2015-09-22

Transient Shape Factors for Thermal Flow Simulation in Fractured Reservoirs

Taabbodi, Loran

Taabbodi, L. (2015). Transient Shape Factors for Thermal Flow Simulation in Fractured Reservoirs (Unpublished doctoral thesis). University of Calgary, Calgary, AB. doi:10.11575/PRISM/28512
http://hdl.handle.net/11023/2476
doctoral thesis

UNIVERSITY OF CALGARY

Transient Shape Factors for Thermal Flow Simulation in Fractured Reservoirs

by

Loran Taabbodi

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN CHEMICAL AND PETROLEUM ENGINEERING

CALGARY, ALBERTA

SEPTEMBER, 2015

ABSTRACT

The simulation of naturally fractured reservoirs (NFRs) is an extremely daunting and challenging task. The most important and difficult aspect of modeling a naturally fractured reservoir is the accurate estimation of the fluid exchange between matrix and surrounded fractures, which is modeled via a shape factor.

The main focuses of this dissertation are the characterization and understanding as to how shape factors are utilized for modeling non-isothermal, fracture-matrix fluid exchange in fractured reservoirs. The intention of this research, therefore, is to achieve a better understanding of the matter exchanges between fractures and matrix blocks in a dual-porosity system by defining an appropriate transient matrix-fracture shape factor formulation that accounts for a multiphase flow system in thermal reservoir simulation of naturally fractured reservoirs. A novel and more general technique are developed for the exchange of fluids between the matrix blocks and the surrounded fractures. The concept of a new transient shape factor is introduced for non-isothermal, dual-porosity models and validated against the most current shape factor formulations. In this study, the CMG STARS simulator is used to estimate and validate the proposed transient shape factor. A MATLAB code is developed and coupled with the CMG STARS to compute transient shape factor (TSF). This new technique describes the most appropriate way of treating a shape factor to capture the pertinent features of non-isothermal fluid flow in NFRs. A robust transformation function (T.F) formulation is developed between the fractures and the matrix for a better representation of a dual-porosity system in thermal recovery. The

proposed transient shape factor (TSF) model is validated against a fine grid single porosity model and an historical field data to confirm the model capability for reproducing accurate physical processes and recovery mechanisms. The results clearly demonstrate that a constant value of the shape factor (e.g., Kazemi's formulation) cannot be used to predict the overall performance in dual-media systems due to invalid assumptions. The conclusions from this study clearly show that a suitable and accurate transfer transient shape factor is required for an appropriate modeling of a thermal recovery process in NFRs when using dual-porosity formulations.

Acknowledgements

I would like to express my deepest gratitude to all those who provided me the opportunity to complete this thesis.

First and foremost, this dissertation might not have been written without the help of my wife, whom I am truly grateful. She provided me with her support and faith in my abilities to handle the challenges throughout my PhD program. She is my excellent source of encouragement and inspiration. Likewise, I wish to thank my lovely children who are part of my life forever.

I am also thankful to my supervisor Dr. Zhangxing John Chen for his support, Dr. Sebastian Geiger from Heriot-Watt University and all the friends and colleagues for their valuable comments on this work.

Lastly, but of great significance, I have to thank Husky Energy for the financial support for this study and also the Department of Chemical and Petroleum Engineering at the University of Calgary for the opportunity.

Dedication

I dedicate this work to my wife for her unconditional love and support and my lovely children.

TABLE OF CONTENTS

ABSTRACT ii

TABLE OF CONTENTS vi

LIST OF TABLES viii

LIST OF FIGURES ix

NOMENCLATURE xii

SUBSCRIPTS xiv

CHAPTER 1 1

Introduction 1

CHAPTER 2 6

2.1. Background and Literature Review 6

2.2. Transfer Function (TF) 7

2.3. Concluding Remarks 25

CHAPTER 3 26

Statement of the Problem 26

CHAPTER 4 29

4.1. Shape Factors 29

4.1.1. 2D-Single and Multi-Set Matrix Blocks 31

4.1.2. 3D Multi-Set Matrix Blocks 35

4.1.3. Transient Shape Factors, TSF (Concept) 37

4.2. Transient Shape Factors (TSF) in Thermal Flow Simulation 39

4.2.1. Model Description of Non-isothermal Process 41

4.2.2. Solution Methodology 43

4.2.3. Utilizing Transient Shape Factors (TSF) 48

4.3. Concluding Remarks 53

CHAPTER 5 55

5.1. Sensitivity Study 55

5.1.1. Grid Block Size Distribution 56

5.1.2. Rock Wettability 58

5.1.3. Capillary Pressure 61

5.1.4. Initial Water Saturation 64

5.1.5. Reservoir Heterogeneity 65

5.1.6. Fracture-Matrix Effective Permeability Ratio (k_f / k_m)........ 69

5.2. Concluding Remarks 71

CHAPTER 6 73

6. Model Validation 73

6. 1. A Case Study 74

CHAPTER 7 79

Summary and Conclusions 79

7.1. Summary of Results 79

7.2. Recommendations for Future Studies 83

REFERENCES 85

Appendix A: Mass and Heat Equations 92

Appendix B: MATLAB Code 97

LIST OF TABLES

Table 2.2. 1: Constant shape factor by various researchers 24

Table 4.1.1. 1: Matrix and fracture parameters 32

LIST OF FIGURES

Figure 2.2. 1: Idealization of the NFR system (Warren and Root) 8

Figure 2.2. 2: Idealization of the NFR system (Kazemi) 10

Figure 2.2. 3: Comparison of the average pressure (Pallav, 2003) 15

Figure 2.2. 4: Pressure comparison of analytical vs. numerical methods 17

Figure 2.2. 5: Temperature profile comparison, 1D case (A. Van Heel) 19

Figure 2.2. 6: Temperature profile comparison, 1D case (A. Van Heel) 19

Figure 2.2. 7: Oil Rate profile comparison, 1D case (A. Van Heel) 20

Figure 2.2. 8: Temperature profile comparison, 1D case (A. Van Heel) 21

Figure 2.2. 9: Temperature profile comparison, different shape factor formulation (A. Van Heel) 22

Figure 4.1.1. 1: 2D Single-matrix block models; a) single-porosity; b) dual-porosity..... 32

Figure 4.1.1. 2: Pressure drawdown comparison, single-porosity vs, Kazemi's dual-porosity (2D model) 33

Figure 4.1.1. 3: 2D Multi-matrix blocks models; a) single-porosity; b) dual-porosity 34

Figure 4.1.1. 4: Pressure drawdown comparison for different shape factor formulations (2D, multi-matrix blocks) 35

Figure 4.1.2. 1: Matrix permeability (md); a) fine-grid single-porosity; b) dual-porosity 36

Figure 4.1.2. 2: Pressure drawdown comparison, single-porosity vs, Kazemi's dual-porosity (3D model) 37

Figure 4.1.3. 1: Pressure profile, transient shape factor concept (Kazemi's model) 39

Figure 4.2.2. 1: Flowchart of TSF Estimator Tool kit 47

Figure 4.2.2. 2: TSF Tool kit interface 48

Figure 4.2.3. 1: 2D Multi-matrix blocks dual-porosity model (stem drive) 49

Figure 4.2.3. 2: Average Pressure Profile, with and without TSF model 50

Figure 4.2.3. 3: Average oil saturation Profile, with and without TSF model.................. 51

Figure 4.2.3. 4: Cumulative oil steam ratio (COSR), with and without TSF model 51

Figure 4.2.3. 5: Dimensions less pressure coefficient factors (PDt = α) 53

Figure 5.1.1. 1:: Variable random block size distribution (dual-porosity) 57

Figure 5.1.1. 2: Outcrop from fractured carbonate formation 57

Figure 5.1.1. 3: Block size distributions sensitivity against reference solution................ 58

Figure 5.1.2. 1: Average Pressure Profile Comparison, Oil-Wet system, with and without TSF.. 60

Figure 5.1.2. 2: Average Pressure Profile Comparison, Mixed-Wet System, with and without TSF 61

Figure 5.1.3. 1: Average Pressure Profile Comparison, with and without capillary pressure 63

Figure 5.1.3. 2: Dimensionless Pressure Comparison, with and without capillary pressure 63

Figure 5.1.4. 1: Dimensionless Pressure Comparison, Low and High Water Saturation . 65

Figure 5.1.5. 1: Bitumen Fractured Reservoir (Pilot application, Husky Energy, 2013) . 67

Figure 5.1.5. 2: Karst Breccia with Cretaceous Siliciclastic Fill (Pilot application, Husky Energy, 2013).. 67

Figure 5.1.5. 3: Water Saturation and Permeability Profile (single-porosity) 68

Figure 5.1.5. 4: Water Saturation and Permeability Profile (dual-porosity)..................... 68

Figure 5.1.5. 5: Dimensionless Pressure Comparison, heterogeneous vs. homogenous .. 69

Figure 5.1.6. 1: Pressure profile, sensitivity of fracture-matrix permeability ratio (2D model) 70

Figure 6.1. 1: Discrete Fracture Network (DFN), (Pilot application, Husky Energy, 2013) 75

Figure 6.1. 2: Buffalo-Creek 10A Pilot Location (Alberta, Canada)..... 76

Figure 6.1. 3: Buffalo-Creek 10A Pilot Facility, (Norwest corporation report, 2008)..... 76

Figure 6.1. 4: Oil Rate, History Matched Results, (Pilot application, Husky Energy, 2013) 77

Figure 6.1. 5: Water Rate, History Matched Results, (Pilot application, Husky Energy, 2013) 77

Figure 6.1. 6: Bottom hole injection pressure, History Matched Results, (Pilot application, Husky Energy, 2013) 78

Figure 6.1. 7: Comparison History Matched Results with and without TSF models....... 78

NOMENCLATURE

A	area,m^2
β	formation volume factor, m^3/m
C	heat capacity, kJ/kg K
Cf	fracture compressibility
Cm	matrix compressibility
H	enthalpy, kJ/kg
K	permeability, md
K_r	relative permeability
K_h	thermal conductivity, $W/m.^oC$
L	fracture distance
m	fracture set
N	fracture set
P	pressure, kPa
q	flow rate, m^3/d
q_H	heat flow rate, kJ/d
q_L	heat loss, kJ/d
S	saturation
Tm	matrix temperature

Tf	fracture temperature
t	time, d
T_{cmf}	flow transmissibility, m^3/pa.s
T_{cmf}	thermal transmissibility, m^3/s
T*	temperature, ^oC
V_b	block volume, m^3
Z	height, m
ϕ	porosity
σ_F	flow shape factor, m^{-2}
σ_T	thermal shape factor,m^{-2}
υ	kinematic velocity,m^2/s
ρ	density, kg/m^3
λ	mobility ratio
η	thermal diffusivity
μ	viscosity, c.p
$\Im_{cmf}$	flow transfer function
$\Im^{\circ}_{cmf}$	thermal transfer function
γ	specific gravity
ω	correction factor
ω_p	weighting factor
Ψ	coefficient factor

SUBSCRIPTS

α	Phase
m	matrix
f	fracture
i	x-coordinate index
j	y-coordinate index
k	z-coordinate index
F	flow
T	thermal
r	rock
P_D	dimensionless pressure
n	last time-step taken
n+1	time-step under consideration
x,y,z	direction

CHAPTER 1

Introduction

There are a significant number of naturally fractured reservoirs (NFR) in the world that contain heavy and extra heavy oil, which are important resources. In fact, fractured petroleum reservoirs represent over 20% of the world's oil and gas reserves (Saidi, 1983). For instance, the Upper Devonian Grosmont, which is composed of limestone and dolomite formations in the north–central regions of Alberta, Canada, contains close to 64 billion m^3 (406 billion barrels) of bitumen in place. This is one of the largest carbonate bitumen deposits in the world and is characterized as a heavily fractured formation with relatively high matrix porosity and permeability.

Due to the multi-scaled character of fracture networks and their high degree of heterogeneity, the characterization of fractured reservoirs is extremely complex. It is unquestionable that modeling and simulating these types of reservoirs are different from conventional modeling in that they require many different techniques to overcome the severe complexities of multiphase flow dynamics. Therefore, single-medium numerical simulators cannot capture the two-scale heterogeneity and two-flow regimes behavior. Fractures can alter the porosity and/or permeability of a reservoir (Nelson, 2001), resulting in a more complex fluid flow compared to non-fractured reservoirs. The concept of treating NFRs as a dual-porosity system was introduced by Barenblatt et al. in 1960. The dual-porosity approach has been the most popular and effective technique to model naturally fractured reservoirs where the fracture and matrix systems are separated into two different continua, each with its own set of properties. A classical dual-porosity model utilizes transfer functions to model the matrix-fracture interactions. Two flow systems are assumed to be superimposed and are connected by a transfer function (TF) to represent the fluid exchange term between the fractures and matrix blocks. The injected fluid (water or gas) can move much faster in the fractures than in the matrix blocks, which causes the transfer of fluids to take place between the fractures and matrix. Therefore, the matter of fluid velocities is very important in naturally fractured reservoirs (Ramires et al., 2007). There have been a number of attempts over the last 50 years to develop methods for a better understanding of impacts of the fracture systems on oil recovery. Most of the naturally fractured reservoirs exhibit two flow regimes: the flowing or fractured region, which carries most of the flow, and stagnant matrix blocks, where most of the hydrocarbons are stored. Simulation of the naturally fractured reservoirs is a challenging task; in particular, representation of the correct behavior of the recovery mechanisms in a flow model is extremely difficult. The dual-medium modeling of naturally fractured reservoirs is a possible solution by idealizing a reservoir as a set of disconnected matrix blocks within a highly conducted fracture network (Abushaikha et al., 2008). Consequently, in order to understand the optimal recovery mechanisms, numerical simulation models require an adequate transfer function between the fracture networks and matrix blocks, which models via shape factor.

Comprehension of the correct behavior of the recovery mechanisms in naturally fractured reservoirs in flow simulators is a challenging task, especially for heavy and extra heavy oil reservoirs. There is a big gap and lack of knowledge of exclusive and an adequate shape factor value, which is critical for accurate modeling of a dual-medium system particularly for thermal recovery. The concept of a shape factor for modeling NFRs was originally introduced by Barenblatt in 1960 (van Heel, 2006). Since then, different formulations have been proposed by researchers mostly to improve the understanding for conventional oil recovery mechanisms (Warren and Root 1960, Kazemi and Gilman 1983, Lim and Aziz 1994, Coast 1989, Thomas 1983, and others). A matrix-fracture transfer shape factor is one of the most important parameters in the modeling of these reservoirs. A shape factor considers the fracture-matrix geometry and is defined as a cross-sectional area for fluid transfer per unit volume divided by fracture spacing. Though this parameter is still not very well understood, it can be described as a geometrical factor that partially governs the flow behavior between the fractures and the matrix blocks. Most of the existing derivations of the transfer shape factor are based on the assumption of a pseudo-steady state condition, which is the main limitation of these formations. It is important to note that when a fractured reservoir is exposed to relatively significant pressure or temperature differences, the pseudo-steady state concept gives a large error compared to transient formulation of fluid exchange between the matrix and fractures. Nevertheless, an appropriate transfer shape factor is required for accurate modeling of thermal recovery processes in naturally fractured reservoirs. Thus the fracture-matrix fluid interaction is still an active area of research.

This research study is divided into four main sections. The first section focuses on the rate of exchange between the fractures and the matrix when the matrix blocks (fine-grid) are surrounded by fractures in a water-oil (bitumen) system, based on a single-porosity simulation model for a thermal recovery method. This result provides a possibility to benchmark the most common models such as the Warren and Root 1960, Kazemi (1984) and Gilman (2003), Lim and Aziz (1994), Coats 1989, and Thomas et al. (1983) models.

In the second section, the concept of a transient shape factor (TSF) is introduced, which leads to the derivation of a new Thermal Transfer Function (T.T. F) for non-isothermal recovery processes. The results are verified with a fine-grid single-porosity model and compared with the most current shape factor formulations.

The third section focuses on sensitivity studies to identify some of the important factors that have the largest impact on the deviation of the dual-medium model with the use of Kazemi's shape factor formulation from a reference solution.

Finally, the results of the sensitivity studies above help to better describe the concept of a transient shape factor (TSF), which is validated against historical field data to confirm the capability of the proposed model for reproducing more accurate reservoir fluid dynamic processes and improved modeling of the oil recovery mechanisms.

The thesis includes seven Chapters. Chapter 1 is a quick overview on modeling naturally fractured reservoirs. Background and an extensive literature review are presented in Chapter 2. Chapter 3 discusses the statement of the problem and motivation for this work. The concept of the new Transient Shape Factor (TSF) is introduced in Chapter 4. In this chapter, a new derivation of a time-dependent shape factor for thermal reservoir simulation is introduced and a MATLAB code is developed and coupled with the CMG STARS reservoir simulator to compute the transient shape factor (TSF). Chapter 5 is mainly focused on some of the important factors that have the largest impact on the deviation from a dual-medium model with the use of Kazemi's shape factor formulation. In this section, sensitivity studies for the main parameters are performed and the effect of each parameter is compared and validated against a reference solution. Chapter 6 covers the model validation, where historical field data is used to validate the proposed transient shape factor concept. Chapter 7 highlights a summary of the key outcomes of the thesis and conclusions accomplished throughout this study and offers a discussion for potential future research work for the development of thermal reservoir simulation in naturally fractured reservoirs.

CHAPTER 2

2.1. Background and Literature Review

In this chapter an historical view of various shape factor formulations for simulating a dual-porosity system is presented. The dual-porosity approach is one of the most efficient methods to model fluid flow in fractured reservoirs, where the matrix and fractures are separated into two different media, each with its own properties. The concept of a dual-porosity system was introduced by Barenblatt et al. in 1960, which is widely used to model the behavior of naturally fractured oil and gas reservoirs (Wuthichmarn, 2011). The complexity of modeling NFRs is mostly attributed to the highly heterogeneous and anisotropic nature of the fracture systems. Due to the multi-scaled character of fracture networks and their high degree of heterogeneity, characterization and modeling of fractured reservoirs is a very challenging task. Most of naturally fractured reservoirs exhibit two flow regimes, the flowing or fractured region, which carries most of the flow, and stationary matrix blocks, where most of the hydrocarbons are stored. These two flow systems are assumed to be superimposed and are connected by a transfer function (TF) to represent the exchange term between the fractures and the matrix, which is dependent on a shape factor. In all fractured reservoirs, the main concern is the rate of matrix-fracture fluid transfer which is directly related to the fracture intensity of the open connected fractures (Gilman et al., 2011). In order to understand the optimal recovery mechanism, numerical simulation models require an adequate transfer function between the fractures and matrix, which is modeled via a shape factor.

During the past 40 years several formulations have been proposed for the matrix-fracture transfer function in dual-porosity flow simulation. An extensive literature review is conducted to establish a detailed understanding of the existing state of knowledge of the matrix-fracture transfer function (TF) over the past few decades. Nonetheless, it is fair to say that the current studies of the dual-medium formulations have been mainly based on the conventional oil recovery methods. Therefore, this shape factor function is not yet fully understood for thermal recovery mechanisms, especially for fluids such as bitumen with extreme thermal viscosity fluctuation. As a result, there is a significant unknown in modeling of fractured reservoirs in thermal flow simulation. Hence any projection model would be extremely affected by the accuracy of the matrix-fracture transfer function (shape factor).

2.2. Transfer Function (TF)

It is unquestionable that modeling and simulating of fractured reservoirs is different from conventional modeling and is a very complicated and challenging task. This is primarily due to the multi-scaled character of fracture networks, their high degree of heterogeneity, and complex interaction of the various mechanisms that govern mass and energy transfer. The standard single-medium numerical simulators cannot fully capture the nature of the two scale heterogeneity (Abushaikha, 2008). The concept of primary and secondary porosities (dual-porosity) was formulated initially by Barenblatt et al. (1960), where the primary porosity region (matrix) contributes significantly to the pore volume but yields negligible contribution to the flow capacity (Warren and Root, 1962). Therefore, an idealized model was developed to characterize and study the fluid interaction between the matrix and fractures, where the matrix blocks were assumed to be isotropic and homogeneous, while the fracture networks were hypothetically represented by identically rectangular spaces with no direct communication between one another through the matrix blocks. Figure 2.2.1 shows the idealization of a NFR system (Warren and Root, 1963).

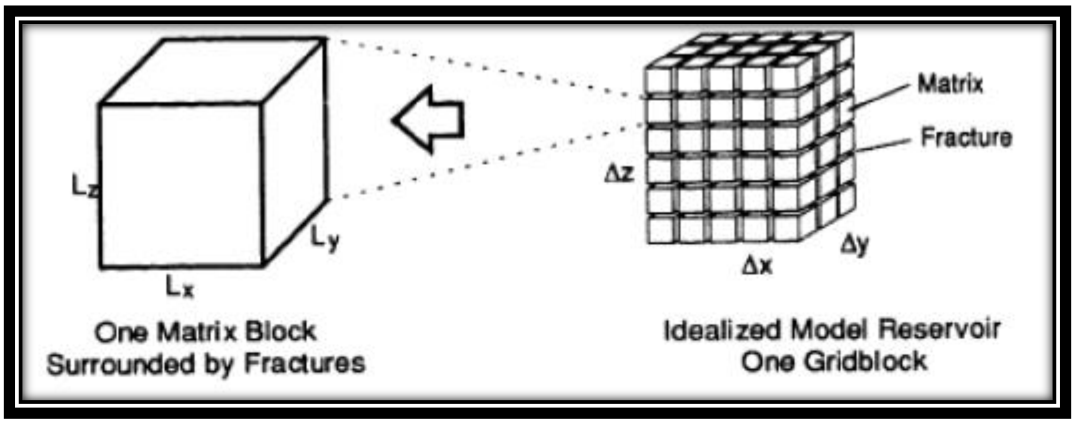

Figure 2.2. 1: Idealization of an NFR system (Warren and Root)

Warren and Root (1963) introduced the theory of the transfer function using single phase flow and a pseudo-steady state assumption (i.e., a single saturation/pressure value in the matrix) in the context of well testing for a dual-porosity system, on which most of the current developments pertaining to transfer functions are based. Generally, a transfer function should account for imbibition, gravity drainage, fluid expansion, and molecular diffusion, all of which are important, depending on the main recovery mechanisms. A mathematical model was developed based on the continuity equation for a 2D slightly compressible fluid in a fracture (Equation 2.2.1). The first part of the equation governs the fluid flow in the fracture system, and the second part represents the fluid flow in the matrix system for a dual-porosity system:

$$1/\mu \left(k_{fx}\frac{\partial^2 P_f}{\partial x^2} + k_{fy}\frac{\partial^2 P_f}{\partial y^2}\right) - \emptyset_m C_m \frac{\partial P_m}{\partial t} = \emptyset_f C_f \frac{\partial P_f}{\partial t} \qquad (2.2.1)$$

The fluid transfer per unit bulk volume between the matrix and the fracture system is proportional to the difference between the fracture pressure and the average matrix block pressure, which is termed the transfer function (Warren and Root, 1963):

$$T = \sigma \frac{K}{\mu}(P^m - P^f) \qquad (2.2.2)$$

where T is the transfer function (1/sec), σ is defined as a shape factor ($1/m^2$), K is the matrix permeability (md), μ is the fluid viscosity (c.p), P is the pressure (atm), and subscripts f and m are for the fracture and matrix denominations, respectively. The matrix-fracture transfer shape factor (σ) has the dimension of a reciprocal area, L^{-2}. The shape factor is defined as a cross-sectional area for fluid transfer per unit volume divided by fracture spacing. The following expression is for the shape factor obtained by Warren and Root (1963), which reflects the geometry of the matrix sections and controls flow between the fracture and matrix media:

$$\sigma = \frac{4N(N+2)}{L^2} \qquad (2.2.3)$$

where N is the number of the set of fractures (1D, 2D or 3D). For cubic matrix blocks having a fracture spacing of L, then σ has the value of $12/L^2$, $32/L^2$, and $60/L^2$ for one, two, and three dimensions of fractures, respectively.

In another work, the application of a shape factor in dual-medium numerical simulation was presented by Kazemi et al. over 40 years ago (1976), and has been used in the most commercial simulators since then. Kazemi et al. (1979) have idealized a reservoir with a set of uniformly spaced horizontal layers and a set of fractures acting as spacers (Figure 2.2.2). They extended the Warren and Root formulation to a multiphase flow system by utilizing the phase relative permeability in the matrix-fracture interaction term and solving a dual-porosity model in three dimensions numerically, assuming that the fractures form a continuum, while the matrix blocks are non-continuous (fractures are the boundaries of the matrix blocks). Gilman and Kazemi (1983-1986) continued improving their model by adding a gravity term and making it saturation dependent (Abushaikha, 2008).

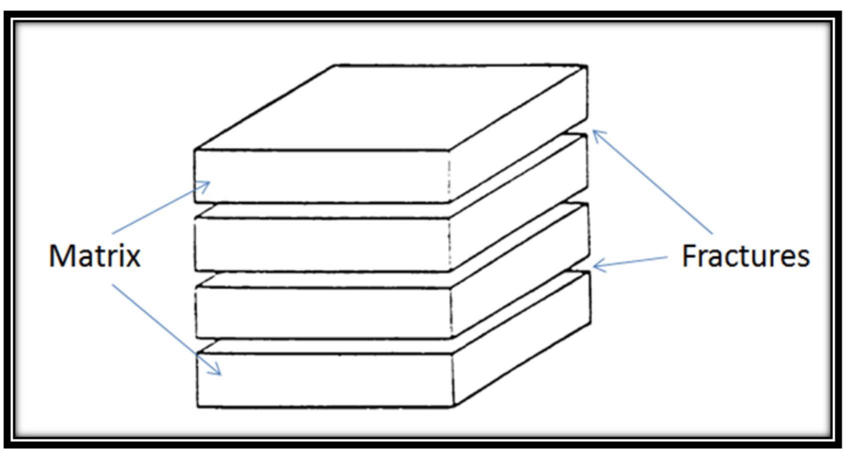

Figure 2.2. 2: Idealization of an NFR system (Kazemi)

The finite difference equations of flow equations for the dual-porosity systems are given as follows (Kazemi et al., 1976):

For fractures:

$$\left\{\sum_l T_l \lambda_p \rho_p \Delta\left(p_p - \gamma_p D\right) + \rho_p q_p^w\right\}_f - \tau_{mf} = \left\{\frac{V}{\Delta t}\Delta_t\left(\phi S_p \rho_p\right)\right\}_f \tag{2.2.4}$$

For matrix:

$$\tau_{mf} = \left\{\frac{V}{\Delta t}\Delta_t\left(\phi S_p \rho_p\right)\right\}_m \tag{2.2.5}$$

where the transfer function is defined as

$$\tau_{mf} = k_m V \sigma \lambda_p \rho_p \left[\left(p_p - \gamma_p D\right)_f - \left(p_p - \gamma_p D\right)_m\right] \tag{2.2.6}$$

A further modification of the original Kazemi's transfer function was presented by Gilman and Kazemi (1983) to include fracture relative permeability, where fluid was flowing from the fractures to the matrix. The ω_p is a weighting factor which varies from 0 to 1 and is equal to 1, if flow is from the matrix to the fractures (Sarma, 2003).

$$\tau_{mf} = k_m V \sigma \left\{\omega_p \lambda_{pm} + \left(1 - \omega_p\right)\lambda_{pf}\right\} \rho_p \left[\left(p_p - \gamma_p D\right)_f - \left(p_p - \gamma_p D\right)_m\right] \tag{2.2.7}$$

Kazemi et al. (1976) developed a numerical algorithm (using finite differences) to solve the fracture flow equations and derive the matrix-fracture fluid transfer. They proposed their shape factor formulation based on a direct material balance on a cubic matrix block under the assumptions of a pseudo-steady state condition in terms of fracture spacing with a coefficient of $\alpha = 4$ for a system with uniform fracture spacing of L_k:

$$\sigma = \alpha \sum_{k=1}^{nd} {}^{1}/_{L_k^2} \qquad (2.2.8)$$

For an isotropic rectangular matrix block, the shape factor was described as

$$\sigma = 4\left[1/L_{mx}^2 + 1/L_{my}^2 + 1/L_{mz}^2\right] \qquad (2.2.9)$$

where L_{mx}, L_{my}, and L_{mz} are the fracture lengths or fracture spacing in the x, y, and z directions, respectively. For equal fracture spacing of L in all directions, a value of the shape factor of 12/L2, 8/L2, and 4/L2 are allocated for one (3D), two (2D), and three (1D) sets of fractures, respectively. Note that their derivation is based on the assumption of a linear pressure gradient between a central matrix and the fracture (Lim et al., 1994).

