394.

[159] J.C. Umavathi, M.S. Malashetty, Oberbeck convection of Flow of a couple stress Fluid through a vertical porous stratum, 34 (1999) 1037–1045.

[160] K.M. Khanafer, A.J. Chamkha, Mixed convection flow in a lid-driven enclosure filled with a fluid-saturated porous medium, Int. J. Heat Mass Transf. 42 (1999) 2465–2481.

[161] M.A. Hossain, M. Wilson, Natural convection flow in a fluid-saturated porous medium enclosed by non-isothermal walls with heat generation, Int. J. Therm. Sci. 41 (2002) 447–454.

[162] S.N. Gaikwad, S.S. Kamble, The effect of Soret parameter on the onset of double diffusive convection in a Darcy porous medium saturated with couple stress fluid, 3 (2012) 1426–1434.

[163] T.Y. Na, I. Pop, Free convection flow past a vertical flat plate embedded in a saturated porous medium, Int. J. Eng. Sci. 21 (1983) 517–526. doi:10.1016/0020-7225(83)90099-X.

[164] P. Cheng, W.J. Minkowycz, Free Convection About A Vertical Flat Plate Embedded In A Porous Medium With Application To Heat TransferFrom A Dike., J Geophys Res. 82 (1977) 2040–2044.

[165] W.J. Minkowycz, P. Cheng, C.H. Chang, Mixed Convection About A Nonisothermal Cylinder And Sphere In A Porous Medium, Numer. Heat Transf. 8 (2007) 349–359. doi:10.1080/01495728508961859.

[166] A.P. Bassom, D.A.S. Rees, Free convection from a heated vertical cylinder embedded in a fluid-saturated porous medium, Acta Mech. 116 (1996) 139–151.

[167] P. Cheng, T.T. Le, I. Pop, Natural convection of a Darcian fluid about a cone, Int. Commun. Heat Mass Transf. 12 (1985) 705–717. doi:10.1016/0735-1933(85)90023-5.

[168] K.A. Yih, The effect of uniform lateral mass flux on free convection about a vertical cone embedded in a saturated porous medium, Int. Commun. Heat Mass

Transf. 24 (1997) 1195–1205. doi:10.1016/S0735-1933(97)00114-0.

[169] I. Pop, N. Tsung-Yen, Naturnal convection of a Darcian fluid about a wavy cone, Int. Commun. Heat Mass Transf. 21 (1994) 891–899. doi:10.1016/0735-1933(94)90042-6.

[170] S. Ghosh Moulic, L.S. Yao, Mixed Convection Along a Wavy Surface, J. Heat Transfer. 111 (1989) 974. doi:10.1115/1.3250813.

[171] S. Ghosh Moulic, L.S. Yao, Natural Convection Along a Vertical Wavy Surface With Uniform Heat Flux, J. Heat Transfer. 111 (1989) 1106. doi:10.1115/1.3250780.

[172] C.M. Vest, V.S. Arpaci, Overstability of a viscoelastic fluid layer heated from below, J. Fluid Mech. 36 (1969) 613. doi:10.1017/S0022112069001881.

[173] P.K. Bhatia, J.M. Steiner, Convection in a viscoelastic fluid layer in hydromagnetics, Phys. Lett. A. 37 (1971) 419–420. doi:10.1016/0375-9601(71)90613-X.

[174] P. Bhatia, J. Steiner, Oscillatory Convection in a Viscoelastic Fluid Layer in Hydromagnetics, Aust. J. Phys. 25 (1972) 695. doi:10.1071/PH720695.

[175] P.. Bhatia, J.. Steiner, Thermal instability in a viscoelastic fluid layer in hydromagnetics, J. Math. Anal. Appl. 41 (1973) 271–283. doi:10.1016/0022-247X(73)90201-1.

[176] Y.T. Shah, J.R.A. Pearson, Stability of non-isothermal flow in channels—II, Chem. Eng. Sci. 29 (1974) 737–746. doi:10.1016/0009-2509(74)80190-9.

[177] A.B. Datta, V.U.K. Sastry, Thermal instability of a horizontal layer of micropolar fluid heated from below, Int. J. Eng. Sci. 14 (1976) 631–637. doi:10.1016/0020-7225(76)90005-7.

[178] M.E.H. and M.R. Islam, A Comprehensive Material Balance Equation with the Inclusion of Memory During Rock-Fluid Deformation, Adv. Sustain. Pet. Eng. Sci. 1 (2009) 141–162.