Kazemi et al. (1992) and Zhang et al. (1996) also defined a generalized shape factor based on laboratory water imbibition experiments. This equation is also the most general form for up-scaling from complex fracture patterns such as discrete fracture network models for both single and multi-phase flows.

$$\sigma = {}^{1}/_{V} \sum_{i=1}^{I} {}^{Ai}/_{di} \qquad (2.2.10)$$

In another study, an analytical expression for a constant shape factor for the flow between the matrix and fractures was introduced by Lim et al. (1994). The matrix-fracture transfer shape factor formulation in dual-porosity was derived without making the pseudo-steady state assumption by combining analytical solutions of pressure diffusion for various flow geometries. This expression was based on the assumption of anisotropic matrix blocks of a linear pressure gradient between the center of the matrix and the fractures with a coefficient of π rather than 4:

$$\sigma = \frac{\pi^2}{\bar{k}} \left(\frac{k_x}{L_x^2} + \frac{k_y}{L_y^2} + \frac{k_z}{L_z^2}\right) \qquad (2.2.11)$$

where

$$\bar{k} = \sqrt[3]{kx.ky.kz} \qquad (2.2.12)$$

For an isotropic rectangular matrix block, the shape factor was described as

$$\sigma = \pi^2 \sum_{k=1}^{nd} {}^{1}/_{L_k^2} \qquad (2.2.13)$$

For a system with uniform fracture spacing of L, Lim and Aziz, 1994 computed the shape factors for two (2D) and three (3D) fracture sets as $2\pi^2/L^2$ and $3\pi^2/L^2$, respectively. From their study, Lim and Aziz concluded that the Warren and Root shape factor tends to overestimate the rate of recovery and the Kazemi et al. formulation tends to underestimate the recovery rate considerably.

In another work, Lim et al. (1992) investigated the impact of the fluid movement on matrix-fracture transfer computation in geothermal reservoirs, where the phase changes occurred as a result of vaporization and condensation of water (Lim et al., 1992). In this study they indicated that a dual-porosity model could not reproduce the same results from the fine-grid model regardless of the shape factor used. They concluded that mass transfer calculated using any constant shape factor in a dual-porosity model may be inconsistent with the observed transport mechanisms in a vapor dominated geothermal system. Also, they stressed that the shape factor is actually not a constant, but a function of time (Lim and Aziz, 1995).

The limitation of the pseudo-steady state shape factors was also investigated by Sarma (2007) for a single phase flow in fractured reservoirs using numerical techniques. The results from this study indicated that none of the dual porosity formulations are in good agreement with a reference solution, especially during early time. From this study, it is clearly concluded that a constant value used for a shape factor cannot be used successfully to obtain the desired match for all times with the reference solution (Eclipse discrete curve) due to the existence of a transient state. Figure 2.2.3 shows the comparison of the average pressure within the matrix block with time for the most common shape factor formulations.

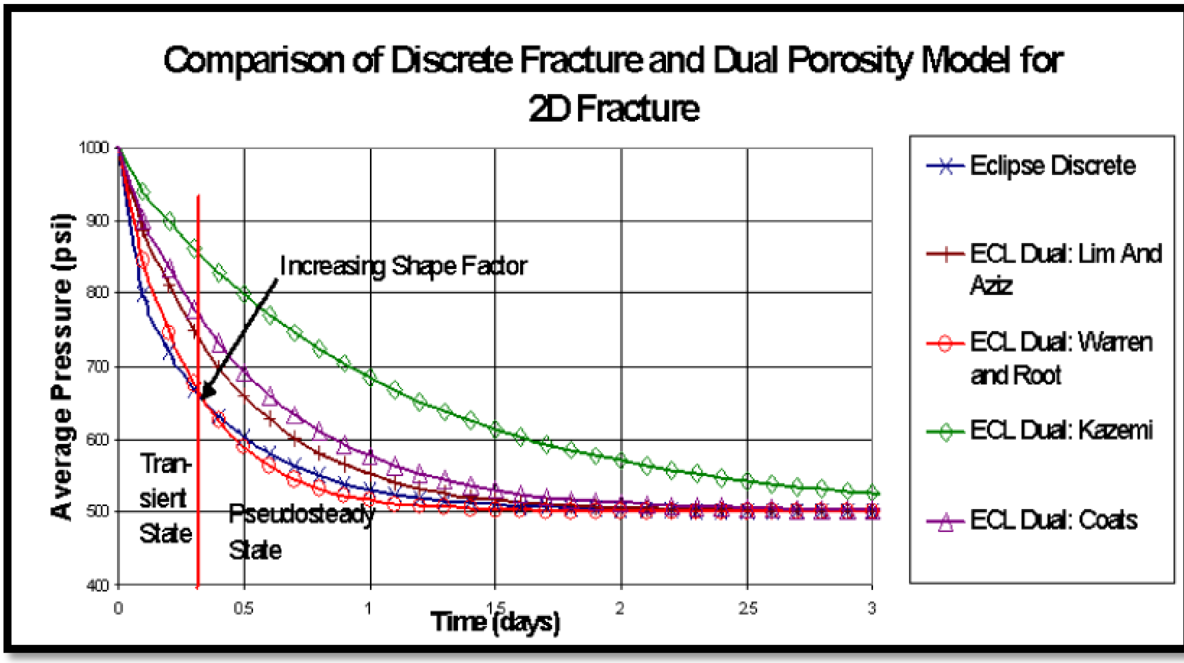

Figure 2.2. 3: Comparison of the average pressure (Pallav, 2003)

The work of Chang (1993) deals with a time-dependent shape factor for flow between the matrix and fracture systems for 1D flow in a relatively short transient period. Chang avoided the pseudo-steady state assumption combining the geometrical aspect of the system with analytical solutions of a pressure diffusion equation, which is not really applicable when it is utilized to model any field simulation cases. In addition, the given analytical solution is too complicated to be incorporated directly into current commercial simulators.

$$\sigma = \frac{\pi^2}{L_x^2} \frac{\sum_{m=0}^{\infty} \exp\left[-(2m+1)^2 \pi \quad t_D\right]}{\sum_{m=0}^{\infty} \frac{1}{(2m+1)^2} \exp\left[-(2m+1)^2 \pi \quad t_D\right]} \tag{2.2.14}$$

A new set of transfer functions for modeling dual-porosity systems were introduced by Sarma and Aziz in 2006. In this work, it was attempted to eliminate the main existing limitations (pseudo-steady state flow and orthogonal fracture systems) by combining the single-phase transfer function with the analytical solution of the pressure diffusion equation. Subsequently, an analytical form of a time-dependent shape factor was derived along with the differential form of the transfer function of two-phase flow for more accurate modeling of fractured reservoirs. The new transfer function accurately takes into account multiphase flow matrix-fracture interaction. Sarma and Aziz (2006) used the ECLIPSE 100 black oil simulator to validate the proposed analytical model using a single-porosity model. Figure 2.2.4 shows a comparison of the numerical and analytical methods for the average matrix drawdown pressure. These results were verified against reference solution (fine-grid) single-porosity models with very good agreement for the transient period of the system; however, after reaching the pseudo-steady state, this analytical solution is no longer valid so a constant shape factor has to be used.

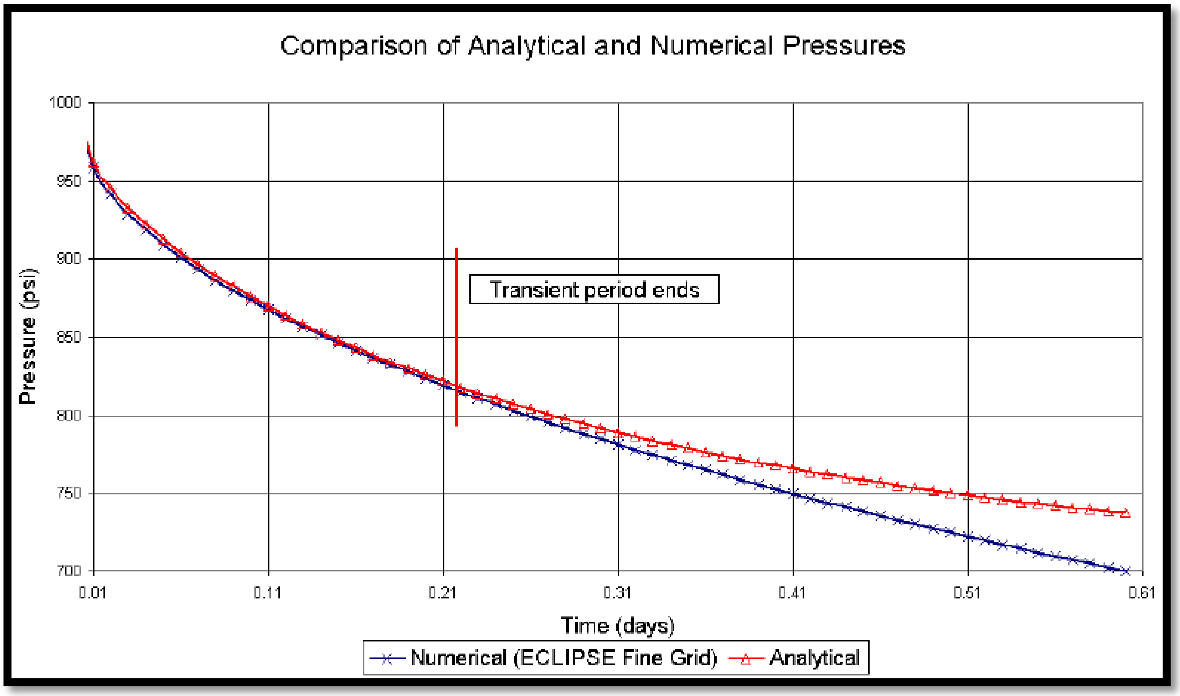

Figure 2.2. 4: Pressure comparison of analytical vs. numerical methods

Different shape factor formulations have been proposed in the literature, which can lead to totally different reservoir behavior; therefore, a suitable shape factor value can be a critical factor for accurate dual-medium modeling. The shape factor concept was also investigated by van Heel et al. (2006) by considering the nature of the main physical recovery mechanisms, namely convection and diffusion processes. Fundamentally, the pseudo-steady state assumption is not applicable for a case where there is a rapid pressure or temperature change in the matrix/fractures. Van Heel et al. (2006) introduced the concept of a thermal shape factor for the heat transfer between fractures and the matrix in a dual-permeability context and compared their results against fine-grid single-porosity simulation. They were mostly interested in transfer of energy from the fractures to the matrix blocks by conduction (temperature of the matrix blocks dominated by a transient diffusion equation), which strictly emphasized that the shape factor for transient diffusion should be a time dependent quantity in reservoir simulation. The following analytical expression was obtained by van Heel et al. for a thermal shape factor (η_{th} is thermal diffusivity):

$$\sigma_{th}(t) = -\frac{d\overline{T}_m}{dt} \Big/ \eta_{th}(\overline{T}_m - T_f) \tag{2.2.15}$$

They considered one- and two-dimensional heating of matrix blocks from initial temperature of 50 °C up to 250 °C by heat conduction from the surrounded fractures (sides) to mimic a steam injection process. Figures 2.2.5 and 2.2.6 show the comparison results of 1D and 2D blocks with their analytical solution with the most common shape factors (Costs 1989, Chang 1993, Lim-Aziz 1995, Kazemi 1976). The pseudo-steady state diffusion shape factor derived by Coast, 1989 provides a good approximation; even it crosses the analytical curve and subsequently overestimates the exact temperature. This evaluation clearly demonstrates that Kazemi's shape factor is not appropriate for modeling a diffusion governed process such as thermal recovery mechanisms.

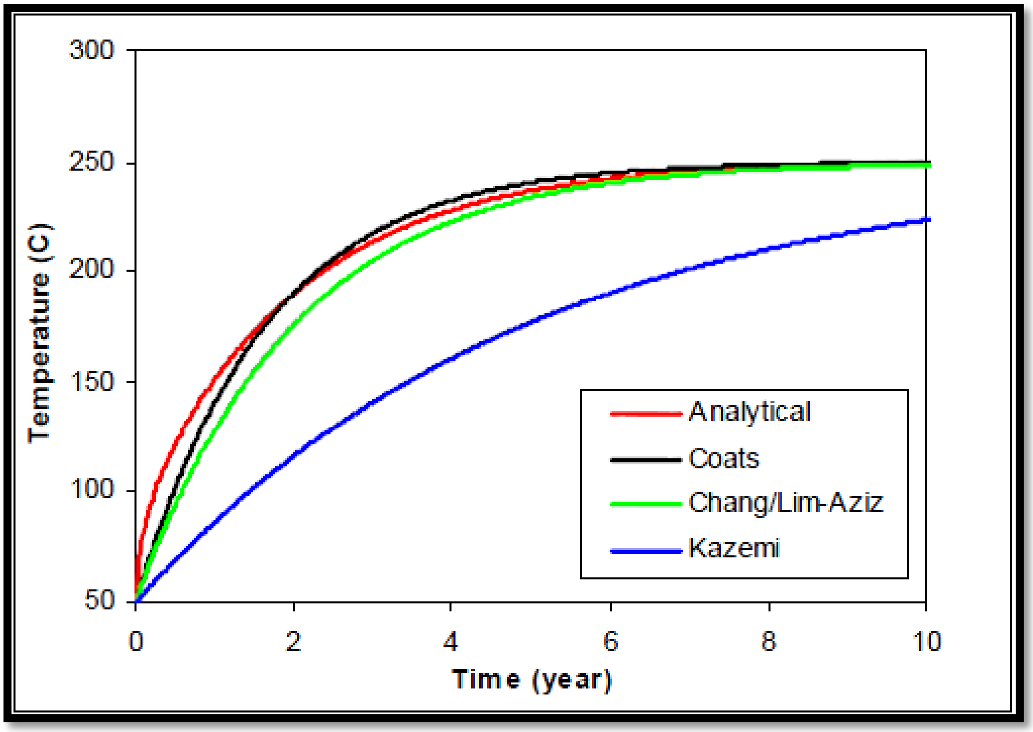

Figure 2.2. 5: Temperature profile comparison, 1D case (van Heel et al., 2006)

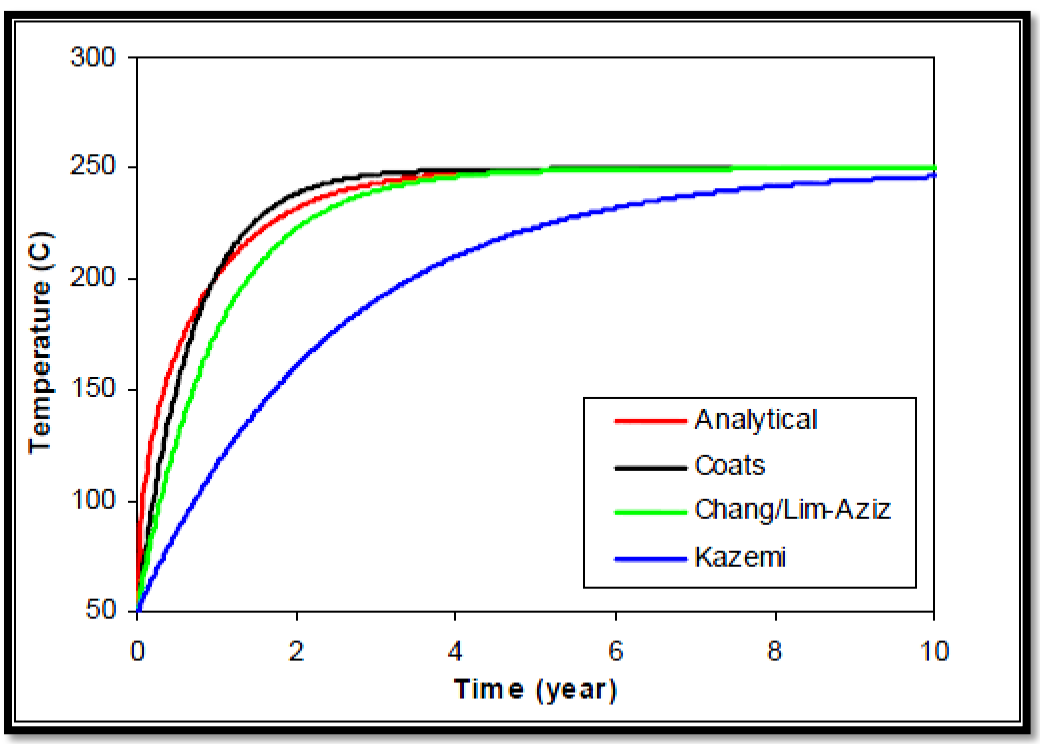

Figure 2.2.6: Temperature profile comparison, 2D case (van Heel et al., 2006)

In order to validate the proposed analytical model, van Heel et al. (2006) conducted a series of dual-permeability numerical simulations using Shell's reservoir simulator (MoReS) to simulate steam enhanced gravity drainage in a fractured reservoir. Single-porosity, fine-grid models were used as a reference solution to compare and validate the dual-permeability models utilizing different shape factor formulations. Four sets of simulation results were compared against the reference solution. Figures 2.2.7 and 2.2.8 show the comparison of the oil rate and temperature profile, respectively. These results also confirm that convection type shape factors such as Kazemi's formulation (constant values) are not suitable for modeling diffusion processes.

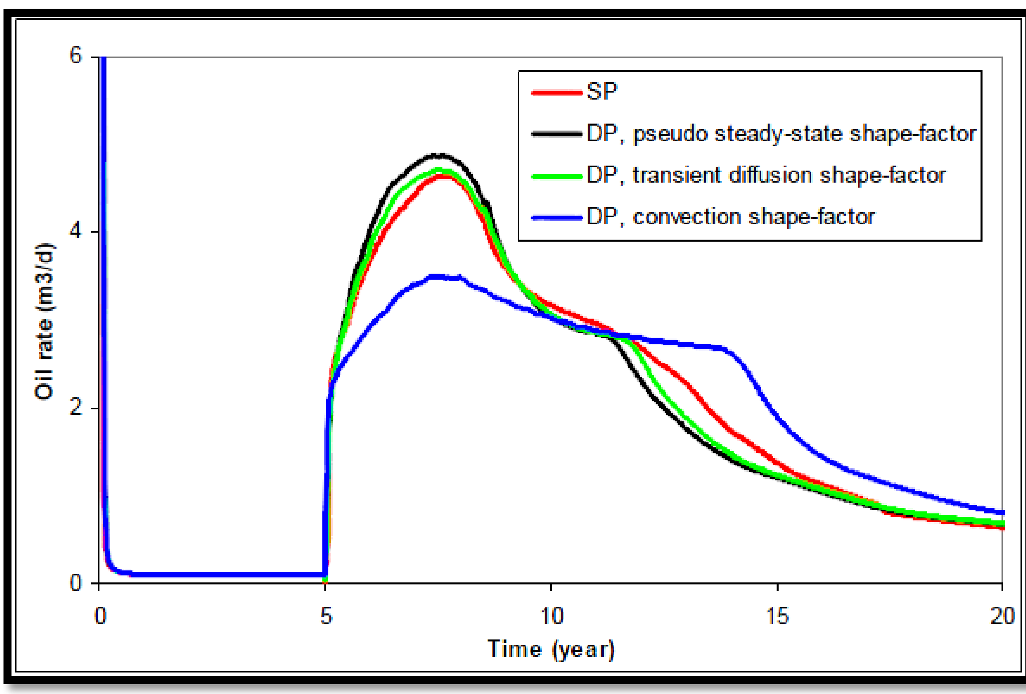

Figure 2.2. 7: Oil rate profile comparison, 1D case (van Heel et al., 2006)

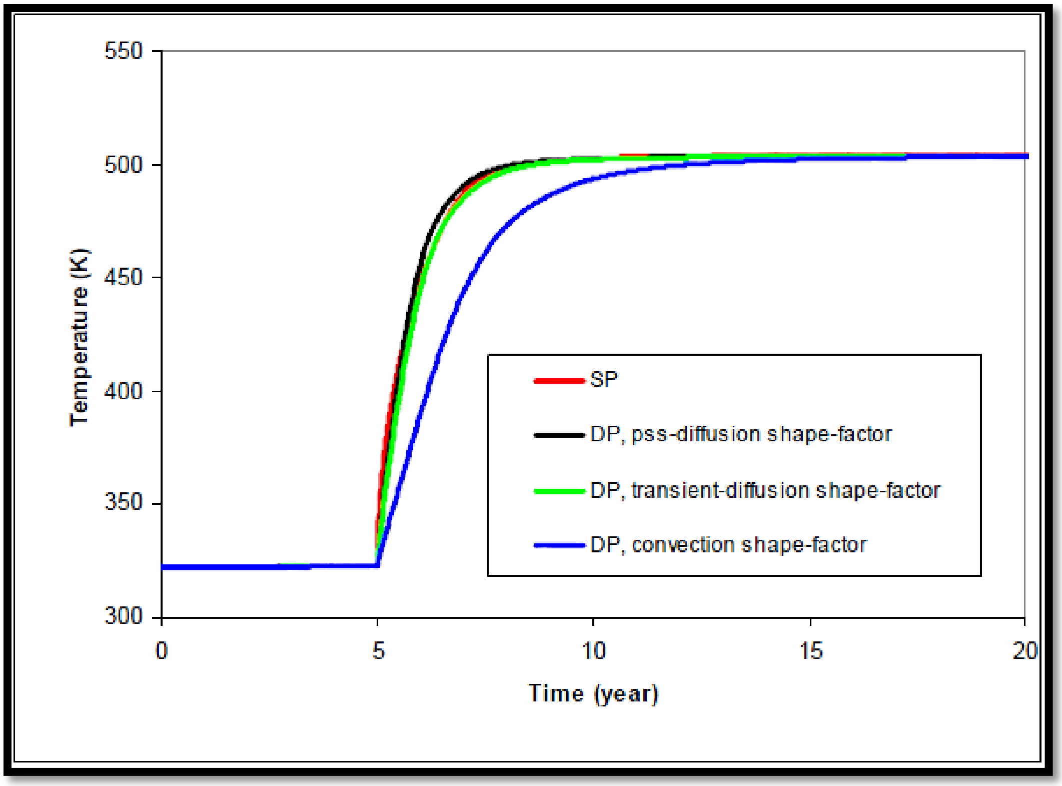

Figure 2.2. 8: Temperature profile comparison, 1D case (van Heel et al., 2006)

In different work, van Heel et al. (2008) derived an analytical expression to model heat and fluid transfer for a single-phase matrix-fracture shape factor in a dual-porosity formulation that retains transient in pressure and temperature diffusion processes by considering gas-oil gravity drainage in a steam injection process. In a heavy oil recovery process, the dominant recovery methods are thermal expansion and thermal viscosity reduction; therefore, the accuracy of thermal simulation results essentially depends on the accuracy of the modeling of the temperature evolution in the matrix blocks. In this paper, van Heel et al. (2008) mainly focused on thermal recovery and presented the results for a 1D case for short and long time approximations of a shape factor and later presented an analytical transient thermal shape-factor for diffusion in 2D matrix blocks in a single-phase system, which are time-dependent shape factors for a pressure transient effect for short-time, medium-time, and long-time regimes. They validated their model against a fine-grid single porosity model (reference solution) with relatively good agreement and illustrated that constant shape factors (fixed values) could not capture the initial fast heating regime of the matrix blocks. Figure 2.2.9 shows the temperature profiles from the van Heel et al. model compared with other common shape factor formulations.

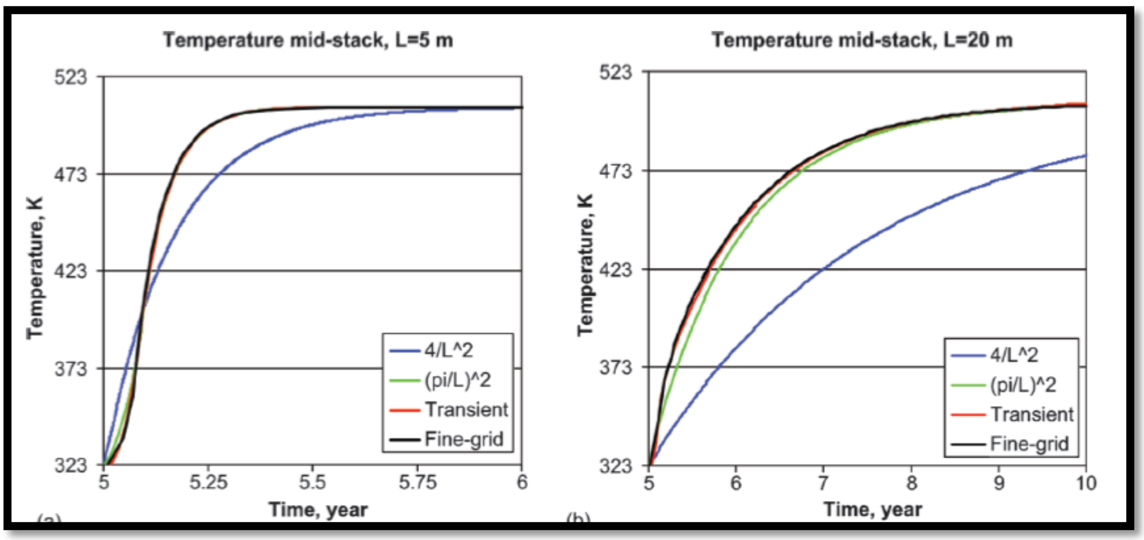

Figure 2.2. 9: Temperature profile comparison, different shape factor formulation (van Heel et al., 2008)

Researchers have considered a transfer function (shape factor) from the well testing literature and attempted to improve it for a better understanding of the reservoir performance. Representing the correct behavior of recovery mechanisms in naturally fracture reservoirs utilizing a dual-media system via shape factor formulations has been also investigated by other researchers. Coats (1989) extended a dual-porosity model to compositional simulation, proposed a different expression for a shape factor, stated that

Kazemi's shape factor was too low, and derived a new formulation by multiplying it by a factor of 2. Thomas (1983) pointed out that the rate of hydrocarbon recovery from a fractured reservoir is a function of the matrix size and properties as well as pressure and saturation history of the fracture system. Thomas et al., 1983 developed a three-dimensional, three-phase model to simulate the flow in naturally fractured reservoirs using a dual-porosity system, which suggested that Kazemi's shape factor should be increased by a factor of 2. Quandalle and Sabathier (1998) corrected the Kazemi and Gilman transfer shape factor formulation by separating the horizontal and vertical flows as well as adding a viscous recovery term to the model. Chen (1995) considered a model in early and late times in matrix blocks with a new formulation. Ishimoto (1988) used a sub-domain technique to solve the pseudo-steady state assumption and allowed for the creation of a saturation front. Another approach to modeling fractured reservoirs was developed by Sonler et al. (1988), which primarily focused on more complex problems and expanded the previous model from 2D to 3D full-field reservoir models to express the fluid exchange between the matrix and fractures. Litvak (1985) described another 3D three-phase fractured reservoir simulator that could be used in full-field model studies, with a similar approach that Kazemi and Gilman, 1988 used for the matrix-fracture exchange term by adding the gravity term into the equations. Lu et al. (2007) followed up the work by Chen and proposed a new original transfer function called a General Transfer Function (GTF) for a mixed-wet system, which introduced a method of capturing the dynamics of expansion, diffusion, and displacement accurately for better predictions than the previous models. Ueda et al. (1989) argued that Kazemi's shape factor should be increased by a factor of 2 to 3. Edgar et al. (2003) introduced a new time-dependent matrix-fracture transfer shape factor based on a dimensional analysis of the experimental studies by Rangel-German (2003), which is dependent on the transient behavior of the water saturation between fractures and matrix blocks.

In review of all the previous work it appears that there is a discrepancy and lack of agreement regarding which shape factor should be used in simulation of dual-media systems. Table 2.2.1 shows a summary of shape factors obtained by some of the researchers referred to above. Although a few researchers acknowledged the time dependence of a shape factor for transient flow between the matrix and fracture systems, an expression for the shape factor particularly for simulating thermal recovery processes in fractured reservoirs has not yet been developed.

Table 2.2. 1: Constant shape factors by various researchers

Researcher (s)	Phase	Set of fracture		
		1	2	3
Warren & Root, 1960	Single Phase	12	32	60
Kazemi et al., 1976	Single Phase	4	8	12
Thomas et al., 1983	Two-Phase	…	…	25
Coats, 1999	Two-Phase	8	16	24
Gilman & Kazemi, 1993	Single Phase	…	…	29.61
Lim & Aziz, 1994	Single Phase	π^2	$2\pi^2$	$3\pi^2$
Quintard et al., 1996	Single Phase	12	28.4	49.6
Noetinger et al., 2000	Single Phase	11.5	27.1	…
Penuela et al., 2002	Two-Phase	9.87	…	…
Sarda et al., 2002	Single Phase	8	24	48

2.3. Concluding Remarks

- There is an insufficient understanding of an adequate matrix-fracture transfer shape factor formulation in a dual-porosity system for thermal recovery mechanisms.
- In review of all the previous work it appears lack of agreement regarding which shape factor should be used in simulation of dual-media systems.
- Most of the existing transfer function formulations (matrix-fracture interaction) are based on the assumptions of orthogonal fracture systems and pseudo-steady state flow, which are not representative of actual reservoir recovery mechanisms.
- Although researchers acknowledged the time dependence of a shape factor for transient flow between the matrix and fracture systems, an expression for the shape factor for simulating thermal recovery for a heavy or extra heavy oil system has not been yet developed.
- All the suggested analytical solutions are too complicated to be incorporated directly into current commercial simulators.

CHAPTER 3

Statement of the Problem

Naturally fractured reservoirs present many unique and complex challenges for modeling and simulation professionals due to many complex features that are present in this type of reservoirs at multiple scales. Due to the multi-scaled character of fracture networks and their high degree of heterogeneity, characterization and modeling of fractured reservoirs, especially those containing extra heavy oil, present many unique and complex challenges. The most important and challenging aspect of modeling a naturally fractured reservoir is the accurate estimation of the fluid exchange between the matrix and surrounding fractures. For many decades, the dual-porosity approach has been the most popular and effective technique in modeling of NFRs. Two flow systems are assumed to be superimposed and are connected by a transfer function (TF) to represent the exchange term between the flow domain (fractures) and the stagnant domain (matrix blocks).

There have been a number of attempts over the last 50 years to develop methods to improve modeling of the fluid exchange between the matrix and fractures via a shape factor. Recently, several formulations have been proposed by researchers mostly for conventional oil recovery methods in naturally fractured reservoirs. A correct understanding of recovery mechanisms in naturally fractured reservoirs with the use of flow simulators present a big challenge, especially for heavy oil and extra heavy oil recovery processes. All the previous studies for developing the dual-medium formulations (shape factor) for simulating naturally fractured reservoirs were based on conventional oil recovery methods, which indicated their inability to model multi-phase flow accurately for thermal recovery systems. The assumption of transient or pseudo-steady state conditions leads to different shape factor values; as a result, there is a significant deficiency in an understanding of an adequate matrix-fracture transfer shape factor in a dual-medium system principally for thermal recovery mechanisms. Furthermore, there is a significant deficiency in an understanding of an adequate matrix-fracture transfer shape factor in a dual-medium system for thermal recovery mechanisms. Although some of the researchers acknowledged that a shape factor is essentially a time-dependent quantity for transient flow between the matrix and fracture systems, an accurate expression for a transient shape factor has not yet been developed.

A suitable time-dependent matrix-fracture transfer function is required for an accurate flow behavior prediction in a dual-porosity system for the thermal recovery mechanisms in fractured reservoirs. The intention of this research, therefore, is to achieve a better understanding of the matter exchanges between fractures and matrix blocks in a dual-porosity system by defining an appropriate transient matrix-fracture shape factor formulation that accounts for a multiphase flow system in thermal reservoir simulation of naturally fractured reservoirs.

The thesis mainly aims to specifically achieve the following:

- Investigate the rate of exchange between fractures and matrix blocks in a water-oil (bitumen) system for a thermal recovery process. This enables the evaluation of the accuracy of the current shape factor formulation.
- Improve a dual-porosity model to simulate thermal recovery processes for a multiphase flow system by developing a robust transient shape factor (TSF) formulation.
- Develop a solution methodology to estimate the proposed transient shape factor for thermal flow simulation in fractured reservoirs.
- Identify some of the important factors that have an impact on the accuracy of modeling a dual-medium system.
- Validate the proposed model to make engineering predictions with confidence.

CHAPTER 4

4.1. Shape Factors

As mentioned above, a significant number of naturally fractured reservoirs (NFRs) discovered in the world contain heavy and extra heavy oil, which are important resources. Due to the multi-scaled character of fracture networks and their high degree of heterogeneity, characterization and modeling of carbonate fractured reservoirs, especially those containing extra heavy oil, present many unique and complex challenges. Moreover, single-medium numerical simulators cannot capture the two-scale heterogeneity and two-flow regimes behavior; therefore, dual-medium modeling of naturally fractured reservoirs is one of the possible solutions. The concept of a dual-porosity system was introduced by Barenblatt et al. in 1960, which is widely used to model the behavior of naturally fractured oil and gas reservoirs. The two-flow regimes are assumed to be superimposed and are connected by a transfer function (TF) to represent the exchange term between the fractures and the matrix, which involves a shape factor concept. The most important and challenging aspect of modeling a naturally fractured reservoir is the accurate estimation of the fluid exchange between matrix blocks and surrounded fractures.

Most of the existing transfer function formulations (matrix-fracture interactions) are based on the assumption of orthogonal fracture systems and pseudo-steady state flow, which are not representative of actual reservoir recovery mechanisms. A shape factor is one of the most important parameters for quantifying fracture-matrix geometry in modeling naturally fractured reservoirs. From the practical view, the shape factor is a second order, distance-related, geometric parameter that is used to calculate the mass transfer coefficient between matrix blocks and surrounded fractures (Gilman, 2003). Different shape factor formulations have been proposed by researchers to model a dual-porosity system. In general, analytical solutions to reservoir flow and energy equations are available based on a series of assumptions for geometry, properties and boundary conditions, which are not valid for complex reservoir problems. Therefore, numerical techniques have been applied to solve the mass and heat equations numerically.

As described earlier, a pseudo-steady state assumption is one of the major limitations of the existing shape factor derivations. This section focuses on the rate of exchange between fractures and matrix blocks in a water-oil (bitumen) system for a thermal recovery process. In order to validate the most common transfer shape factor formulations, 2D (single and multi-set matrix blocks) and 3D (multi-set matrix blocks) fine grid single-porosity models that are surrounded by fractures are constructed. In this study, a thermal and advanced processes reservoir simulator, CMG STARS from Computer Modeling Group Ltd is used. The results from the simulation provide a referenced solution for benchmarking the dual-porosity system with the application of the existing constant shape factor models, such as the Warren and Root 1960, Kazemi 1984 and Gilman 203, Lim and Aziz 1994, Coats 1989, and Thomas models. Subsequently, the concept of a transient shape factor (TSF) for modeling a dual-porosity system for thermal recovery processes is introduced and the results from the proposed model are compared and validated with the fine-grid single-porosity model. Finally, a MATLAB code is developed and coupled with the CMG STARS thermal reservoir simulator to compute the transient shape factor utilizing Kazemi's formulation (a TSF estimator tool kit).

4.1.1. 2D-Single and Multi-Set Matrix Blocks

Consider a fine grid single-porosity model with dimensions of a 1m x 1m x 1m cubic matrix block and the grid resolution of 1cm x 12.5cm x 100cm, surrounded by a set of fractures. A coarse dual-porosity model is defined with the same dimensions and grid resolution of 1m x 1m x 1m (Figure 4.1.1.1). Table 4.1.1.1 shows the matrix and fracture properties defined in this evaluation. This study tests two scenarios for the above described models: a single and multi-set of matrix blocks with the same grid resolutions. In the first case, the model was initialized by filling the fractures with the saturated steam at 4,000kPa, while maintaining a constant matrix pressure at 1,100kPa. Subsequently, the fracture pressure is suddenly dropped to 1,000kPa and maintains the system at this constant pressure for the entire test. Figure 4.1.1.2 demonstrates a reduction of the average system pressure with time. The red curve is obtained using a fine grid single-porosity model as a reference solution. The other curve (green) represents the dual-porosity model result using Kazemi and Gilman's constant shape factor. It is observed that the dual-porosity curves using Kazemi and Gilman's formulation do not match the reference solution.

Table 4.1.1. 1: Matrix and fracture parameters

Parameters	Matrix	Fracture
Porosity	25%	1%
Permeablity, md	50	10000
Aperture, mm	na	5
Pressure, kpa	1100	4000
Temperature, C	11	250
Sw	20%	20%
Dropped Fracture Press	...	1000

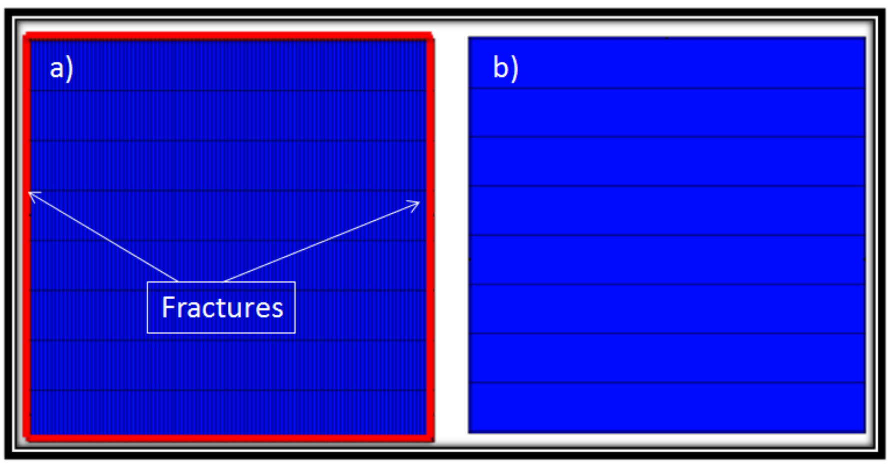

Figure 4.1.1. 1: 2D Single-matrix block models; a) single-porosity; b) dual-porosity

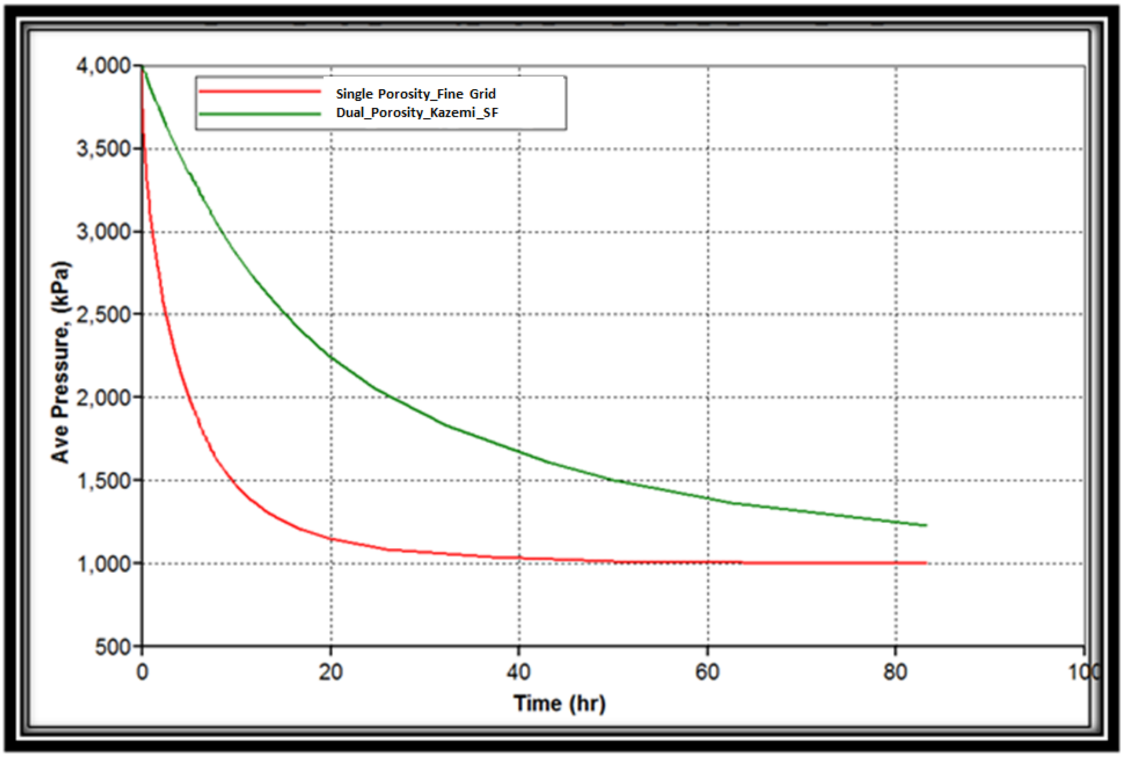

Figure 4.1.1. 2: Pressure drawdown comparison, single-porosity vs Kazemi's dual-porosity (2D model)

To understand the impact of a multiple grid blocks and various fracture sets, the previously described models are modified to 2m x 2m x 2m size dimensions with multi-fractures and matrix blocks (Figure 4.1.1.3). Ultimately, the models are initialized with the same process as previously described. Originally, the fractures are filled with saturated steam at 4,000kPa and the matrix block pressures are set at a constant pressure of 1,100kPa. Next, the fracture pressures are suddenly reduced to 1,000kPa and are maintained constant during the test. Figure 4.1.1.4 displays a comparison of the pressure drawdown of the fine grid, single-porosity model versus other common constant shape factors. The same observations are obtained; none of the dual-porosity models matches with the reference solution (red curve). The results indicate that the Warrant and Root shape factor model has a relatively good match with the reference solution during the early time, a better match can be achieved with the Lim and Aziz (1994) formulation in the mid time, and the Coats and Thomas constant shape factors are able to produce a good match during the late time only. However, overall results demonstrate that none of the above tested dual-porosity models can be used to generate a good solid match at all times with the reference solution; particularly, Kazemi's formulation (orange curve) shows the maximum deviation from the base solution. Moreover, the results show that the existing shape factors in a dual-porosity model are not suitable for predicting thermal recovery mechanisms of the naturally fractured reservoirs. Therefore, an appropriate transient transfer shape factor model must be developed between the fracture and the matrix for a better representation of a dual-porosity thermal system.

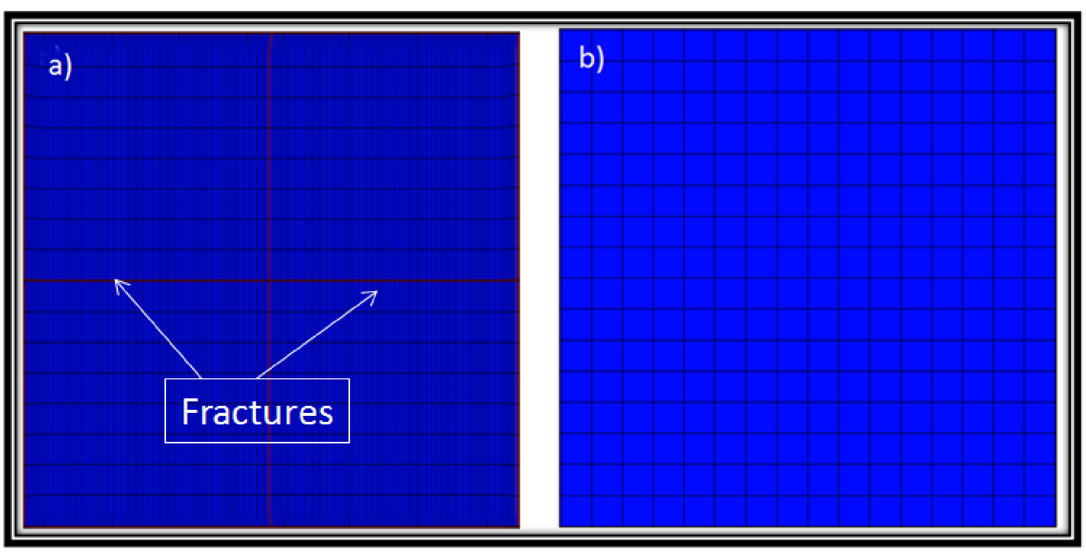

Figure 4.1.1. 3: 2D Multi-matrix blocks models: a) single-porosity; b) dual-porosity

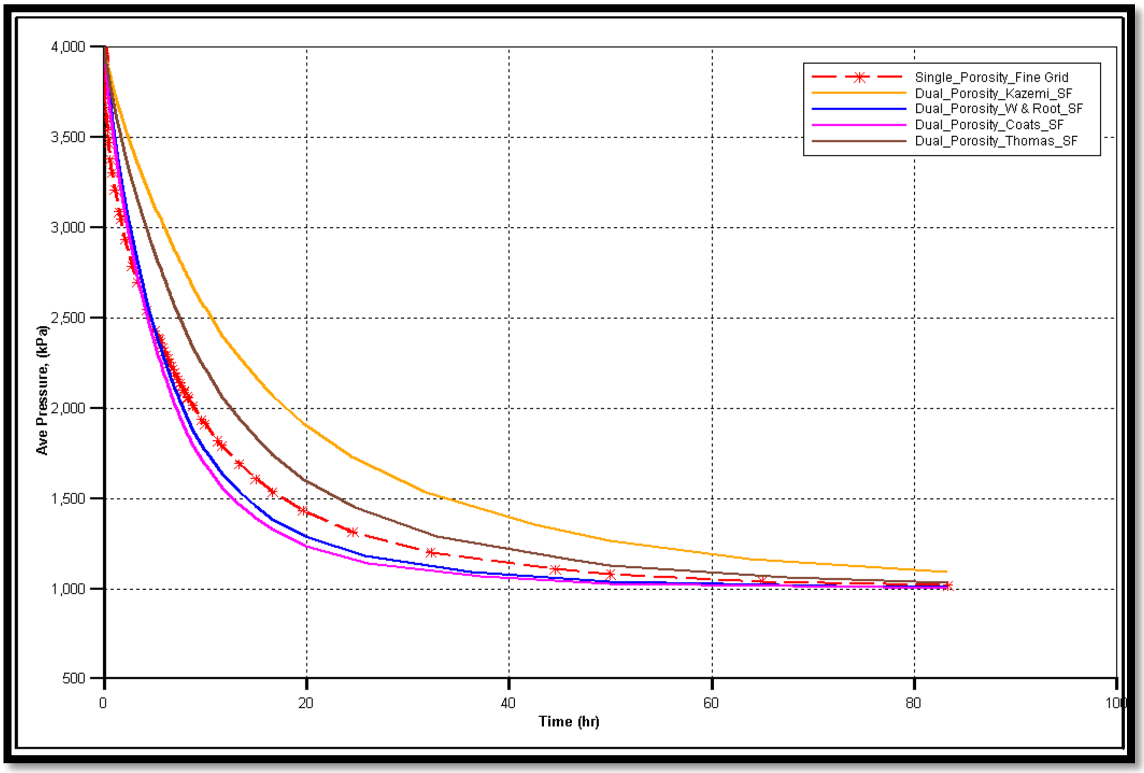

Figure 4.1.1. 4: Pressure drawdown comparison for different shape factor formulations (2D, multi-matrix blocks)

4.1.2. 3D Multi-Set Matrix Blocks

To comprehend the difference between 3D and 2D models, the same set of exercises as the 2D multi-matrix blocks cases are repeated and analyzed. Assume a single-porosity model with cubic matrix block dimensions of 2m x 2m x 2m and a grid resolution of 10cm x 10cm x 10cm that are surrounded by a set of fractures, while a dual-porosity model is defined with the same dimensions and grid resolution of 50cm x 50cm x 50 cm. Unlike the 2D cases, various matrix permeability is assigned for each layer (Figure 4.1.2.1); eventually, the system is initialized the same as the previous 2D cases. Finally, the results of the 3D dual-porosity models using Kazemi's shape factor are compared and validated against the fine grid single-porosity model (a reference solution). The variations of the pressure drawdown are shown in Figure 4.1.2.2. These results show decent evidence suggesting that Kazemi's shape factor appears to be an inappropriate model for thermal recovery process prediction in a dual-porosity system. Since the same conclusion is observed from the 3D model, in the entire study the 2D multi-matrix blocks case is used to evaluate the transient shape factor concept.

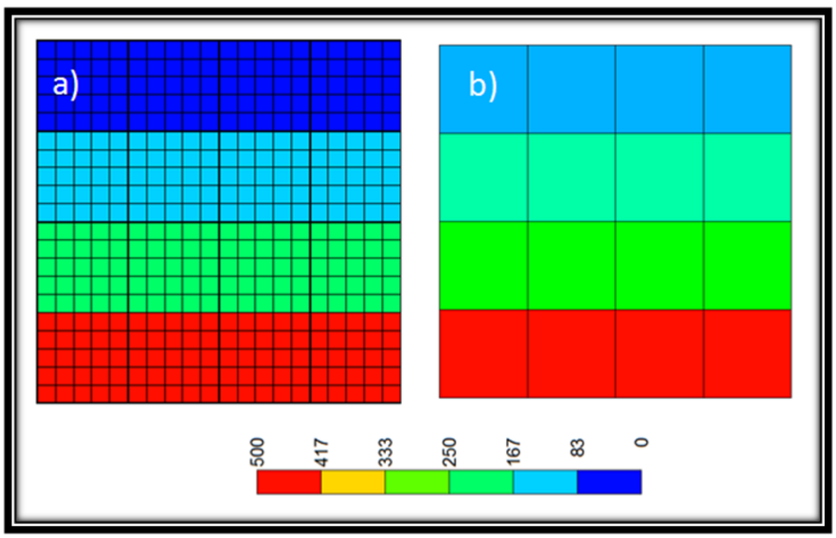

Figure 4.1.2. 1: Matrix permeability (md): a) fine-grid single-porosity; b) dual-porosity

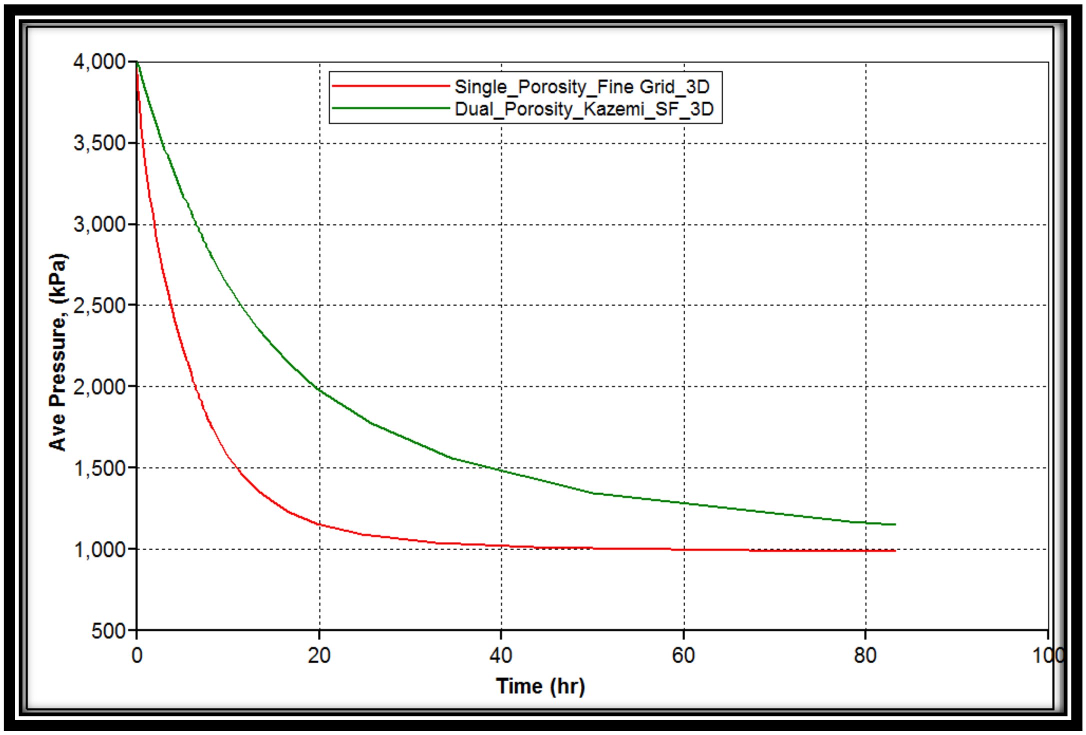

Figure 4.1.2. 2: Pressure drawdown comparison, single-porosity vs Kazemi's dual-porosity (3D model)

4.1.3. Transient Shape Factors, TSF (Concept)

A shape factor is essentially a time-dependent quantity and one of the most important parameters in the modeling of fractured reservoirs. Although some of the researchers acknowledged the time dependence of this factor for transient flow between matrix and fracture systems, in the most recent formulations a pseudo-steady state condition and constant shape factors are assumed for the exchange between fractures and matrix blocks in dual-porosity systems.