[179] N. Arihara, A study of non-isothermal single and two-phase flow through consolidated sandstones, Stanford Univ., CA (USA). Stanford Geothermal Program, 1974. http://www.osti.gov/scitech/biblio/7306602.

[180] H. Jamal, F.A. SM, I. M Rafiq, Petroleum Reservoir Simulation: A Basic Approach, Gulf Publishing Company, 2006.

[181] C.L. MacDonald, N. Bhattacharya, B.P. Sprouse, G.A. Silva, Efficient computation of the Grünwald-Letnikov fractional diffusion derivative using adaptive time step memory, J. Comput. Phys. (2015). doi:10.1016/j.jcp.2015.04.048.

[182] O. Ozcan, Fractional Diffusion in Naturally Fractured Unconventional Reservoirs, ProQuest Dissertations Publishing, 2014. http://hdl.handle.net/11124/10641.

[183] A.A. Awotunde, R.A. Ghanam, N. Tatar, Artificial boundary condition for a modified fractional diffusion problem, Bound. Value Probl. 2015 (2015) 1–17.

[184] A.A. Awotunde, R.A. Ghanam, S.S. Al-Homidan, T. Nasser-eddine, Numerical Schemes for Anomalous Diffusion of Single-Phase Fluids In Porous Media, Commun. Nonlinear Sci. Numer. Simul. (2016).

[185] D.A. Nield, A. Bejan, Convection in Porous Media, Springer New York, 2006. doi:10.1007/978-0-387-76543-3.

[186] K. Vafai, Handbook of porous media, Crc Press, 2015.

[187] N. Arihara, A study of non-isothermal single and two-phase flow through consolidated sandstones, Stanford Univ., CA (USA). Stanford Geothermal Program, 1974.

[188] W.S. Tortike, S.M. Ali, Saturated-steam-property functional correlations for fully implicit thermal reservoir simulation, SPE Reserv. Eng. 4 (1989) 471–474. doi:http://dx.doi.org/10.2118/17094-PA.

[189] S.A. Bories, M.A. Combarnous, Natural convection in a sloping porous layer, J. Fluid Mech. 57 (1973) 63–79.

[190] S. Bories, Natural convection in porous media, in: Adv. Transp. Phenom. Porous Media, Springer, 1987: pp. 77–141.

[191] T. Ertekin, J.H. Abou-Kassem, G.R. King, Basic applied reservoir simulation, Society of Petroleum Engineers Richardson, TX, 2001.

[192] S. Fomin, V. Chugunov, T. Hashida, Application of fractional differential equations for modeling the anomalous diffusion of contaminant from fracture into porous rock matrix with bordering alteration zone, Transp. Porous Media. 81 (2010) 187–205. doi:10.1007/s11242-009-9393-2.

[193] H.A. Nooruddin, M.E. Hossain, Modified Kozeny–Carmen correlation for enhanced hydraulic flow unit characterization, J. Pet. Sci. Eng. 80 (2011) 107–115.

[194] S.S. Ali, S. Nizamuddin, A. Abdulraheem, M.R. Hassan, M.E. Hossain, Hydraulic unit prediction using support vector machine, J. Pet. Sci. Eng. 110 (2013) 243–252.

[195] R.A. Almehaideb, Improved correlations for fluid properties of UAE crude oils, Pet. Sci. Technol. 21 (2003) 1811–1831. doi:10.1081/LFT-120024563.

[196] D.W. Peaceman, Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability, Soc. Pet. Eng. J. 23 (1983) 531–543.

[197] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for fractional advection–dispersion flow equations, J. Comput. Appl. Math. 172 (2004) 65–77.

[198] F. Barends, Complete Solution for Transient Heat Transport in Porous Media, Following Lauwerier's Concept, in: SPE Annu. Tech. Conf. Exhib., Society of Petroleum Engineers, 2010. doi:10.2118/134670-MS.

[199] S.B. Yuste, L. Acedo, On an explicit finite difference method for fractional diffusion equations, (2003) 22. http://arxiv.org/abs/cs/0311011.

# Vitae


Name :Obembe Abiola David

Nationality :Nigerian

Date of Birth :11/29/1983

Email :obeabi@yahoo.com

Address :Dhahran


Academic Background :PhD (Petroleum Engineering), May 2016

King Fahd University of Petroleum and Minerals

Dhahran, Saudi Arabia

MSc (Petroleum Engineering), September 2009

Heriot Watt University

Edinburgh, United Kingdom

BSc (Electrical and Electronics Engineering), November 2006

Obafemi Awolowo University

Ile-Ife, Osun

Nigeria