In this study, a series of numerical simulations are conducted to investigate the effect of a time-dependent shape factor in a dual-porosity model. In this work, Kazemi's model is evaluated by multiplying the shape factor by different constant values and the results are compared with the fine grid-single porosity model (reference solution). Figure 4.1.3.1 shows the comparison results of different multipliers. In the first three scenarios (cases 1 to 3), Kazemi's shape factor is multiplied by a constant value at the first time step for each individual case and the outcomes are compared with the reference solution. The results illustrate that a multiplier factor can only produce a good match with the reference curve either during the early time (case 2) or late time (case 3). Another scenario (case 4) is considered by multiplying the shape factor by different constant coefficients at various times and the goal was to match the coarse dual-porosity model with the fine grid single-porosity case. An excellent match is achieved by multiplying Kazemi's shape factor by constant values at different time steps (black dots). The result from this scenario (case 4) clearly confirms that a transient shape factor is required for an appropriate modeling of a thermal recovery process in naturally fractured reservoirs using dual-porosity systems. The purpose of these simulations is to demonstrate and establish the concept of a transient shape factor. Further discussions on defining a new transient shape factor formulation and how to estimate and compute the coefficient factors are presented in the following sections.

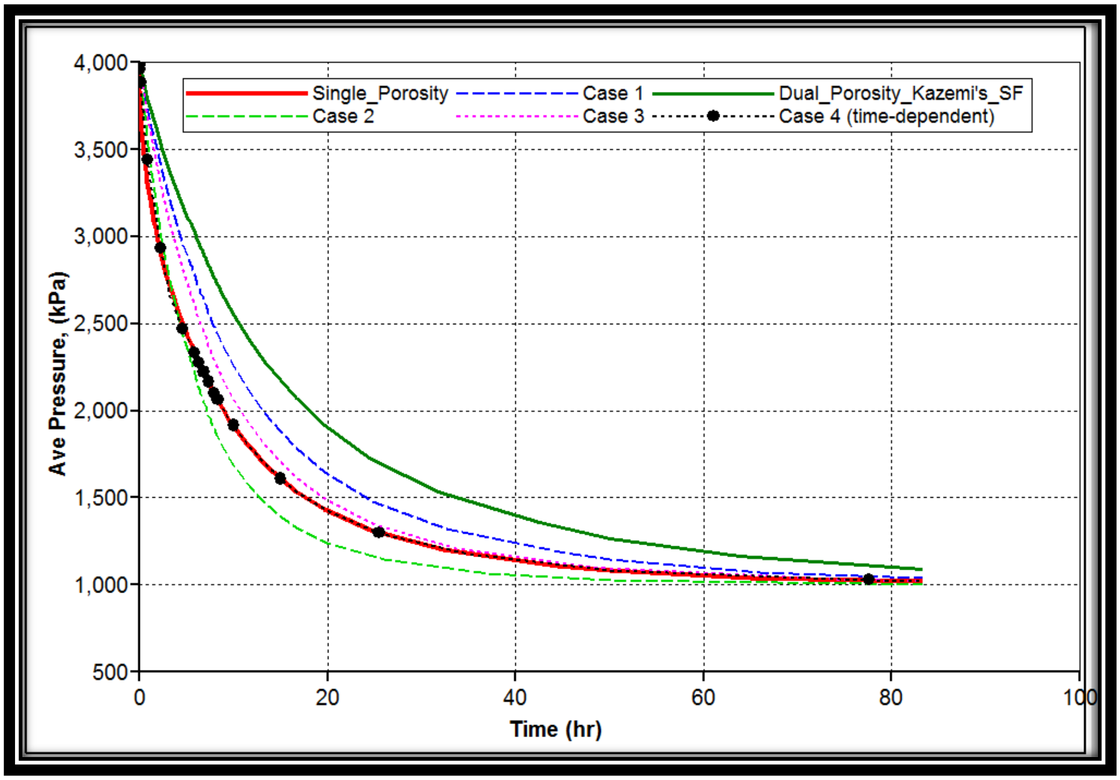

Figure 4.1.3. 1: Pressure profile, transient shape factor concept (Kazemi's model)

4.2. Transient Shape Factors (TSF) in Thermal Flow Simulation

There is an insufficient understanding of an adequate matrix-fracture transfer shape factor formulation in a dual-porosity system for thermal recovery mechanisms. A shape factor considers the fracture-matrix geometry in modeling naturally fractured reservoirs. The main limitation of the existing formulations for the shape factor is in the assumption that the matrix-fracture system remains at a pseudo-steady state condition.

In order to predict recovery under thermal mechanisms in fractured reservoirs for an optimal reservoir management using a dual-porosity system, suitable matrix-fracture transfer shape factors and transfer functions are required for the accurate flow behavior description between the fractures and the matrix. In this section a new derivation of a transient shape factor for thermal reservoir simulation is introduced and the methodology of how to compute it is presented. Finally, the results of using the transient shape factor formulation are validated and compared with a fine-grid single-porosity model and historical filed data using the commercial simulator, CMG STARS.

4.2.1. Model Description of Non-isothermal Process

In this section, the main equations for the flow and energy equations governing three phases: water, oil, and steam are briefly presented. The main goal of solving these equations is to obtain accurate matrix-fracture transient shape factor coefficients that can best represent the recovery methods in fractured reservoirs in a thermal process. The proposed model incorporates a three-dimensional, three-phase fluid physics in porous media for a thermal recovery process.

Flow Equations:

Mass balance in the matrix blocks and fractures for any phase in form of finite differences can be expressed as follows:

$$\nabla.\left[T_{\alpha m}\left(\nabla P_{\alpha m}-\gamma_{\alpha m}\nabla Z\right)-T_{\alpha mf}\left(P_{\alpha m}-P_{\alpha f}\right)\right]=\frac{V_b}{\Delta t}\left(\frac{\phi_m S_{\alpha m}}{\beta_{\alpha m}}\right) \qquad (4.2.1.1)$$

$$\nabla.\left[T_{\alpha f}\left(\nabla P_{\alpha f}-\gamma_{\alpha f}\nabla Z\right)+T_{\alpha mf}\left(P_{\alpha m}-P_{\alpha f}\right)\right]+q_{\alpha f}=\frac{V_b}{\Delta t}\left(\frac{\phi_f\, S_{\alpha f}}{\beta_{\alpha f}}\right) \qquad (4.2.1.2)$$

where α = water, oil, or steam and the subscripts *m* and *f* refer to the matrix and the fractures, respectively.

The flow transfer transmissibility can be defined as:

$$T_{\alpha mf} = \sigma_F \, . K_m \, {K_{r\alpha}} \big/ {\mu_\alpha . \beta_\alpha} \qquad (4.2.1.3)$$

where σ_F is a flow shape factor.

Energy Equations:

For the heat transfer equation, it is assumed that thermal equilibrium between the fluid and the rock matrix is instantaneously achieved (Shutler, 1969). The energy balance law is expressed as

$$\textit{Conduction H.T} + \textit{Convection H.T} + \textit{Heat Source} - \textit{Heat Loss} = \textit{Accumulation (Pore + Rock)}$$

Therefore, energy balance in the matrix blocks and fractures for any phase in the form of finite differences can be expressed as follows (assuming there are no convection in the matrix and no conduction in the fractures):

$$\nabla K_h \, . \nabla T - T^{\circ}{}_{\alpha mf} \left(T^{*}_{\alpha f} - T^{*}_{\alpha m} \right) = \frac{\partial}{\partial t} \left[\left(\phi \sum \rho_\alpha \, H_\alpha \, S_\alpha \right)_m + \left(1 - \phi_m \right) \rho_r C_r \, T \right]$$

(4.2.1.5)

$$\nabla . \left(\sum \rho_{\alpha f} \, {K_f} \big/ {\mu_{\alpha f}} \, H_{\alpha f} \left(\nabla P_{\alpha f} - \gamma_{\alpha f} \, \nabla Z \right) \right) + q_{Hf} + q_{Lf} + T^{\circ}{}_{\alpha mf} \left(T_{\alpha f} - T_{\alpha m} \right) = \frac{\partial}{\partial t} \left[\left(\phi \sum \rho_\alpha \, H_\alpha \, S_\alpha \right)_f \right]$$

(4.2.1.6)

where α = water, oil, or steam and the subscripts *m* and *f* refer to the matrix and the fractures, respectively.

4.2.2. Solution Methodology

Due to the nonlinearity of the equations and their dependence on each other, it may not be possible to find their analytical solutions. A numerical technique is required to obtain the solutions. The solutions of the above finite difference equations by the numerical technique give the estimations of the flow transient shape factors, σ_{F_t} .

In this study, only Kazemi's shape factor model is considered to be corrected and converted to the transient form. Nevertheless, this methodology can likewise be used to compute the other shape factor formulations and estimate the transient shape factors. The main objective here is to find the α factor at different time steps to convert the shape factor to a transient form. Equation (4.2.2.1) shows Kazemi's formulation, which was under the assumptions of a pseudo-steady state condition in terms of fracture spacing for a system with uniform fracture spacing of L_k (the α coefficient is equal to 4):

$$\sigma = \alpha \sum_{k=1}^{nd} {}^{1}/_{L_k^2} \tag{4.2.2.1}$$

To compute a transient shape factor, a dimensionless average pressure P_{D_t} is introduced as:

$$P_{D_t} = \left[\frac{P_f}{P_m - P_i} \right]_t^{\omega_t} \qquad (4.2.2.2)$$

ω_t is an exponent factor which is a time-dependent factor and can be expressed as a ratio of the matrix pressure and the fracture pressure at each time step (P_i is the initial reservoir pressure):

$$\omega_t = \psi \left(\frac{P_m}{P_f} \right)_t \qquad (4.2.2.3)$$

Ψ is a coefficient factor, which depends on the rock wettability ranging from 1.25 to 3.75. Based on this study, a factor of 1.45 is recommended for an oil-wet rock formation (more detailed discussions are described in Chapter 5). However, this factor is reservoir-dependent and can be used as one of the matching parameters. To estimate the transient shape factor σ_{Ft} Kazemi's formulation (or any other formulations) can be multiplied by the dimensionless average pressure, P_{D_t} :

$$\sigma_{Ft} = P_{D_t} . 4 \sum \left(1 / L_i^2 \right) \qquad (4.2.2.4)$$

where

$$P_{D_t} = \alpha \qquad (4.2.2.5)$$

As mentioned earlier, equation (4.2.2.6) was expressed as a transfer function (Warren and Root, 1963), where T is the transfer function (1/sec), σ is defined as a shape factor ($1/m^2$), K is the matrix permeability (md), μ is the fluid viscosity (c.p), P is the pressure (atm), and subscripts f and m are for fracture and matrix denominations, respectively. The matrix-fracture transfer shape factor (σ) has the dimension of a reciprocal area, L^{-2}:

$$T = \sigma \frac{K}{\mu}(P^m - P^f) \qquad (4.2.2.6)$$

Kazemi et al. (1976) developed a numerical algorithm and proposed the following expression for a shape factor based on direct material balance on a cubic matrix block with the α coefficient equal to 4:

$$\sigma = 4\sum_{k=1}^{nd} {}^{1}/_{L_k^2} \qquad (4.2.2.7)$$

From the transient shape factor formulation developed earlier (Equation 4.2.2.4), by utilizing the Kazemi and Gilman shape factor formulation and substituting it into Equation (4.2.2.6), a novel Thermal Transfer Function (T.T.F) can be obtained as follows for modeling a thermal recovery process in naturally fractured reservoirs:

$$T.T.F = 4\sum_{k=1}^{nd} {}^{1}/_{L_k^2} \,.\left(\frac{P_{fi}}{P_{mi} - P_i}\right)^{\omega t} .K.K_r/\mu \; (P_m - P_f) \qquad (4.2.2.8)$$

In order to compute P_{D_t} and ω_t, the average fracture and matrix pressures are required at different time steps, which can be obtained using any commercial thermal simulator. In this study, the CMG STARS simulator is used to estimate and validate the proposed transient shape factor. A MATLAB code is developed and coupled with the CMG STARS to compute the P_{D_t} and ω_t parameters for estimating the transient shape factor (TSF) at any desired time step and updates the flow model likewise. At each required time step, the fracture and matrix pressures are extracted from the STARS OUT data file (*.out) to compute the exponent factor of ω_t (Equation 4.2.2.3). Next, using Equation (4.2.2.2), dimensionless average pressure P_{D_t} can be calculated by applying the developed MATLAB code (Appendix B). This calculated P_{D_t} at time step t is used to modify the matrix-fracture transfer shape factor coefficient at that time step and resume the flow model to the next time step. The above calculation steps are repeated for all the desired time steps to the last simulation time. Figure 4.2.2.1 represents the flowchart of this coupling procedure for estimating the transient shape factor coefficient by implementing this technique using the commercial simulator CMG STARS. Figure 4.2.2.2 shows the interface of the coupling application, which is referred to as a "TSF Estimator" tool kit in this work.

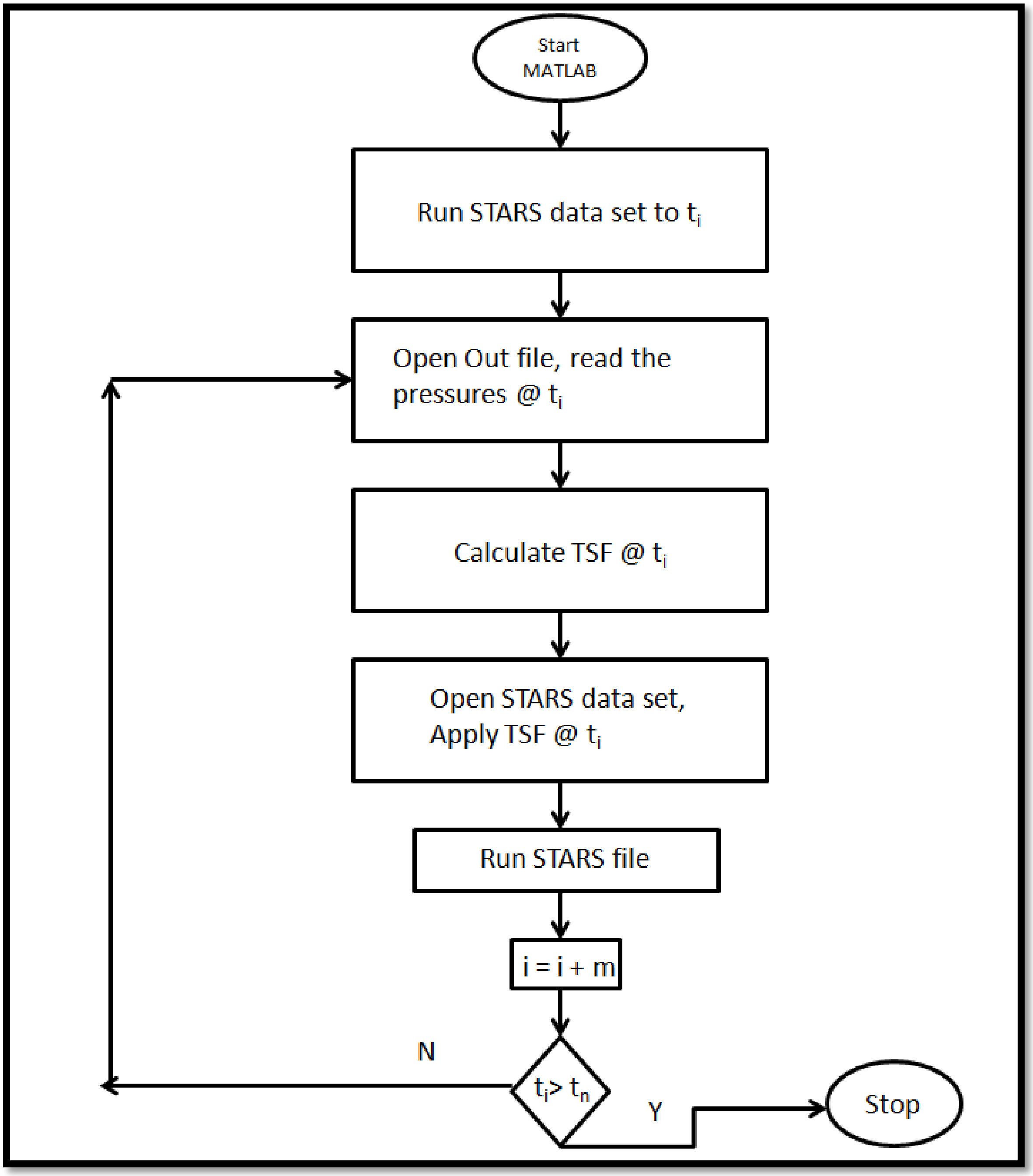

Figure 4.2.2. 1: Flowchart of TSF Estimator Tool kit

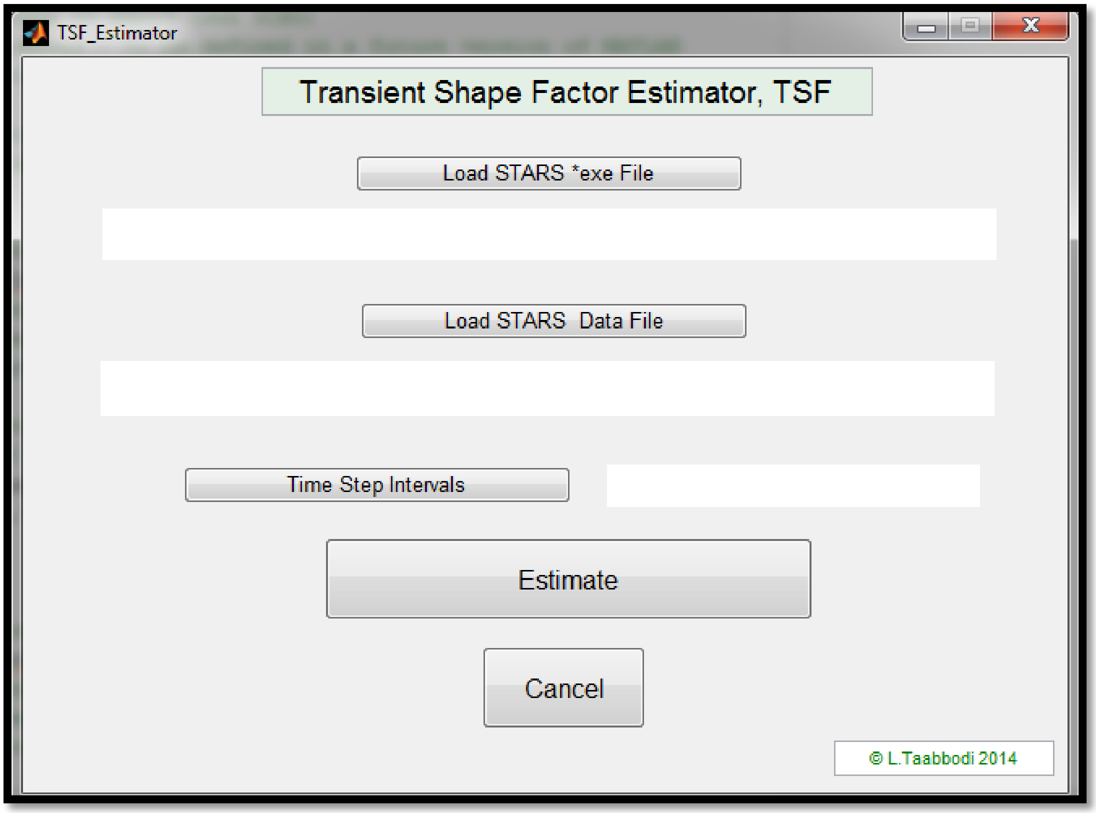

Figure 4.2.2. 2: TSF Tool kit interface (Demo)

4.2.3. Utilizing Transient Shape Factors (TSF)

The same 2D multi-matrix blocks dual-porosity model as described early (Figure 4.1.1.3, 2m x 2m x 2m size dimensions) is used to compute and evaluate the transient shape factor using Kazemi`s formulation. A steam drive recovery method is selected to evaluate and validate the transient shape factor for this assessment. The saturated steam is injected at one end at 4,000kPa while the heated bitumen is producing at the minimum bottom-hole pressure of 1,000kPa at the other end. Figure 4.2.3.1 depicts the schematic of the numerical model used in this study.

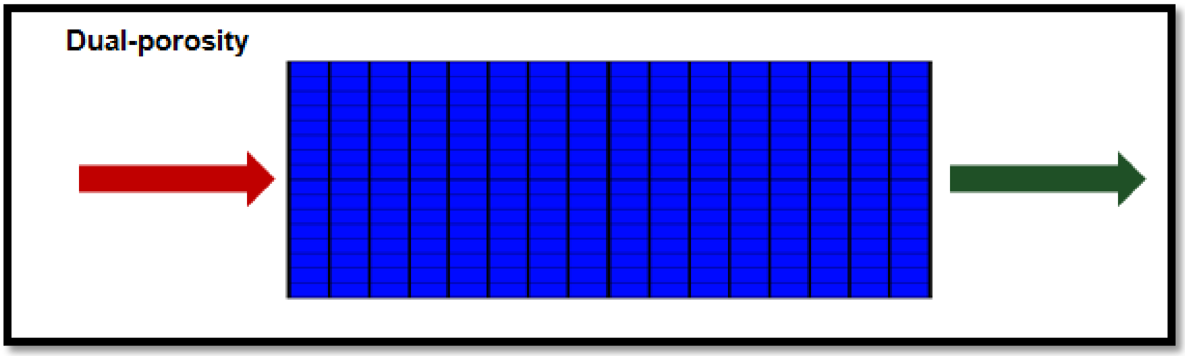

Figure 4.2.3. 1: 2D multi-matrix blocks dual-porosity model (steam drive)

The TSF Estimator tool is used to estimate the transient shape factor coefficients and convert the constant Kazemi's formulation to transient mode. The model is run at each desired time step and updated by implementing the transient shape factor formulation at each specified time step. Figure 4.2.3.2 compares the pressure profiles of the default dual porosity model (Kazemi's formulation) with the corrected model (transient shape factor formulation), and they are validated against the fine grid single porosity model (reference solution). Figures 4.2.3.3 and 4.2.3.4 present the oil saturation and cumulative oil-steam ratio comparisons. The green lines are presenting the reference solution and the dual porosity model using Kazemi's constant shape factors, and the dots show the outcomes of the transient shape factor model. The results from this evaluation clearly illustrate that a significant improvement is achieved using the transient shape factor (TSF) model, which demonstrates a very good agreement with the reference solution. The dimensionless pressure coefficient factors ($P_{D_t} = \alpha$) versus time are shown in Figure 4.2.3.5. The results indicate that different coefficient factors at different times are required for correcting and converting the constant shape factor to a transient shape factors form.

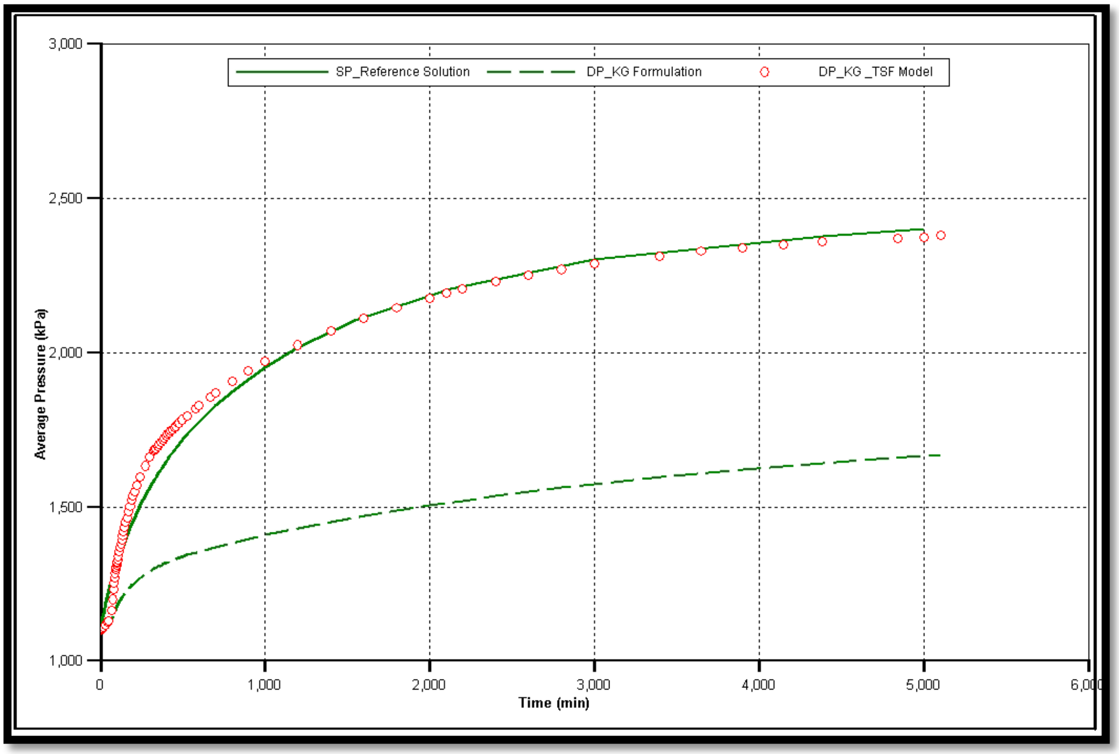

Figure 4.2.3. 2: Average pressure profile, with and without the TSF model ($\Psi = 1.45$)

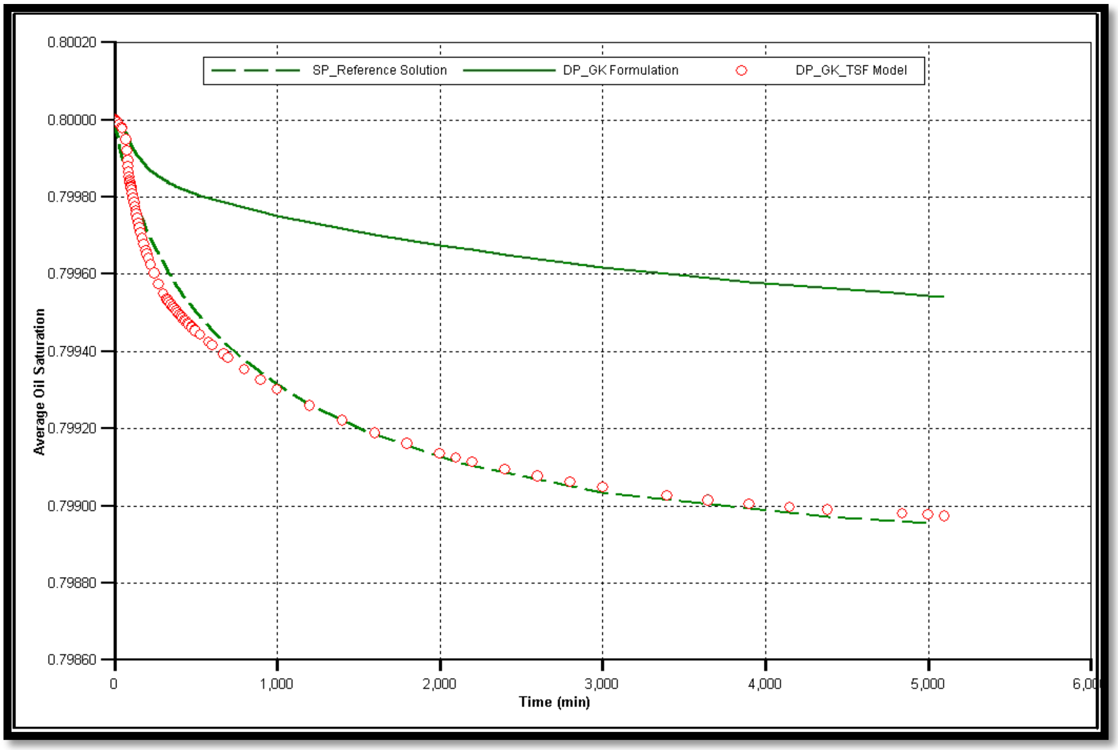

Figure 4.2.3. 3: Average oil saturation profile, with and without the TSF model (Ψ = 1.45)

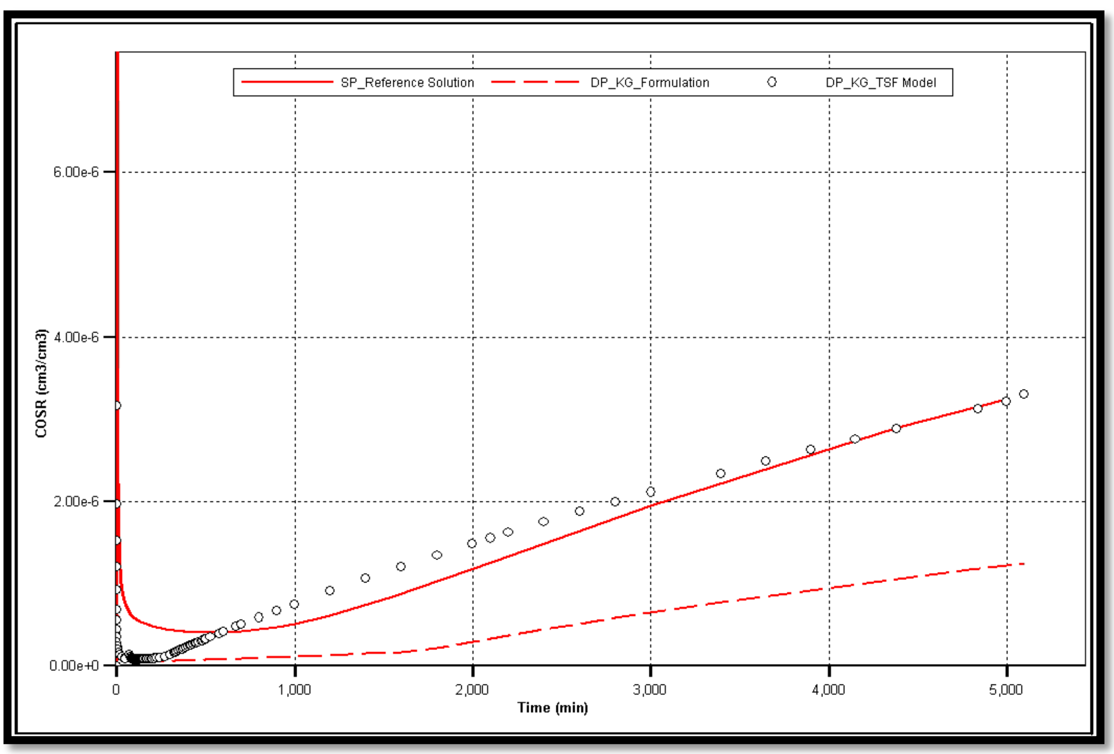

Figure 4.2.3. 4: Cumulative oil steam ratio (COSR), with and without the TSF model (Ψ = 1.45)

The results of the dimensionless pressure coefficient factor plot (Figure 4.2.3.5) indicate that different coefficient factors at different times are required for correcting the constant shape factor to a transient shape factors form. This plot can be divided into three regimes, early-time, mid-time, and late-time. Usually, a sharper slope is expected for the early time regime compared to the other ones. This steep behavior can be explained due to the extreme changes in the pressure and/or saturation profiles in the early time of fluid (steam) injection, which require higher coefficient factors to correct and modify the constant shape factor formulation to the transient mode. Note that this coefficient profile is very case-dependent, which is mainly subject to the size of the model, the grid resolutions, the reservoir parameters, and operation conditions. Since the model used for this study is a relatively small 2D model, a steeper slope is observed for the early-time period. Nevertheless, an exponential type curve versus time relationship is expected to yield the transfer of a constant shape factor formulation to a transient shape factor type.

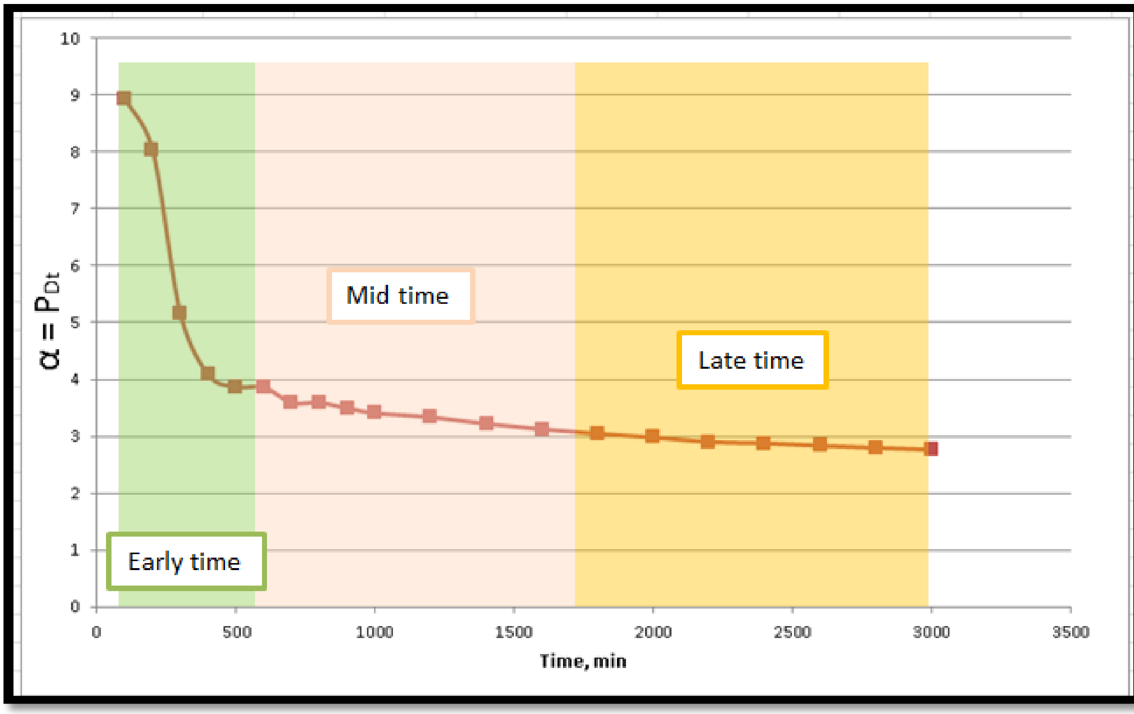

Figure 4.2.3. 5: Dimensionless pressure coefficient factors ($P_{D_t} = \alpha$)

4.3. Concluding Remarks

- Most of the existing transfer function formulations (matrix-fracture interaction) are based on the assumption of orthogonal fracture systems and pseudo-steady state flow, which are not representative of actual reservoir recovery mechanisms.
- The most important and challenging aspect of modeling a naturally fractured reservoir is the accurate estimation of the fluid exchange between the matrix and surrounded fractures.
- Overall results demonstrate that none of the existing dual-porosity models can be used to generate a consistent match at all times with the reference solution;

particularly, Kazemi's formulation shows the maximum deviation from the reference solution (fine grid single-porosity).

- Suitable matrix-fracture transfer shape factors and transfer functions are required for the accurate flow behavior description between the fractures and the matrix in a thermal flow simulator.
- A new transient shape factor (TSF) for non-isothermal dual-porosity models is introduced and compared with the existing shape factor models. The results clearly confirm that a transient shape factor is required for appropriate modeling of a thermal recovery process in NFRs when using dual-porosity formulations.
- A MATLAB code is developed and coupled with the CMG STARS thermal reservoir simulator to compute the transient shape factor (TSF Estimator tool kit).
- The results indicate that during the early time the pressure drop is steeper for the reference solution compared to the dual-porosity model using the existing shape factors, which essentially suggests that all the current shape factors are smaller than that required.
- Clearly a constant value of the shape factor (e.g., Kazemi's model) cannot be used to predict the overall performance in dual-media systems due to invalid assumptions.
- The transient shape factor estimated in this study is based on Kazemi's formulation only. However, this methodology can be used to investigate and evaluate the other shape factor formulations and estimate the transient shape factors.

CHAPTER 5

5.1. Sensitivity Study

The purpose of this section is to identify some of the important factors that have the largest impact on the deviation of a dual-medium model from a reference solution with the use of Kazemi's shape factor formulation. In this section, the main parameters sensitivity studies for a water-oil system are performed and the effect of each parameter is compared with the reference solution case (fine grid single-porosity). The following sensitivity cases are investigated:

- Grid Block Size Distribution
- Rock Wettability
- Capillary Pressure
- Initial Water Saturation
- Reservoir Heterogeneity
- Fracture-Matrix Effective Permeability Ratio

5.1.1. Grid Block Size Distribution

Block size distribution is considered to be one of the main parameters in the modeling of fractured reservoirs. Depending on the rock wettability, block size distributions can control the rate of production by imbibition, or displace fluid from the blocks into the fractures. The matrix block size is inversely related to the fracture intensity. Therefore, for an accurate production prediction, it is essential to characterize and model fractured porous media by considering variable matrix block size features (Belani, 1988).

The effect of variable matrix block size distributions on the matrix-fracture fluid transfer in a dual-porosity system using Kazemi's shape factor is briefly investigated utilizing numerical simulation, and the results are compared with the fine grid single-porosity model (reference solution). For these runs, the same 3D multi-set matrix blocks model as described earlier is used. Figure 5.1.1.1 shows the selected random matrix block size distribution models in this work. In the classical dual porosity model, all matrix block geometry and properties are assumed to be uniform within each simulation cell. Nevertheless, most of the fractured formation outcrops suggest the heterogeneity of the matrix blocks in the reservoir scale, which come in all shapes and sizes (Ahmad, 2014). Figure 5.1.1.2 shows the outcrop of a fractured carbonate formation demonstrating the potential reservoir complexity. Figure 5.1.1.3 shows the comparison results against the reference solution for different matrix block size distributions. The results demonstrate that the more non-uniform the matrix block sizes feature (Dist_1 case), the more divergence it shows from the reference solution curve; therefore, more thoughts should be considered in modeling dual media using constant shape factor formulations such as Kazemi's model to obtain more accurate results.

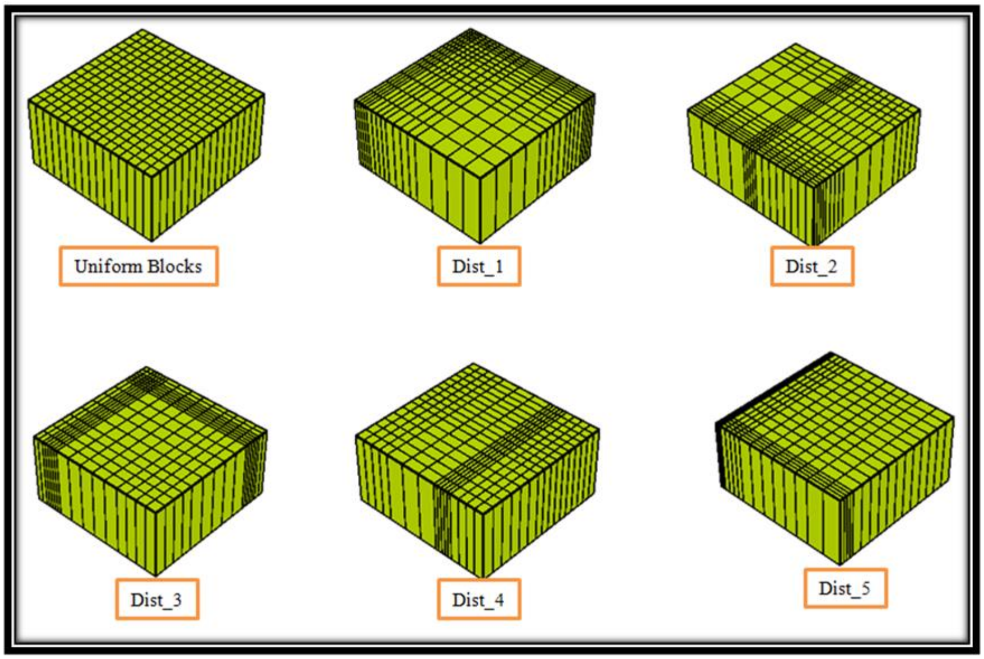

Figure 5.1.1. 1: Variable random block size distribution (dual-porosity)

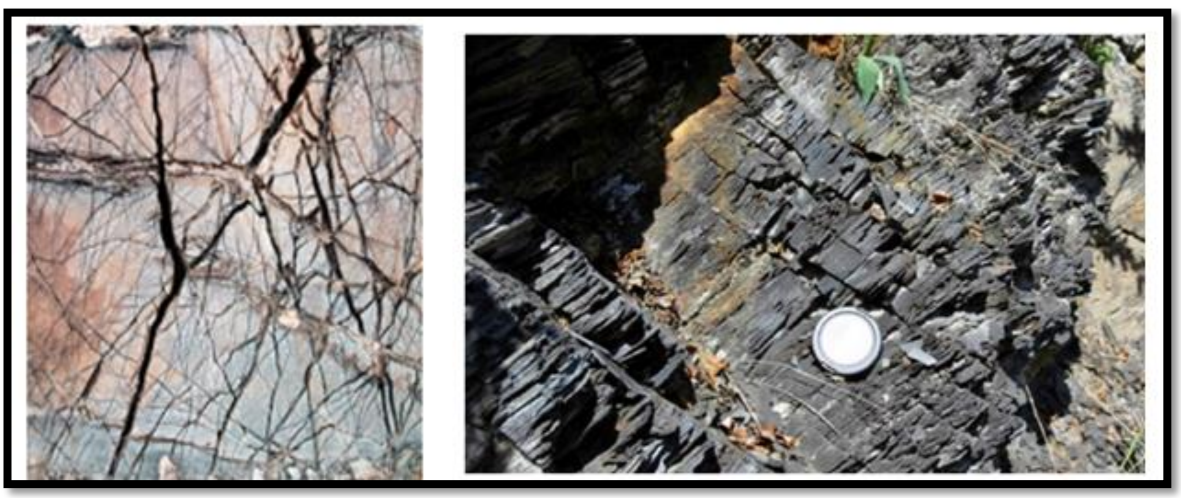

Figure 5.1.1. 2: Outcrop from fractured carbonate formation

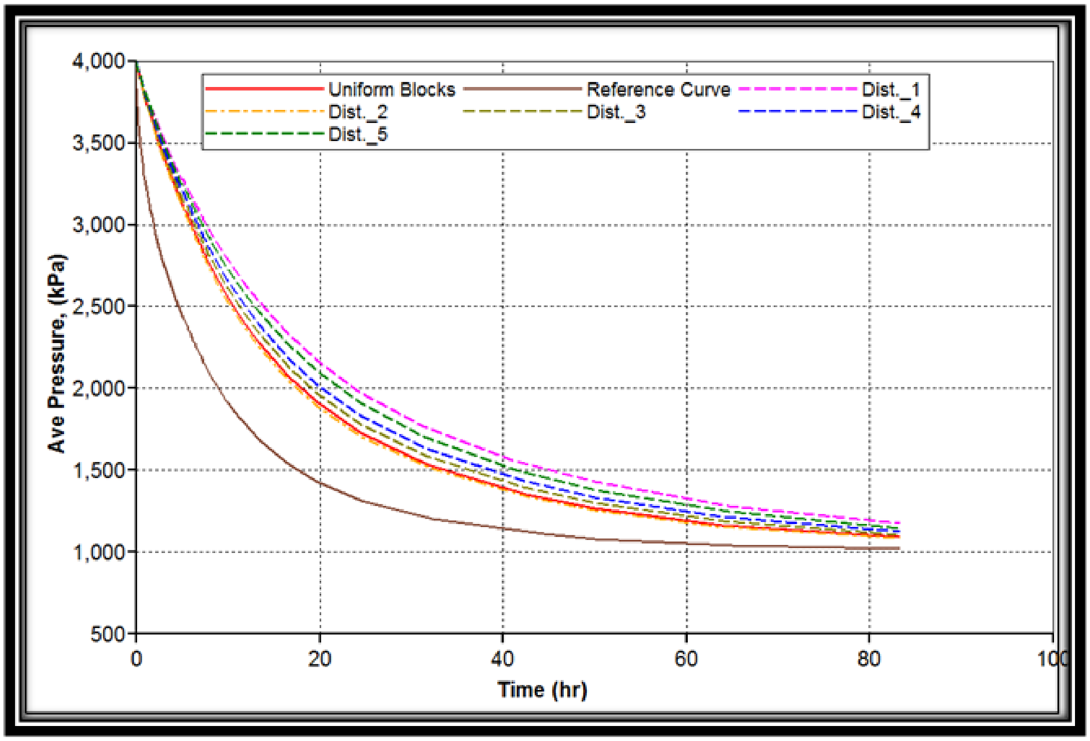

Figure 5.1.1. 3: Block size distributions sensitivity against reference solution

5.1.2. Rock Wettability

Extensive research work from examination of 161 limestone, dolomitic limestone, calcitic dolomite, and dolomite cores on wettability of carbonate reservoir rocks by different researchers indicate that 15 percent of these rocks are strongly oil-wet (some are bitumen coated), 65 percent are oil-wet, 12 percent have intermediate wettability, and 8 percent are water-wet (Chilingar and Yen, 2007). These results on carbonate reservoir rock cores from all over the world show that the majority of carbonates are mostly oil-wet. Experiments on cores from fields in Oman and elsewhere have indicated that the rock is in transition from oil-wet to water-wet as temperature increases. The temperature can be increased in a reservoir due to steam or hot-water injection, which can induce a wettability change, rendering the matrix water-wet. Hot water (steam condensate) in the fractures can spontaneously imbibe into the matrix, displacing oil and resulting in favourable oil recoveries.

It is well known that the recovery is very sensitive to the wettability of the matrix blocks, which mainly controls the recovery during thermal processes. Since carbonate rocks are mostly oil-wet or mixed wet, it is important to examine the effect of wettability on the matrix-fracture fluid transfer in a dual-porosity system using Kazemi's shape factor. In this section, two wettability cases, oil-wet and mixed-wet are considered and evaluated. The same 2D multi-set matrix blocks model is used to investigate the impact of the rock wettability. Consequently, the results are compared and validated against the fine grid single-porosity model. Lastly, the proposed transient shape factor (TSF) concept is tested and the results are analyzed for both cases. Figures 5.1.2.1 and 5.1.2.2 show the comparison of the average pressure profile for the oil-wet and mixed-wet cases, respectively. The dashed lines are the reference solutions, the solid lines are the dual-porosity solutions using Kazemi's formulation, and the dots are the transient shape factor profiles (TSF). The results indicate that in the case of mixed wettability formation, a transient shape factor is not perfectly corrected using the Kazemi's formulation compared to the oil-wet case, mostly in the late time. Nevertheless, this clearly illustrates that dual-porosity models, regardless of whether they are oil-wet or mixed-wet cases, can be improved significantly by employing the proposed transient shape factor (TSF). The results show a very good agreement with the reference solution.

As mentioned earlier, the coefficient factor (Ψ) in Equation (4.2.2.3) is defined to be in the range of 1.25 to 3.75, within which a factor of 1.45 is recommended for an oil-wet rock formation. This factor is also reservoir-dependent and it can be simply used as a matching parameter in a dynamic flow simulation.

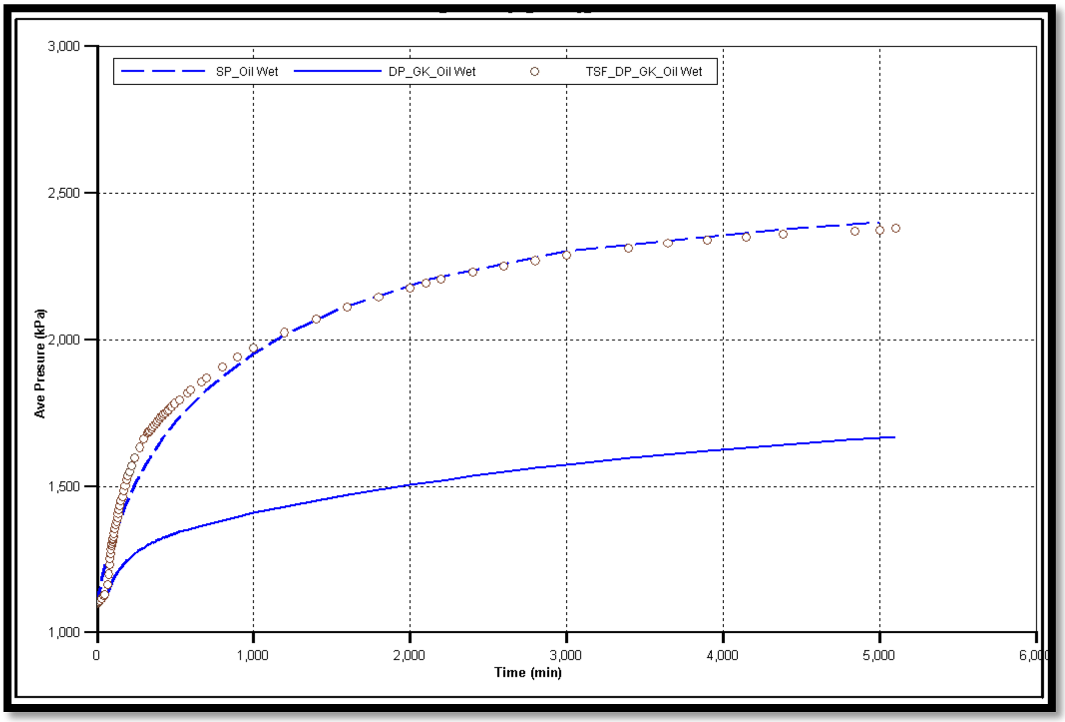

Figure 5.1.2. 1: Average pressure profile comparison, oil-wet system, with and without TSF

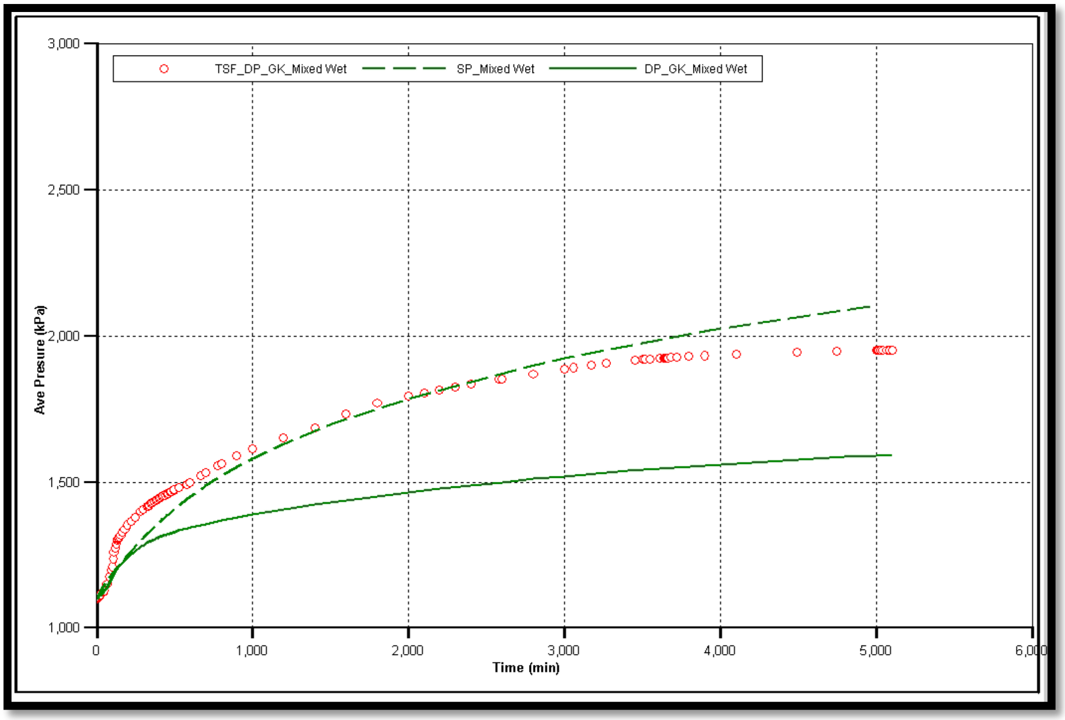

Figure 5.1.2. 2: Average pressure profile comparison, mixed-wet system, with and without TSF

5.1.3. Capillary Pressure

Reservoir rock typically contains the three phases: oil, water, and gas. The forces that hold these fluids in equilibrium with each other and with the rock are capillary forces. Capillary pressure, P_c, is defined as the pressure difference between the non-wetting phase and the wetting phase, which depends strongly on the wettability of the formation. For an oil and water system in porous rock (water-wet), oil is considered to be the non-wetting phase, while for an oil-wet system, water is considered to be the non-wetting phase. In reservoir engineering, P_c is an important parameter for simulation studies particularly in heterogeneous systems. In naturally fractured reservoirs, capillary pressure and relative permeability functions have a major impact on fluid exchange between matrix blocks and fractures. The fluid transfer between fractures and matrix blocks is dominated by capillary and gravity forces. Since most oil is contained inside the matrix, capillary and gravity forces can be more important in NFRs compared to un-fractured reservoirs. For example, capil1lary forces may either enhance or reduce recovery from matrix blocks depending on wettability (e.g., Gilman and Kazemi, 1988; Gang and Kelkar, 2008).

The same 2D multi-set matrix blocks model is used to investigate the impact of the case with or without capillary pressure. Figure 5.1.3.1 shows the results of the transient shape factor for both cases, with and without the presence of capillary pressure against the reference solutions. The dimensionless pressure coefficient factor ($\alpha = P_{Dt}$, in Equation 4.2.2.5) versus time indicates that a relatively higher coefficient factor is required for the case with capillary pressure compared to the case where there is no capillary pressure (Figure 5.1.3.2). From this, it can be concluded that for the case with the higher capillary pressure, the more divergence from the reference solution is expected using constant shape factor formulations such as Kazemi's model and it can be subsequently modeled more accurately by applying the transient shape factor concept to modify the factors as a time-dependent function.

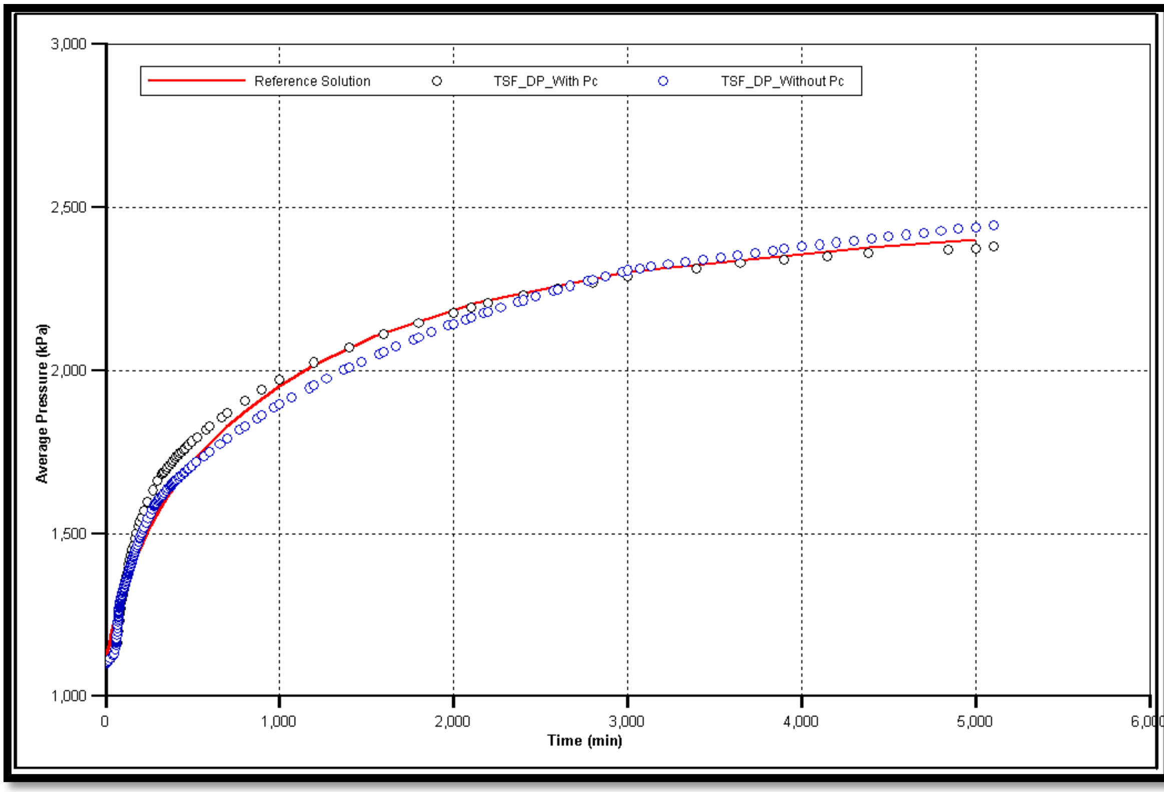

Figure 5.1.3. 1: Average pressure profile comparison, with and without capillary pressure

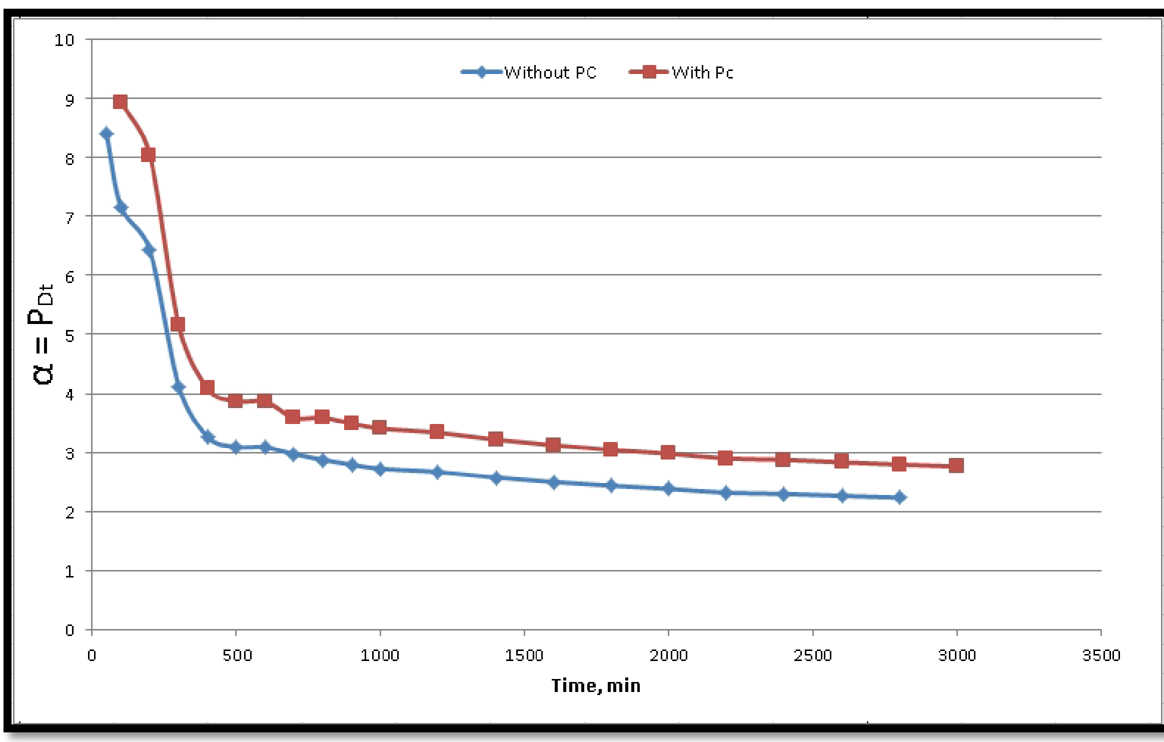

Figure 5.1.3. 2: Dimensionless pressure comparison, with and without capillary pressure

5.1.4. Initial Water Saturation

Water saturation in fractured reservoirs is a complex issue and has been studied extensively. Initial water saturation has a pronounced effect on fluid (steam) injection in an intermediate-wet rock and much less obvious in a strongly water-wet case. The effect of relatively high and low water saturation (for both the matrix and fractures) on the matrix-fracture fluid transfer in a dual-porosity system using Kazemi's shape factor is briefly investigated using a numerical simulation technique, and the results are compared with the fine grid single-porosity model (reference solution). For this evaluation, the same 2D multi-set matrix blocks model is applied. The dimensionless pressure coefficient factor ($\alpha = P_{Dt}$ in Equation 4.2.2.5) versus time indicates that a higher coefficient factor is required for the case where the water saturation is lower in the matrix and fractures compared to the case of the higher water saturation (20% saturation versus 40% saturation). The dimensionless pressure coefficient factors (P_{Dt}) are shown in Figure 5.1.4.1. The results indicate that for a system with lower water saturation values, using constant shape factor formulations such as Kazemi's model leads to higher inaccuracy in the prediction; therefore, applying a transient shape factor concept is recommended for more accurate modeling of a dual porosity system in thermal recovery processes.

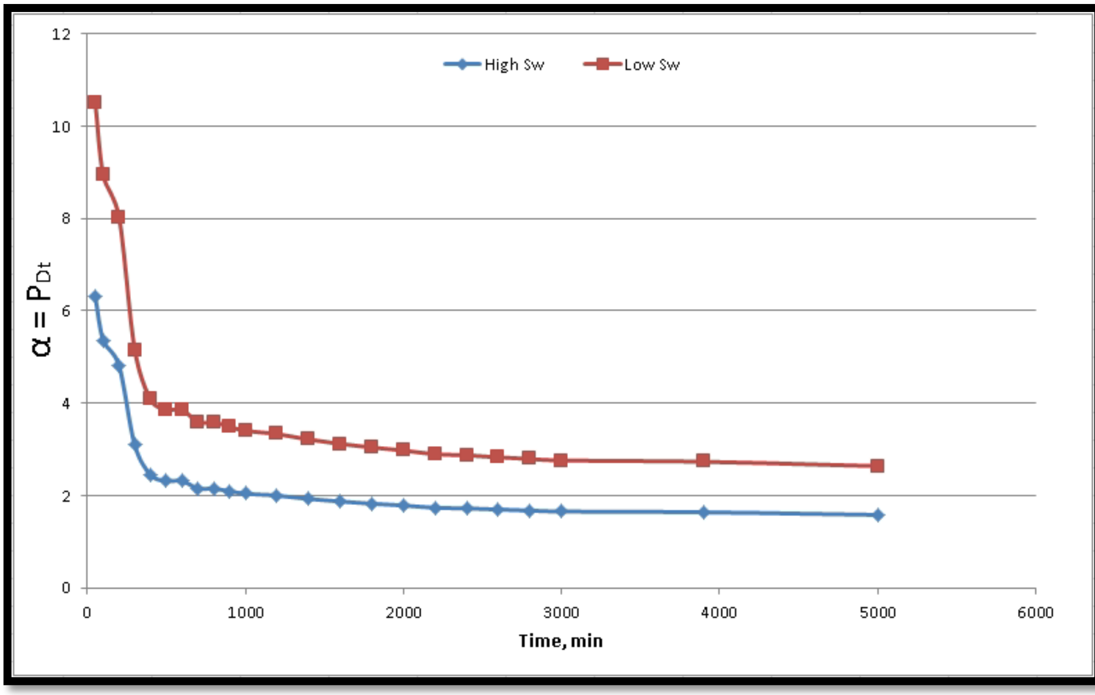

Figure 5.1.4. 1: Dimensionless pressure comparison, low and high water saturation

5.1.5. Reservoir Heterogeneity

In comparison to sandstone reservoirs, modeling of carbonate reservoirs is commonly more challenging because of intrinsic heterogeneities, occurring at all scales of observation and measurement. Heterogeneity in carbonate reservoirs can be attributed to variable lithology, chemistry/mineralogy, pore types, pore connectivity, and sedimentary facies, which present a number of specific characteristics posing complex challenges in reservoir characterization, modeling and management.

Since fractured reservoirs are usually very heterogeneous, the accuracy of the most common shape factor formulation (the Kazemi and Gilman model) is examined and validated in this section. Figures 5.1.5.1 and 5.1.5.2 show the core photos of a fractured carbonate formation, which describes the complexity of a fractured reservoir including varied lithology. The same 2D multi-set matrix blocks model with dimensions of 2m x 2m x 2m including multi-fractures and matrix blocks with the grid resolution of 1cm x 12.5cm x 100cm is used to investigate the impact of the reservoir heterogeneity. The various matrix permeabilities and water saturations are defined to represent the most common reservoir facies in fractured carbonate reservoirs such as mudstone, vugs, tight formation as well as existence of high unconsolidated oil sands depositions. In this study, random matrix properties (permeability and water saturation) are assigned through the reservoir to model the formation complexity, while retaining fracture properties as constant values. Figures 5.1.5.3 and 5.1.5.4 show the matrix permeability and water saturation distributions for both single and dual porosity systems, respectively. Subsequently, the proposed transient shape factor (TSF) concept using the developed "TSF Estimator" tool is examined and the results are compared against the homogenous case for a steam-drive recovery mechanism. Figure 5.1.5.5 shows the dimensionless pressure coefficient factors ($\alpha = P_{Dt}$) for both the homogenous and heterogeneous cases. The results demonstrate that due to the heterogeneous nature of carbonate reservoirs, Kazemi's formulation requires a higher coefficient factor mainly in the early time to convert to the transient shape factor form with less inaccuracy of outcomes. Note that the estimated coefficient factor is reservoir-dependent and it may vary for different reservoir geology. However, regardless of the heterogeneity, it is recommended to utilize the transient shape factor approach for more precise predictions of modeling a dual-porosity systems during thermal recovery processes.

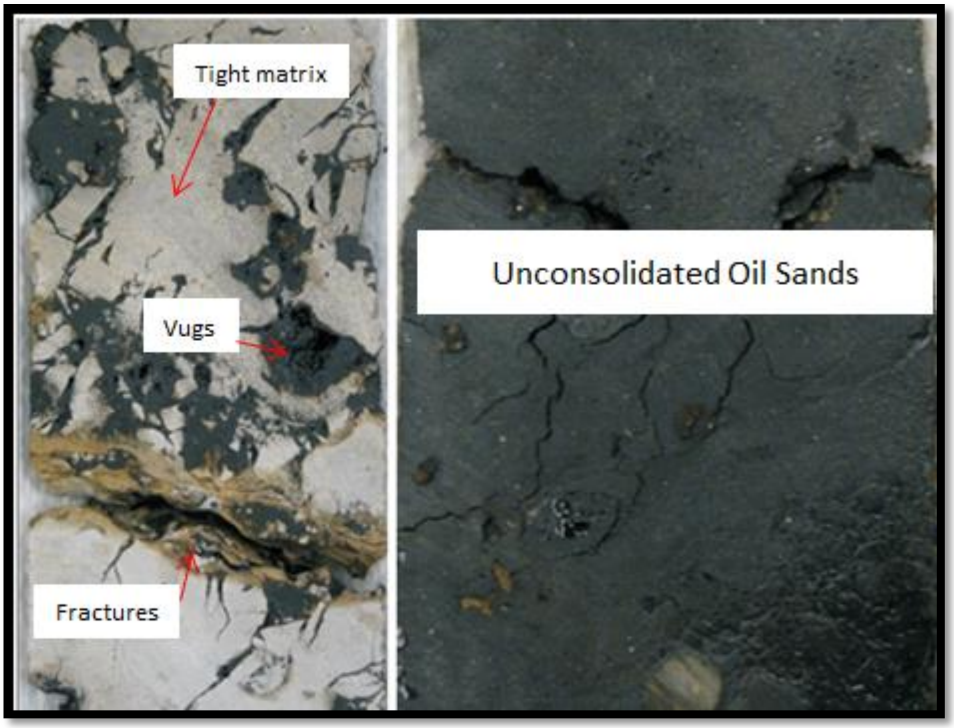

Figure 5.1.5. 1: Bitumen fractured reservoir (pilot application, Husky Energy, 2013)

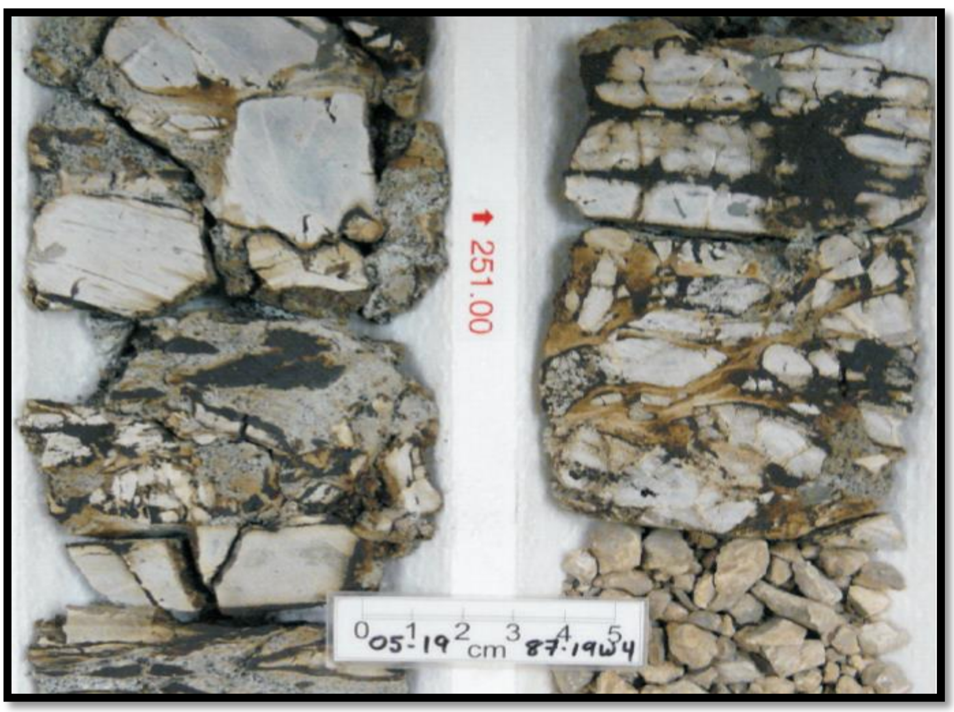

Figure 5.1.5. 2: Karst breccia with Cretaceous siliciclastic fill (pilot application, Husky Energy, 2013)

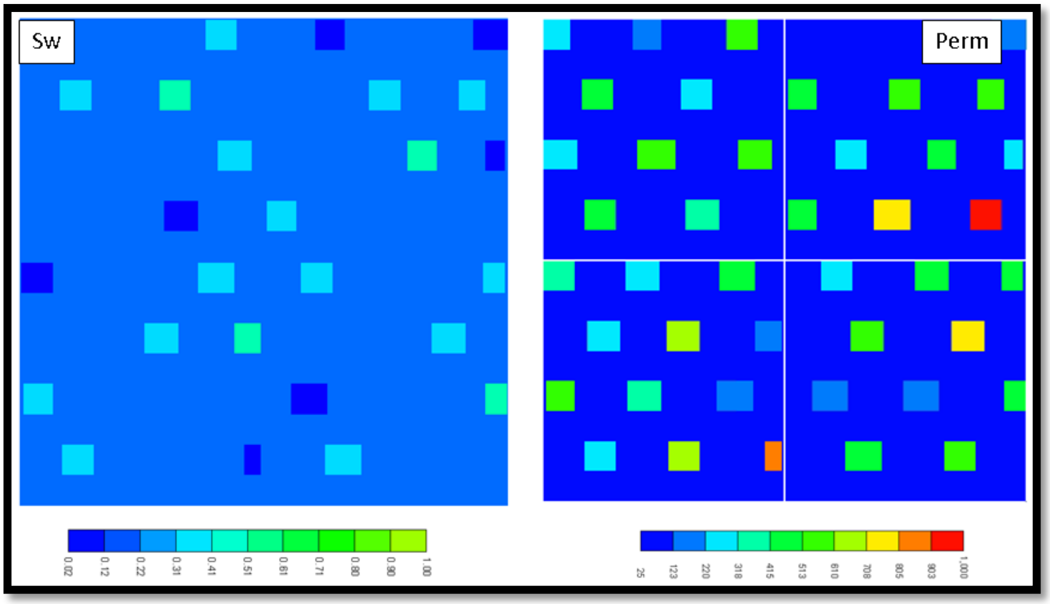

Figure 5.1.5. 3: Water saturation and permeability profile (single porosity)

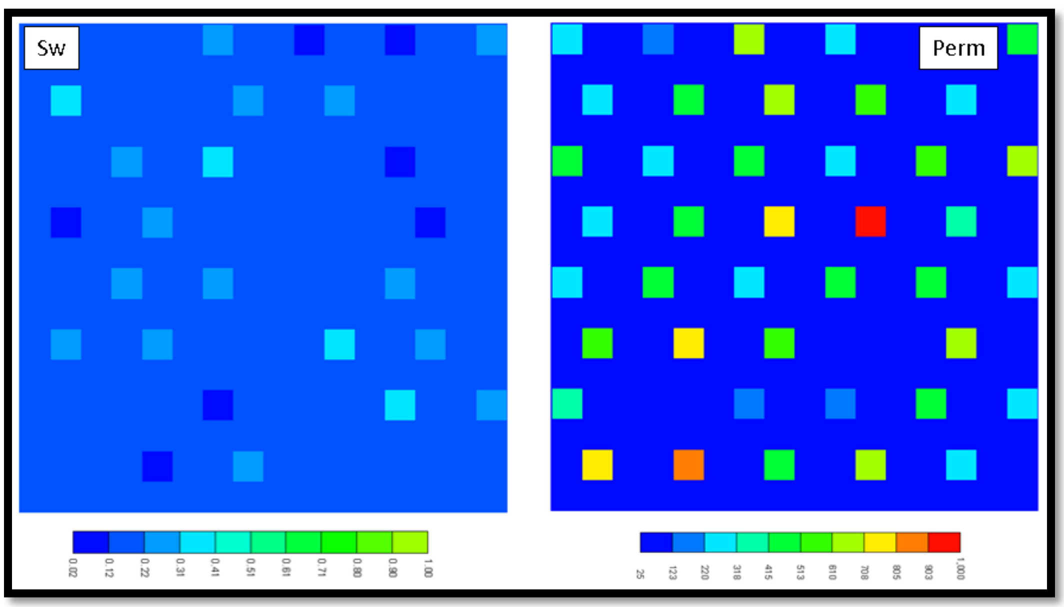

Figure 5.1.5. 4: Water saturation and permeability profile (dual-porosity)

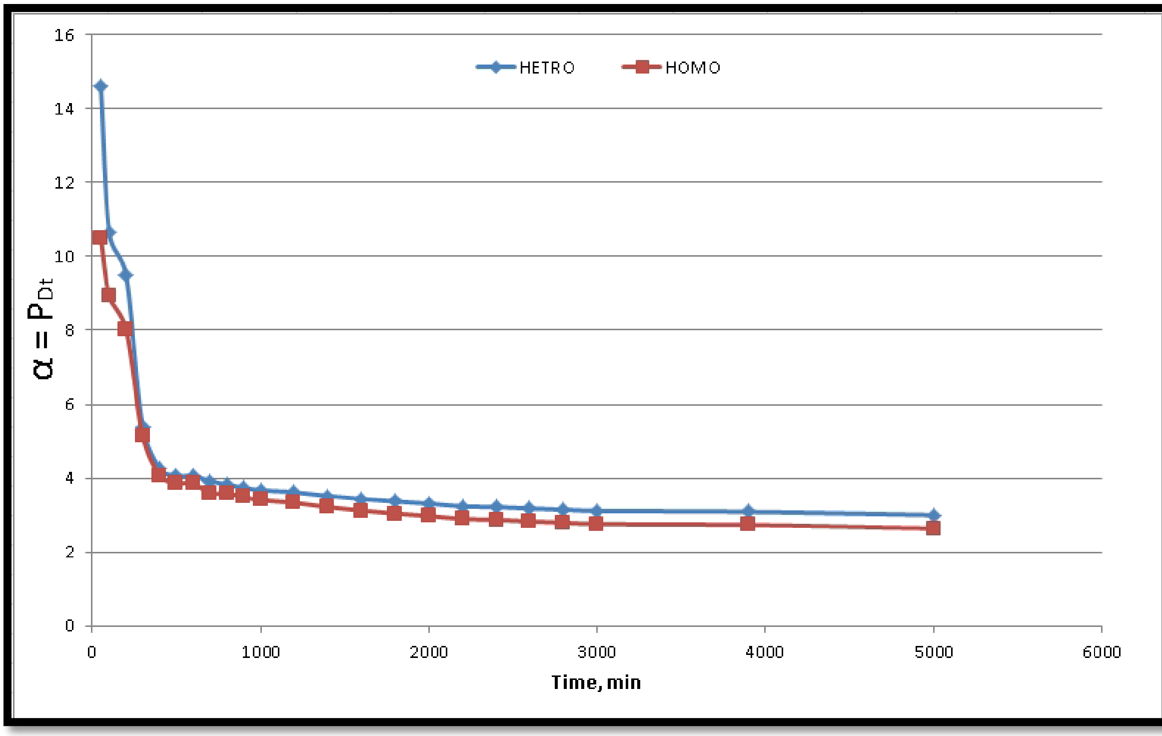

Figure 5.1.5. 5: Dimensionless pressure comparison, heterogeneous vs. homogenous

5.1.6. Fracture-Matrix Effective Permeability Ratio (k_f/k_m)

In any rock type, fractures above a certain aperture and length become preferential fluid flow paths. Aperture and fracture permeability are non-linearly related, but are correlated between fracture transmissibility and flow (Matthäi, 2005). Therefore, when the fracture flow is dominant, the fracture-matrix permeability ratio becomes an important factor and can have a significant impact on the overall reservoir performance. The accuracy of the dual-porosity system using Kazemi's shape factor model is tested with various fracture-matrix permeability ratios using the same 2D multi-set matrix blocks model as described earlier. The results are presented in Figure 5.1.6.1. The outcome from these tests indicates that the higher the effective fracture permeability, the better agreement between a dual-porosity model and the reference solution. Therefore, Kazemi's formulation might be appropriate for prediction of thermal recovery processes for the reservoirs consisting of massive fracture networks and big fracture apertures, which leads to very high fracture permeability. In this case, the assumption of the pseudo-steady state condition may be satisfied, which allows using a constant shape to count for the fluid exchange between matrix blocks and fractures.

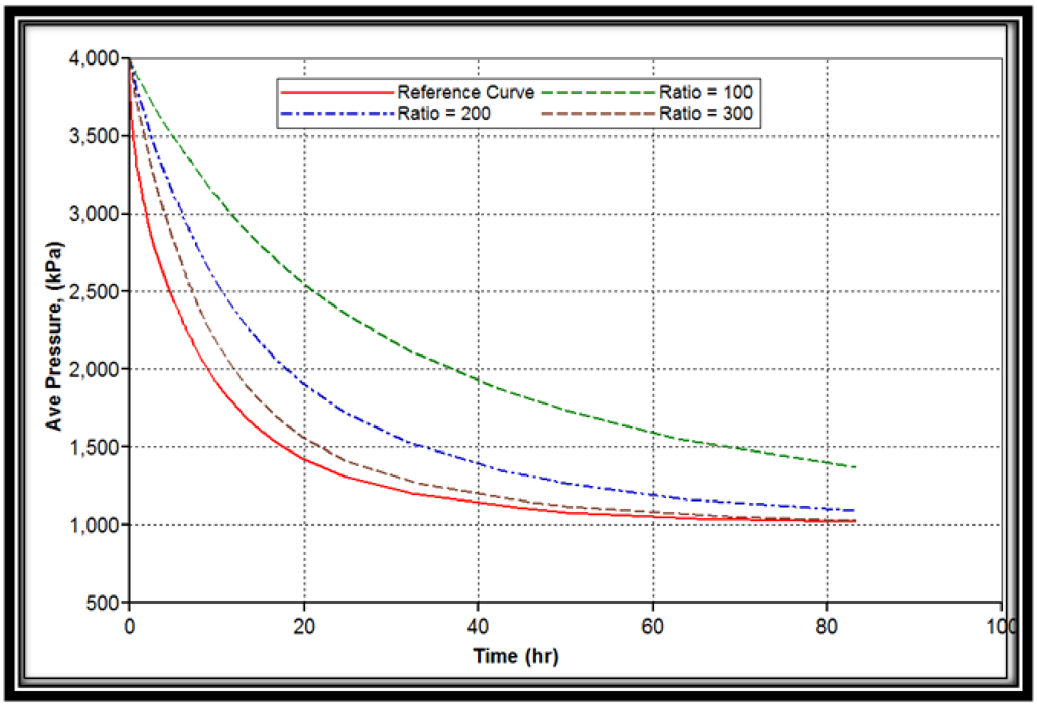

Figure 5.1.6. 1: Pressure profile, sensitivity of fracture-matrix permeability ratio (2D model)

5.2. Concluding Remarks

In this chapter, the effect of grid block size distribution, rock wettability, capillary pressure, initial water saturation, reservoir homogeneity, and a fracture-matrix effective permeability ratio are investigated using Kazemi's model. The results are compared with the proposed transient shape factor model against the reference solution case. The following conclusions are made from this study:

- The more complex and non-uniform matrix block sizes feature higher divergence from the reference solution, when utilizing constant shape factor formulations such as Kazemi's model.
- The results indicate that a transient shape factor can adjust Kazemi's formulation significantly in both mixed- and oil-wet cases.
- The dimensionless pressure coefficient factor ($\alpha = P_{Dt}$) versus time indicates that a relatively higher coefficient factor is required to modify Kazemi's formulation for the case with capillary pressure compared to the case where there is no capillary pressure.
- The results from the initial water saturation study indicate that for a system with lower water saturation values, using constant shape factor formulations leads to greater inaccuracy in prediction; therefore, applying the transient shape factor concept is recommended for accurate modeling of a dual-porosity system.

- The results demonstrate that, regardless of heterogeneity, it is recommended to utilize the transient shape factor approach for more precise predictions of modeling a dual-porosity system for thermal recovery processes.
- The outcome from the fracture-matrix effective permeability ratio work indicates that the higher the fracture permeability, the better agreement with the reference solution.
- Kazemi's formulation may be appropriate for prediction of thermal recovery processes in a reservoir with massive fracture networks. Hence the assumption of the pseudo-steady state condition may allow for use of a constant shape factor to count for the fluid exchange between matrix blocks and fractures.

CHAPTER 6

6. Model Validation

Model validation is an enabling methodology for the development of computational models that can be used to make engineering predictions with quantified confidence. Quantifying the confidence and predictive accuracy of model calculations provides the decision-maker with the information necessary for making high-consequence decisions. In order to check the reliability and consistency of the results of any particular model, capabilities of the model are required to be confirmed and validated with either a physical model or actual field data. The result of the above described transient shape factor (TSF) concept is validated against historical field data to check if this model is capable of reproducing accurate physical processes and recovery mechanisms. The historical thermal pilot data used in this work is from a full, twelve Cycle Steam Stimulation (CSS) process, which was conducted in a Grosmont carbonate formation back in 1980. This outcome clearly indicates that by implementing the TSF approach, significant improvements are achieved, showing a very good agreement with the actual historical field data from the CSS pilot.

6. 1. A Case Study

The Alberta Provincial Government funded pilot projects were administered by AOSTRA, the Alberta Oil Sand Research and Technology Authority. The work on the pilots was undertaken by Unocal at two principal sites referred to as Buffalo Creek and McLean, addressing Grosmont Formation carbonate in both cases. The Buffalo Creek 10A Pilot was commenced twelve cycles of the CSS (Cycle Steam Stimulation) operation in 1980 and its testing was continued until 1987.

In this section a complex geological model from the Grosmont formation, including a Discrete Fracture Network (DFN), is used to validate the transient shape factor concept (Figure 6.1.1). The historical field data from the Buffalo Creek 10A Pilot is used as a reference case to validate the newly proposed transient shape factor concept and compared the results with Kazemi's matrix-fracture shape factor formulation for a dual-porosity system. Figures 6.1.2 and 6.1.3 show the location of the Buffalo Creek pilot. The model was constructed and history matched using the commercial thermal simulator, CMG STARS, from Computer Modeling Group Ltd. Due to confidentiality, the model description and history matching are not presented here. Figures 6.1.4 through 6.1.6 show the results of the history match. Initially, the model is matched for the entire twelve cycles applying the default shape factor (Gilman and Kazemi, GK, 1988). Finally, the history match process is repeated using the transient shape factor approach (TSF). Figure 6.1.7 shows the history matched results using TSF against the conventional shape factor formulation (Gilman and Kazemi, 1988), while all the other parameters are kept the same. The dotted lines are the cumulative bitumen and water production and the solid lines are the simulation history matched outcome for both with and without applying the transient shape factor (TSF) model. The results clearly demonstrate that the history match is improved by utilizing the TSF formulation. This shows that in order to model a dual-medium system more accurately, a suitable transfer function is required, which can be modeled via an appropriate transient shape factor formulation.

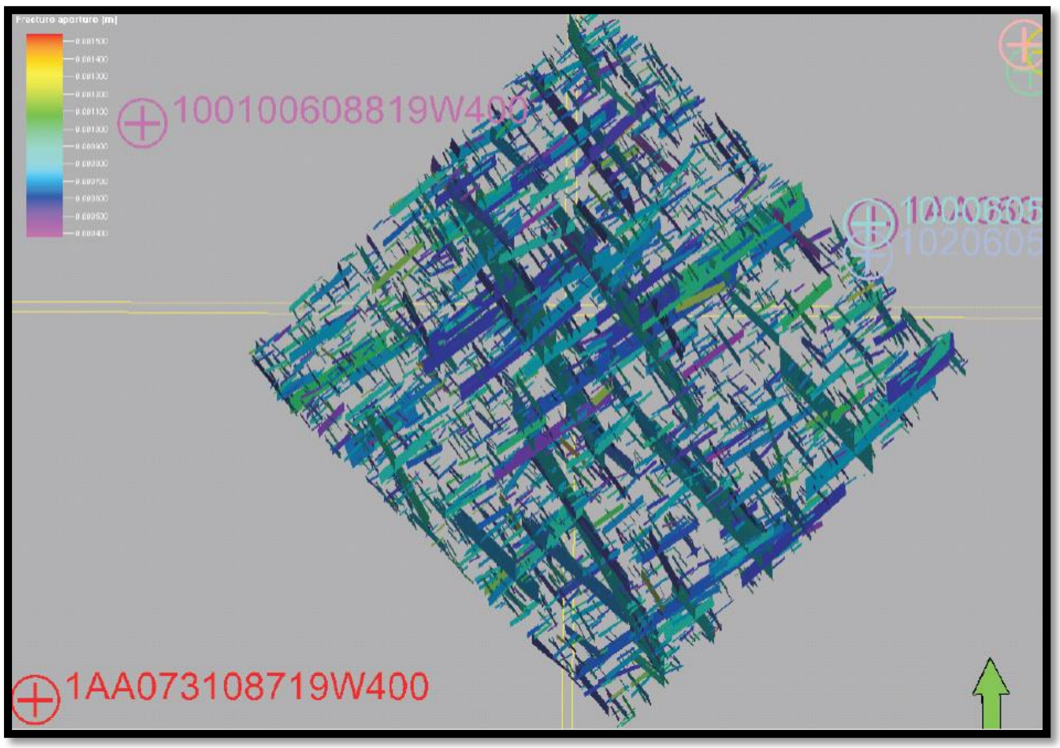

Figure 6.1. 1: Discrete fracture network (DFN) (Pilot application, Husky Energy, 2013)

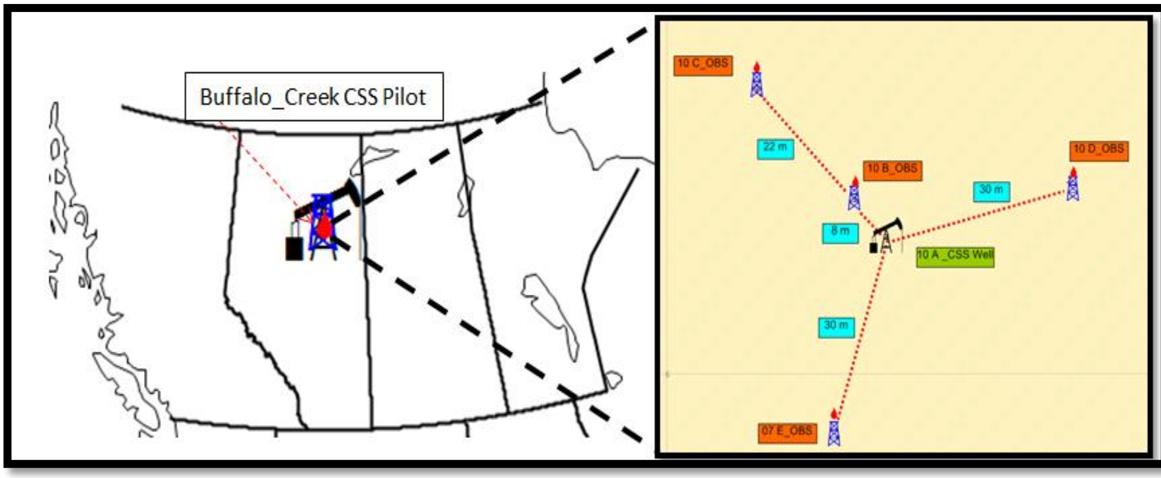

Figure 6.1. 2: Buffalo-Creek 10A Pilot location (Alberta, Canada)

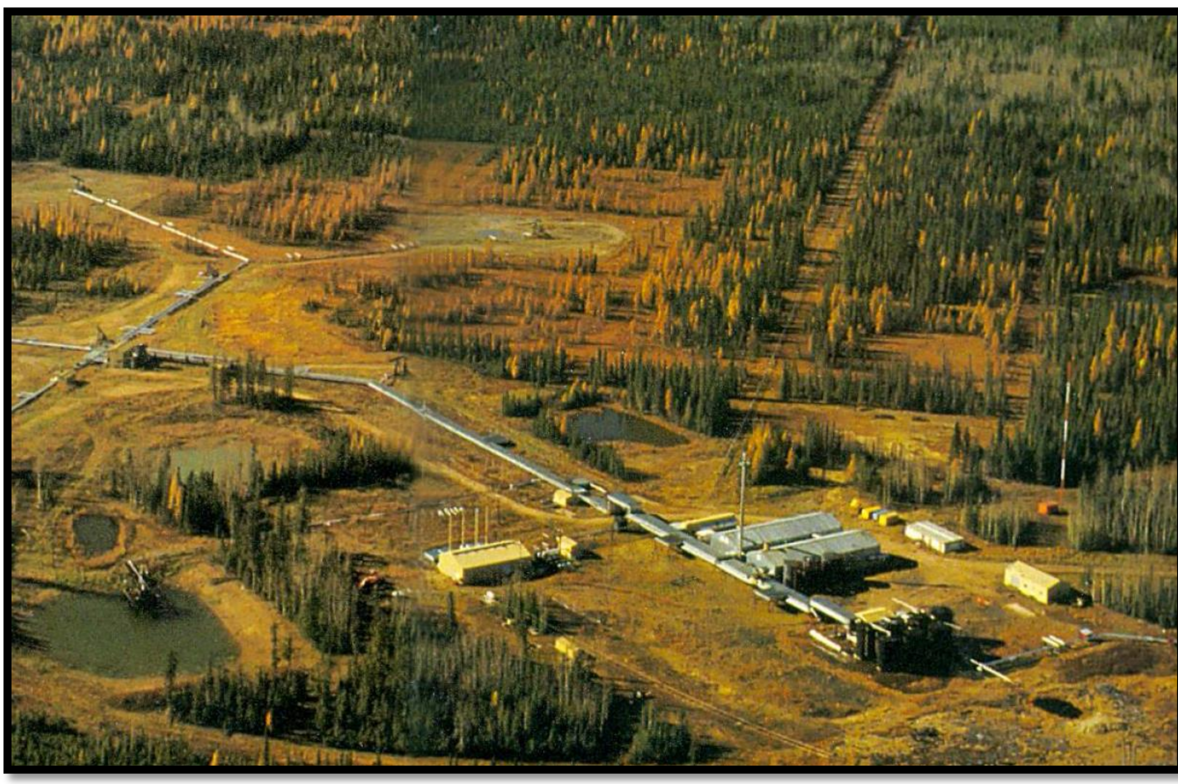

Figure 6.1. 3: Buffalo-Creek 10A Pilot facility (Norwest corporation report, 2008)

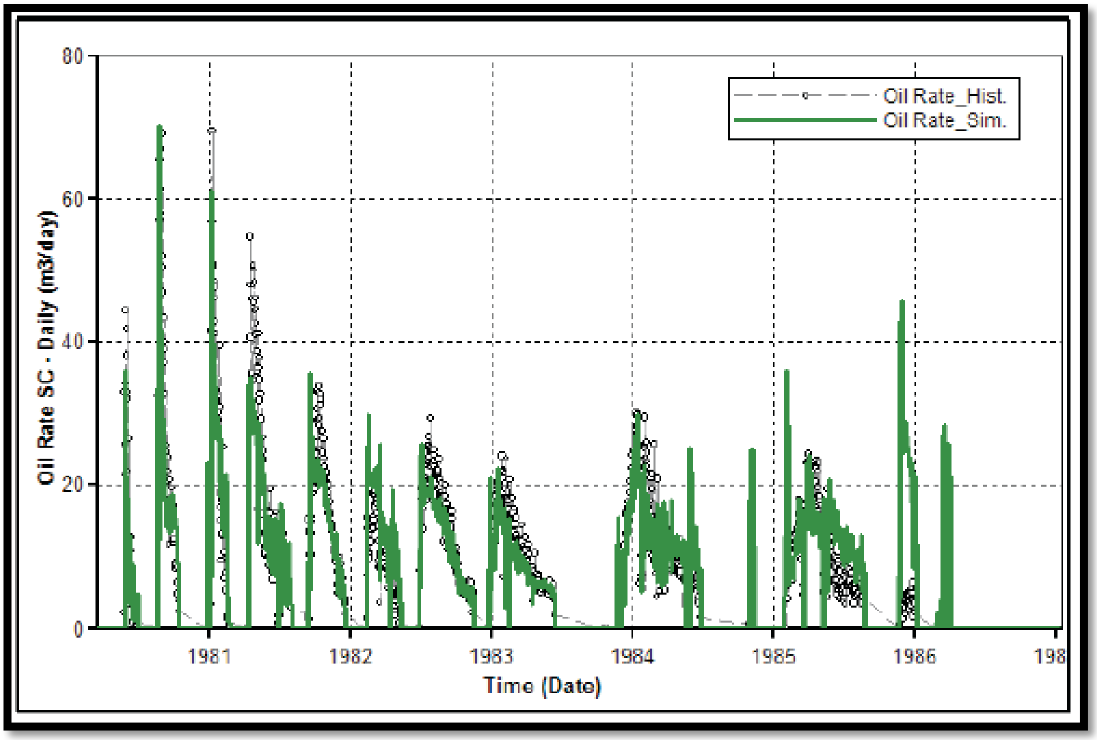

Figure 6.1. 4: Oil rate, history matched results (Pilot application, Husky Energy, 2013)

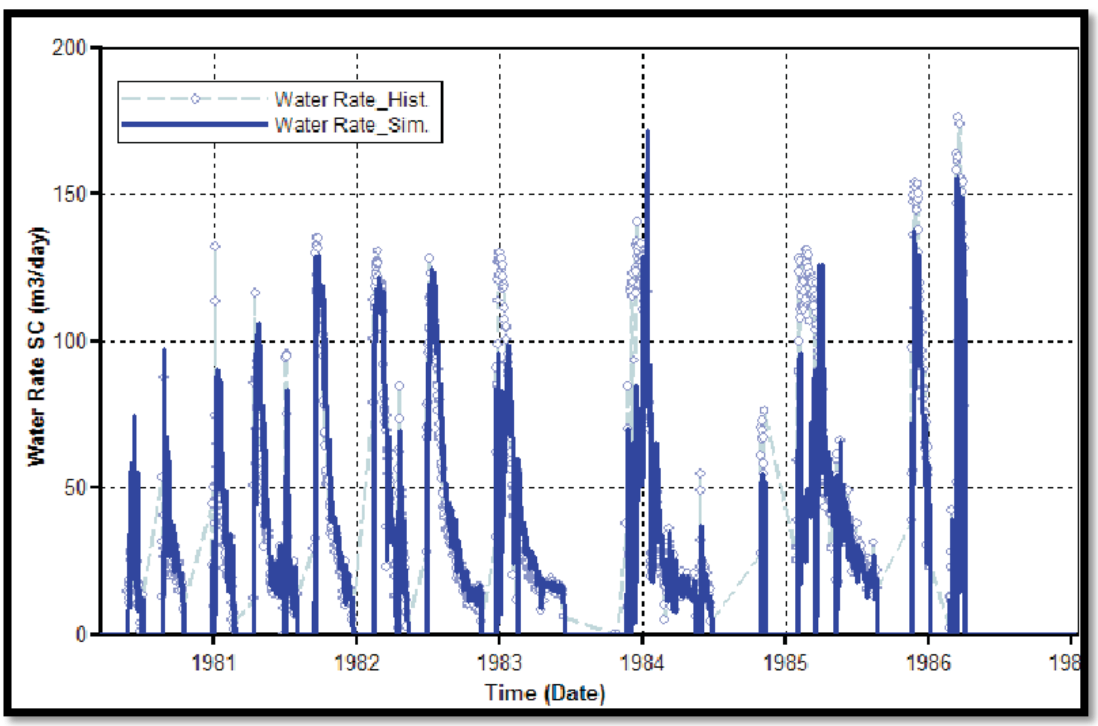

Figure 6.1. 5: Water rate, history matched results (Pilot application, Husky Energy, 2013)

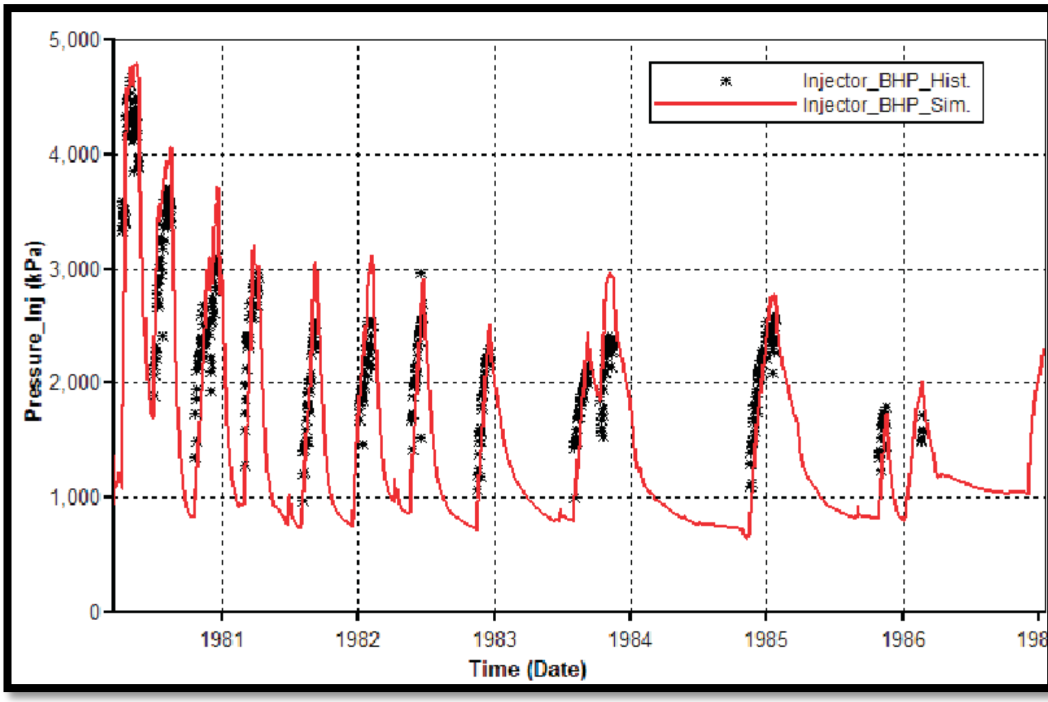

Figure 6.1. 6: Bottom hole injection pressure, history matched Results (Pilot application, Husky Energy, 2013)

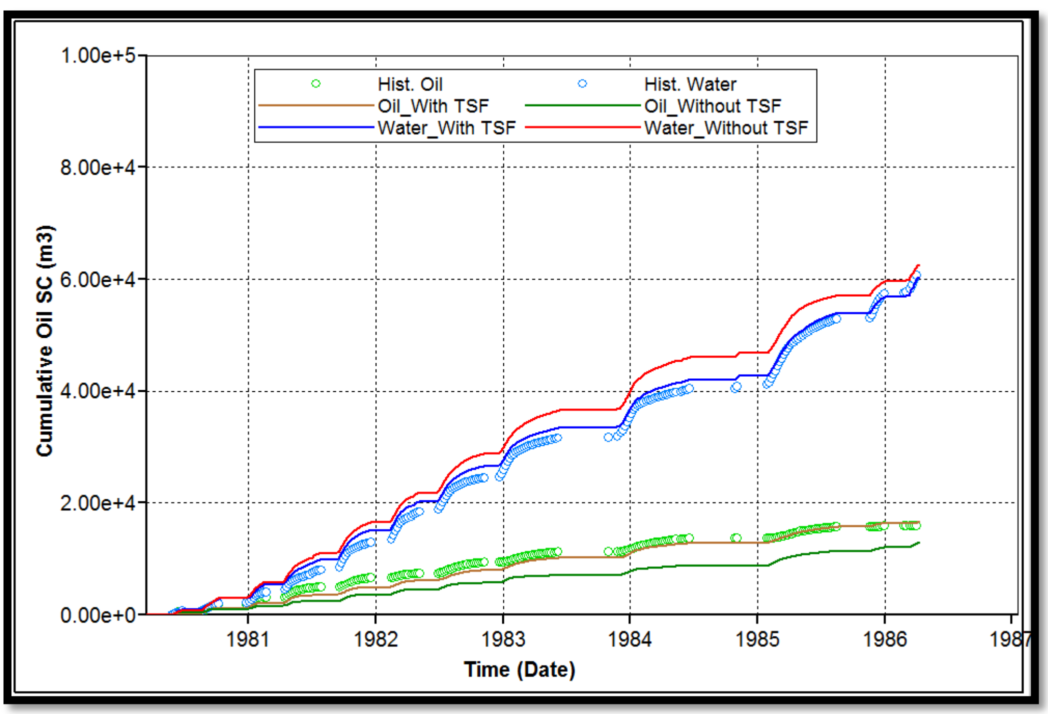

Figure 6.1. 7: Comparison of history matched results with and without TSF models

CHAPTER 7

Summary and Conclusions

This chapter will serve to summarize the results of the work described in this thesis as well as discuss the overall conclusions. Finally, it will conclude with suggestions for future studies, which should fill in some of the remaining gaps in presence for a better understanding of modeling shape factors of dual-media systems for thermal recovery processes.

7.1. Summary of Results

This research study is divided into three main sections. The first section is focused on the rate of exchange between the fractures and the matrix when the matrix blocks (fine-grid) are surrounded by fractures in a water-oil (bitumen) system, based on a single-porosity simulation model for a thermal recovery method. In the second section, a concept of a transient shape factor is introduced, which has led to the derivation of a new Thermal Transfer Function (TTF) and how to compute the transient shape factor (TSF) for non-isothermal recovery processes. The third section performs a series of sensitivity studies to identify some of the important factors that have the most significant impact on the deviation of a dual-medium model with the use of Kazemi's shape factor formulation from the reference solution.

The concept of the new Transient Shape Factor (TSF) is introduced in Chapter 4, where the new derivation of a transient shape factor for thermal reservoir simulation is proposed. A MATLAB code is developed and coupled with the CMG STARS to compute the transient shape factor (TSF). In Chapter 5, the important factors that have the largest impact on the deviation of the dual-medium model with the use of Kazemi's shape factor formulation are determined. Chapter 6 covers the model validation, using the historical field data of an old CSS pilot in a Grosmont carbonate.

The following summary and conclusions are made based on this study:

- There was an insufficient understanding of an adequate matrix-fracture transfer shape factor formulation in a dual-porosity system for thermal recovery mechanisms.
- Most of the existing transfer function formulations (matrix-fracture interaction) were based on the assumption of orthogonal fracture systems and pseudo-steady state flow, which are not representative of actual reservoir recovery mechanisms.
- All the suggested analytical solutions were too complicated to be incorporated directly into current commercial simulators.
- Overall, the results demonstrate that none of the existing dual-porosity models can be used to generate a consistent match at all times with the reference solution; particularly, Kazemi's formulation shows the maximum deviation from the reference solution (fine grid single-porosity).
- A new transient shape factor (TSF) for non-isothermal, dual-porosity models is introduced and compared with the existing shape factor models. The results

clearly show that a transient shape factor is required for an appropriate modeling of a thermal recovery process in NFRs when using dual-porosity formulations.

- The results clearly demonstrate that a constant value of the shape factor (e.g., Kazemi's model) cannot be used to predict the overall performance in dual-media systems due to invalid assumptions.
- A MATLAB code is developed and coupled with the CMG STARS thermal reservoir simulator to compute the transient shape factor (TSF Estimator tool).
- The results indicate that during the early time the pressure drop is steeper for the reference solution compared to the dual-porosity model using the existing shape factors, which suggests that all the current shape factors are smaller than that required.
- The transient shape factor estimated is based on Kazemi's formulation; however, this methodology can be used to investigate other shape factor formulations and estimate the transient shape factors.
- More complex and non-uniform matrix block sizes feature higher divergence from the reference solution, when utilizing constant shape factor formulations such as Kazemi's model.
- The results indicate that a transient shape factor can correct Kazemi's formulation significantly in both mixed- and oil-wet cases.
- The dimensionless pressure coefficient factor ($\alpha = P_{Dt}$) versus time indicates that a relatively higher coefficient factor is required to correct Kazemi's formulation for the case with capillary pressure compared to the case where there is no capillary pressure.

- The results from the initial water saturation study indicates that for a system with lower water saturation values, using constant shape factor formulations leads to grater inaccuracy in prediction; therefore, applying the transient shape factor concept is recommended for accurate modeling of a dual-porosity system.
- The results demonstrate that due to heterogeneous nature of carbonate reservoirs, Kazemi's formulation requires a higher coefficient factor, mainly in the early time, to convert to the transient shape factor with less inaccuracy of outcomes.
- The outcome from the fracture-matrix effective permeability ratio work indicates that the higher the fracture permeability, the better agreement with the reference solution.
- Kazemi's formulation may be appropriate for prediction of thermal recovery processes in a reservoir with massive fracture networks. Hence the assumption of the pseudo-steady state condition may allow the use of a constant shape to account for the fluid exchange between matrix blocks and fractures.

7.2. Recommendations for Future Studies

There are still several gaps within the present understanding of modeling dual-media systems. The current numerical simulation techniques of fractured reservoirs are far from complete and there are many areas for improvement. Based on the work presented in this thesis, it is clear that much more work remains to be accomplished regarding a shape factor function. Nevertheless, the following studies are recommended for future investigation on a transient shape factor for improving the fluid modeling in dual-media systems:

- The transient shape factor estimated in this study is focused only on Kazemi's formulation. It is recommended to investigate this transient shape factor concept for other existing shape factor models.
- In this study, the flow transient shape factor is examined only. Thermal transient shape factor is recommended to be investigated.
- The proposed model has been evaluated using 2D numerical simulation; this work can be extended to utilizing a complex 3D model.
- The transient shape factor concept can be examined using a complex fracture model such as Discrete Fracture Network (DFN).
- This method can be tested and validated either using other historical field data or validated against an experimental work for any recovery mechanisms in naturally fracture reservoirs.

- A detailed sensitivity analysis is recommended using a commercial simulator such as CMOST for better evaluating the impact of reservoir parameters for the existing shape factor formulations as well as the proposed transient shape factor model.
- This study is focused on thermal recovery processes; the similar technique can be implemented to evaluate other recovery mechanisms in naturally fractured reservoirs.
- Improve the “TSF Estimator” tool kit to be more robust and user-friendly.
- This study is focused on dual-porosity model; it is recommended the concept of the transient shape factor in dual-permeability system to be examined.

REFERENCES

- Ahmad, S.A, and Abushaikha, A. "Matrix-Fracture Transfer Function in Dual-Medium Flow Simulation: Review, comparison, and Validation." SPE 113890, presented at the Europec/EAGE annual conference, Rome, Italy, 2008.
- Al-Kobaisi, M., Kazemi, H., Ramirez, B., Ozkan, E., and Atan, S., 2009, A Critical Review for Proper Use of Water/Oil/Gas Transfer Functions in Dual-Porosity Naturally Fractured Reservoirs: Part II: SPE Res Eval & Eng, v. 12, no. 2, p. 211-217.
- A.P.G van Heel, P.M Boerrigter, and J.J van Drop "Thermal and Hydraulic Matrix-Fracture Intraction in Dual-Permeability Simulation." SPE 102471, Annual Technical Conference and Exhibition, San Antonio, Texas, 2008.
- Bahrami. N and Pena. D et al., "Characterizing Dual-Porosity Stimulated Rock Volume in Multi-Fractured Shale Oil and Gas Reservoirs: Field Example from Eagle Ford Shale". SPE Reservoir Characterisation and Simulation Conference and Exhibition, Abu Dhabi, UAE. 2015.
- Barenblatt, G. I., Iu, P. Z., and Kochina, I. N., 1960, Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks [strata]: PMM vol.24, no.5, 1960, pp. 852-864 (Russian), v. 24, p. 1286-1303.
- Butler, R. "Thermal Recovery of Oil and Bitumen." Text book, 1997.
- Barker, J.A. (1985). Block-geometry functions characterizing transport in densely fissured, media. *J. Hydrol.*, 77, 263-279.

- Coast, K.H. “Implicit Compositional Simulation of Single-Porosity and Dual-Porosity Reservoirs.” SPE 18427, presented at Symposium in Reservoir Simulation, Houston, TX, USA, 1989.
- Chen, J. “Investigations of Matrix-Fracture Transfer Flows in Dual-Porosity Modeling of Naturally Fractured Reservoirs.” SPE 29562, presented at the Reservoir Symposium, Denver, CO, USA, 1995.
- Chen, Z. “Reservoir simulation, Mathematical Techniques in Oil Recovery.” Text book, 2007.
- Civan, F. and Rasmussen, M.L. (2002). Analytical hindered-matrix-fracture transfer models for naturally fractured petroleum reservoirs, Paper SPE 74364.
- Chang, M.-M., 1993, “Deriving the shape factor of a fractured rock matrix”. National Inst. for Petroleum and Energy Research, Bartlesville, OK (United States).
- Dean, R., and Lo, L., 1988, Simulations of naturally fractured reservoirs: SPE Reservoir Engineering, v. 3, no. 02, p. 638-648.
- Dernaika. M, and Naseer. Y et. al. “Multi-Scale Rock Analysis for Improved Characterization of Complex Carbonates”. SPE Reservoir Characterization and Simulation Conference and Exhibition, Abu Dhabi, UAE, 2015.
- Farouq-Ali, S.M. “Practical Heavy Oil Recovery.” November, 2008.
- F. S. Kadhim, and A. Samauri, “A Review in Correlations between Cementation Factor and Carbonate Rocks Properties”, University Teknologi Malaysia and University of Technology, Baghdad, Iraq, Life Science Journal 2013.
- Gotfried, B.S., “A Mathematical Model of Thermal Oil Recovery in Linear Systems.” SPE 1117-PA, 1965, Volume 5, 196-210.

- Gilman, J.R. and Kazemi, H. "Improvement in Simulation of Naturally Fractured Reservoirs", SPE 10511-PA, Volume 23, page 695-707, 1983.
- Gilman, J.R. "Practical Aspects of Simulation of Fractures Reservoirs." International Forum on Reservoir Simulation, USA, 2003.
- George V. Chilingar and T. F. Yen, "Energy Sources" Text book, 2007.
- Gerke, H.H. van Genuchten, M.Th. (1993). A dual-porosity model for simulating the preferential movement of water and solutes in structured porous media, *Water Res.,* 29(2), 305-319.
- Gilman, J. R., and Kazemi, H., 1988, Improved Calculations for Viscous and Gravity Displacement in Matrix Blocks in Dual-Porosity Simulators: Journal of Petroleum Technology, v. 40, no. 1, p. 60-70.
- Gurpinar, O. M., and Kossack, C. A., 2000, Realistic Numerical Models for Fractured Reservoirs: SPE Journal, v. 5, no. 4, p. 485-491.
- Hassanzadeh, H. and Pooladi-Darvish, M. "Effect of Fracture Boundary Conditions on Matrix-Fracture Transfer Shape Factor." Journal of Transport in Porous Media, 2006, page 51-71.
- H.S. Al-Hadhrami (Petroleum Development Oman), Martin J. Blunt (Imperial College), "Thermally Induced Wettability Alteration To Improve Oil Recovery in Fractured Reservoirs", SPE-71866-PA, 2001.
- Hassanzadeh, H. and Pooladi-Darvish, M. Atabay, S. (2009). Shape factor in the drawdown solution for well testing of dual-porosity systems, *Adv. Water Res.,* 32(11), 1652–1663.

- Khan, F.U. “Mathematical Simulation of Horizontal Well Performance in Naturally Fractured Rock Formations.” Texas Tech University, 1998.
- Kazemi, H., Merrill, L.S. and Zeman, P.R. “Numerical Simulation of Water-Oil Flow in Naturally Fractured Reservoirs.” SPE 5719 presented at SPE Fourth Symposium, Los Angeles, CA, USA, 1976.

  Kazemi, H. Merrill, L.S. Porterfield, and K.L. Zeman, P.R. (1976). Numerical simulation of water-oil flow in naturally fractured reservoirs. *Soc. Pet. Eng. J.,* 16(6), 317-326.
- Lu, H. and Blunt, M.J. “General Fracture/Matrix Transfer Function for Mixed-Wet System.” SPE 107007, presented at Europec/EAGE Annual Conference and Exhibition, London, UK, 2007.
- Lu, H. and Di Donato, G. “General Transfer Functions for Multiphase flow.” SPE 102542, presented at the Annual Technical Conference and Exhibition, San Antonio, CA, USA, 2006.
- Lim, K.T. and Aziz, K. “Matrix-Fracture Transfer Shape Factor for Dual-Porosity Simulations.” Journal of Petroleum Science and Engineering, 1994.
- Litvak, B. L.: "Simulation and Characterization of Naturally Fractured Reservoirs," Proc., Reservoir Characterization Technical Conference, Dallas, Academic Press, New York, (1985) 561-83.
- Lim, K.T. and Aziz, K. (1995). Matrix-fracture transfer shape factors for dual-porosity simulators. *J. Pet. Sci. Eng.,* 13, 169-178.
- Lim, K., and Aziz, K., 1995, Matrix-fracture transfer shape factors for dual-porosity simulators: Journal of Petroleum Science and Engineering, v. 13, no. 3, p. 169-178.

- Lu, H., DiDonato, G., and Blunt, M. J., 2008, General Transfer Functions for Multiphase Flow in Fractured Reservoirs: SPE Journal, v. 13, no. 3, p. pp. 289-297.
- Mohammadi, S., Ehsani, M.R. and Nikookar, M. "Study of Steam Injection in a Fractured Carbonate Heavy Oil Reservoir in Iran." SPE 144943, presented at the SPE Heavy Oil Conference, Calgary, AB, Canada, 2012.
- Mourzenko, V.V. Bogdanov, I.I. and Thovert, J.F. Adler, P.M. (2011). Three-dimensional numerical simulation of single-phase transient compressible flows and well-tests in fractured formations. *Math. Comput. Simul*., 81, 2270–2281.
- O.P Wennberg, and G.Casini, "The Characteristics of Open Fractures in Carbonate Reservoir Rocks", May 2011.
- Penuela, G., Hughes, R.G. and Civan, F. "Time-Dependent Shape Factor for Secondary Recovery in Naturally Fractured Reservoirs." SPE 75234, presented at the Improved Oil Recovery Symposium, Tulsa, OK, USA, 2002.
- Paul M Harris, "Heterogeneity within Carbonate Reservoirs". University of California Davis, April 30, 2004.
- Penuela, G. Civan, F. Hughes, R.G. and Wiggins, M.L. (2002). Time-dependent shape factors for interporosity flow in naturally fractured gas-condensate reservoirs. SPE Paper 75524.
- Penuela, G., Hughes, R., Civan, F., and Wiggins, M., Time-dependent shape factors for secondary recovery in naturally fractured reservoirs - SPE 75234, SPE/DOE Improved Oil Recovery Symposium, 2002.

- Rangle-German, E.R. and Kovscek, R. “Time-Dependent Matrix-Fracture Shape Factors for Partially and Completely Immersed Fractures.” SPE 84411, annual technical conference and exhibition, Denver, CO, USA, 2003.
- Ranjbar, E. and Hassanzadeh, H. (2011). Matrix-fracture transfer shape factor for modeling flow of a compressible fluid in dual-porosity media. *Adv. Water Res.*, 34(5), 627-639.
- Rangel-German, E. R., and Kovscek, A. R., 2006, Time-dependent matrix-fracture shape factors for partially and completely immersed fractures: Journal of Petroleum Science and Engineering, v. 54, no. 3, p. 149-163.
- Shutler, N.D. “Numerical, Three-Phase Simulation of the Linear Steamflood Process.” SPE 2233, presented at Annual Fall Meeting, Houston, TX, USA, 1969.
- Sarma, P. “New Transfer Function for Simulation of Naturally Fractured Reservoirs with DualPorosity Models.” Stanford University, 2003.
- S.K. Mattai, A. and Mezentsev, M.Belayneh. Control-Volume Finite-Element Two-Phase Flow Experiments with Fractured Rock Represented by Unstructured 3D Hybrid Meshes. SPE 93341-MS, SPE Reservoir Simulation Symposium, 31 January-2 February, The Woodlands, Texas.
- Thomas, L. K., Dixon, T. N., and Pierson, R. G., 1983, Fractured Reservoir Simulation, SPE Journal, v. 23, no. 1, p. 42-54.
- Van Dijkum, C. E., and Walker, T., 1991, Fractured Reservoir Simulation and Field Development, Natih Field, Oman. SPE 22917, SPE Annual Technical Conference and Exhibition: Dallas, Texas, 1991, Society of Petroleum Engineers.

- Van Heel, A.P.G., van Drop, J.J. and Boerrigter, P. M. “Thermal and Hydraulic Matrix-Fracture Interaction in Dual-Permeability Simulation”. SPE 102471, presented at the Annual Technical Conference and Exhibition, Texas, USA, 2007.
- Van Heel, A.P.G., van Drop, J.J. and Boerrigter, P. M. “Heavy-Oil Recovery by Steam Injection in Fractures Reservoirs.” SPE 113461, presented at the Improved Oil Recovery Symposium, Tulsa, OK, USA, 2008.
- Ueda, Y., Murata, S., and Watanabe, Y. “Investigation of the Shape Factor Used in the Dual-Porosity Reservoir Simulation.” SPE 19469, presented at the SPE Asia-Pacific Conference, Sydney, Australia, 1989.
- Warren, J.E. and Root, P.J. “The Behavior of Naturally Fractured Reservoirs.” Journal of Society of Petroleum Engineers, 1963, page 245-255.
- Wuthicharn, K. and Zimmerman, R.W. (2011). Shape factors for irregularly matrix blocks, SPE Paper 148060.
- Zimmerman, R.W. and Bodvarsson, G.S. (1995). Effective block size for imbibitions or absorption in dual-porosity media. *Geophys. Res. Lett., 22(11),* 1461-1464.

Appendix A: Mass and Heat Equations

Flow Equations:

Mass balance in the matrix blocks for any phase in form of finite differences can be expressed as follows:

$$\nabla\left[T_{\alpha m}\left(\nabla P_{\alpha m}-\gamma_{\alpha m}\nabla Z\right)-T_{\alpha mf}\left(P_{\alpha m}-P_{\alpha f}\right)\right]={}^{V_b}\!/_{\Delta t}\left(\frac{\phi_m S_{\alpha m}}{\beta_{\alpha m}}\right) \quad \text{(A.1)}$$

$$\nabla\left[T_{\alpha f}\left(\nabla P_{\alpha f}-\gamma_{\alpha f}\nabla Z\right)+T_{\alpha mf}\left(P_{\alpha m}-P_{\alpha f}\right)\right]+q_{\alpha f}={}^{V_b}\!/_{\Delta t}\left(\frac{\phi_f\ S_{\alpha f}}{\beta_{\alpha f}}\right) \quad \text{(A.2)}$$

where α = water, oil, or steam and the subscripts *m* and *f* refer to the matrix and the fractures, respectively.

Flow transmissibility involves a geometric shape factor, which is defined as

$$T_{\alpha mf}=\sigma_F\ .\ K_m\ {}^{K_{r\alpha}}\!/_{\mu_\alpha .\beta_\alpha} \quad \text{(A.3)}$$

$$\mu_\alpha = f\left(Temp\right) \quad \text{(A. 4)}$$

Therefore, the flow transfer function can be shown as (Kazemi and Gilman, 1976)

$$\Im_{\alpha mf}=T_{\alpha mf}\ \left[\ \left(P_{\alpha f}-\gamma_{\alpha f}Z\right)-\left(P_{\alpha m}-\gamma_{\alpha m}Z\right)\ \right] \quad \text{(A.5)}$$

Heat Equations:

For the heat transfer equation, it is assumed that thermal equilibrium between the fluid and the rock matrix is instantaneously achieved (Shutler, 1969). The energy balance law is expressed as

$$\textit{Conduction H.T} + \textit{Convection H.T} + \textit{Heat Source} - \textit{Heat Loss} = \textit{Accumulation (Pore + Rock)}$$

where;

Conduction H.T = $\nabla K_h . \nabla T$

Convection H.T = $\nabla \left[\sum \rho_\alpha v_\alpha H_\alpha \right]$

Heat source (q_H) = $\sum \rho_\alpha \, q_\alpha \, H_\alpha$

Heat loss (q_L) = $-K_h A \, \partial T / \partial x$

Accumulation (Pore + Rock) = $\frac{\partial}{\partial t}\left[\left(\phi \sum \rho_\alpha H_\alpha S_\alpha \right)_\varepsilon + (1-\phi_i)\rho_r C_r T \right]$

α = water, oil, or steam, ε= matrix or fracture, H = enthalpy, and the subscript *r* refers to the reservoir rock (assuming no convection in the matrix and no conduction in the fractures).

$$\nabla K_h . \nabla T - T^{\circ}{}_{\alpha mf}\left(T^{*}_{\alpha f} - T^{*}_{\alpha m} \right) = \frac{\partial}{\partial t}\left[\left(\phi \sum \rho_\alpha H_\alpha S_\alpha \right)_m + (1-\phi_m)\rho_r C_r T \right] \quad \text{(A.6)}$$

$$\nabla\left(\sum \rho_{\alpha f} \, K_f / \mu_{\alpha f} \, H_{\alpha f}\left(\nabla P_{\alpha f} - \gamma_{\alpha f} \, \nabla Z \right) \right) + q_{Hf} + q_{Lf} + T^{\circ}{}_{\alpha mf}\left(T_{\alpha f} - T_{\alpha m} \right) = \frac{\partial}{\partial t}\left[\left(\phi \sum \rho_\alpha H_\alpha S_\alpha \right)_f \right]$$

(A.7)

In thermal modeling, the matrix-fracture interactions involve both fluid exchange and heat exchange (van Heel et al., 2008). Therefore, the heat or energy transfer function can be described as:

$$\Im^{\circ}{}_{\alpha mf} = T^{\circ}{}_{\alpha mf} \left(T^{*}_{\alpha f} - T^{*}_{\alpha m} \right) \quad \text{(A.8)}$$

The heat or energy transfer transmissibility can be defined as follow. However, this study focused on the flow transient shape only and thermal transient shape factor is recommended to be investigated by describing a suitable thermal transmissibility factor.

$$T^{\circ}{}_{\alpha mf} = \sigma_T \cdot \frac{K_h}{\rho C} \quad \text{(A.9)}$$

where σ_T can be defined as a thermal shape factor, and α = water, oil, or steam.

To compute a flow transient shape factor, a dimensionless average pressure P_{D_t} is introduced as:

$$P_{D_t} = \left[\frac{P_f}{P_m - P_i} \right]_t^{\omega_t} \quad \text{(A.9)}$$

$$P_{D_t} = \alpha \quad \text{(A.10)}$$

ω_t is an exponent factor which is a time-dependent factor and can be expressed as a ratio of the matrix pressure and the fracture pressure at each time step:

$$\omega_t = \psi \left(\frac{P_m}{P_f} \right)_t \tag{A.11}$$

Ψ is a coefficient factor, which depends on the rock wettability ranging from 1.25 to 3.75.

Flow transient shape factor, $\sigma_{F(t)}$:

$$\sigma_{F(t)} = 4 \sum \left(1/L_i^2 \right) . P_{D_t} \tag{A.12}$$

Thermal transient shape factor, $\sigma_{T(t)}$:

$$\sigma_{T(t)} = T^{\circ n}{}_{\alpha mf} . \frac{\rho_\alpha C_\alpha}{K_{\alpha h}} \tag{A.13}$$

Thermal Transient Transfer function:

$$T.T.F = 4 \sum_{k=1}^{nd} 1/L_k^2 . \left(\frac{P_{fi}}{P_{mi} - P_i} \right)^{\omega t} . K . K_r / \mu \ (P_m - P_f) \tag{A.14}$$

Appendix B: MATLAB Code

```
% Coupling CMG STARS to Calculate the TSF
% No_Accum_TSF_Kazemi_Coupling_w_Vari_Time_Step_Run

% T1

% ===================================
%Time Step RUNS...!....Variable w
% ===================================


status1 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

%
======================================================================
=

% w =(Pm/Pf)t
a=1.45;
%a=1.42;
%a=1.38;
%a=1.36;
%a=1.37;
       w=0.656;
%
======================================================================
=
fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));

 ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F0 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F0(2:45,3) = cell2mat(E);
```

```
%
F0(:,4)=(F0(:,1)./(F0(:,2)-F0(1,1))).^(w);

F0(4,5)=(F0(4,4)+F0(5,4))./2;
Coff0 = F0(4,5);

fid = fopen('Coff.txt','w', 'n');

 %fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff0);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff0);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff0);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff0);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff0);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff0);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 50') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
```

```
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

 status2 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));

ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F1 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F1(2:45,3) = cell2mat(E);

F1(:,6)=a*((F1(:,2)./(F1(:,1))));

i = 100;
Time= F1(:,3)==i;
w = F1(Time, 6);
F1(:,4)=(F1(:,1)./(F1(:,2)-F1(1,1))).^(w);

Coff1 = F1(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',Coff1);
fclose(fid);
 fid = fopen('Coff.txt','a+');
```

```
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff1);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff1);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff1);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff1);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff1);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff1);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 100') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T2
```

```
status3 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);


fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));

ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F2 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F2(2:45,3) = cell2mat(E);

F2(:,6)=a*((F2(:,2)./(F2(:,1))));

j = 2*i;
Time= F2(:,3)==j;
w = F2(Time, 6);
F2(:,4)=(F2(:,1)./(F2(:,2)-F2(1,1))).^(w);

Coff2 = F2(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',Coff2);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff2);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
```

```
fprintf(fid,'%f\n',Coff2);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff2);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff2);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff2);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff2);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 200') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T3
status4 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');
```

```
C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));

ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F3 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F3(2:45,3) = cell2mat(E);
F3(:,6)=a*((F3(:,2)./(F3(:,1))));

j = 2*i+i;
Time= F3(:,3)==j;
w = F3(Time, 6);
F3(:,4)=(F3(:,1)./(F3(:,2)-F3(1,1))).^(w);
Coff3 = F3(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',Coff3);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff3);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff3);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff3);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
```

```
fprintf(fid,'%f\n',Coff3);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff3);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff3);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 300') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);


%T4
status5 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));

ind = A;
B=C(ind);
```

```
D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F4 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F4(2:45,3) = cell2mat(E);

F4(:,6)=a*((F4(:,2)./(F4(:,1))));

j = 3*i+i;
Time= F4(:,3)==j;
w = F4(Time, 6);
F4(:,4)=(F4(:,1)./(F4(:,2)-F4(1,1))).^(w);
Coff4 = F4(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',Coff4);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff4);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff4);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff4);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff4);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff4);
fclose(fid);
```

```
fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff4);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 400') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);


%T5
status6 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));

ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F5 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
```

```
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F5(2:45,3) = cell2mat(E);
F5(:,6)=a*((F5(:,2)./(F5(:,1))));

j = 4*i+i;
Time= F5(:,3)==j;
w = F5(Time, 6);
F5(:,4)=(F5(:,1)./(F5(:,2)-F5(1,1))).^(w);

Coff5 = F5(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',Coff5);
fclose(fid);

% fid = fopen('Coff.txt','a+');
% fprintf(fid,'*TRANSMULT *ACCUMULATE  ');
% fprintf(fid,'%f\n','');
% fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff5);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff5);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff5);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff5);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff5);
fclose(fid);

fid = fopen('Coff.txt','a+');
```

```
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff5);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 500') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T6

status7 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F6 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);
```

```
E=cellfun(@str2num,D,'un',0);

F6(2:45,3) = cell2mat(E);

F6(:,6)=a*((F6(:,2)./(F6(:,1))));

j = 5*i+i;
Time= F6(:,3)==j;
w = F6(Time, 6);
F6(:,4)=(F6(:,1)./(F6(:,2)-F6(1,1))).^(w);
Coff6 = F6(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',Coff6);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff6);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff6);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff6);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff6);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff6);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff6);
fclose(fid);


%# read & write to other file
```

```
bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 600') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T7

status8 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F7 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F7(2:45,3) = cell2mat(E);
```

```
F7(:,6)=a*((F7(:,2)./(F7(:,1))));

j = 6*i+i;
Time= F7(:,3)==j;
w = F7(Time, 6);
F7(:,4)=(F7(:,1)./(F7(:,2)-F7(1,1))).^(w);
Coff7 = F7(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',Coff7);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff7);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff7);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff7);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff7);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff7);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',Coff7);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
```

```
while ~strcmp(l, 'TIME 700') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);


%T8

status9 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F8 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F8(2:45,3) = cell2mat(E);

F8(:,6)=a*((F8(:,2)./(F8(:,1))));

j = 7*i+i;
Time= F8(:,3)==j;
w = F8(Time, 6);
F8(:,4)=(F8(:,1)./(F8(:,2)-F8(1,1))).^(w);
CofF8 = F8(Time, 4);
```

```
fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF8);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF8);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF8);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF8);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF8);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF8);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF8);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 800') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
```

```
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T9

status10 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F9 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F9(2:45,3) = cell2mat(E);
F9(:,6)=a*((F9(:,2)./(F9(:,1))));

j = 8*i+i;
Time= F9(:,3)==j;
w = F9(Time, 6);
F9(:,4)=(F9(:,1)./(F9(:,2)-F9(1,1))).^(w);
CofF9 = F9(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF9);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF9);
fclose(fid);

fid = fopen('Coff.txt','a+');
```

```
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF9);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF9);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF9);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF9);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF9);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 900') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T10

status11 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);
```

```
fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));


ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F10 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F10(2:45,3) = cell2mat(E);

F10(:,6)=a*((F10(:,2)./(F10(:,1))));


j = 9*i+i;
Time= F10(:,3)==j;
w = F10(Time, 6);
F10(:,4)=(F10(:,1)./(F10(:,2)-F10(1,1))).^(w);
CofF10 = F10(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF10);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF10);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF10);
fclose(fid);

fid = fopen('Coff.txt','a+');
```

```
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF10);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF10);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF10);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF10);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 1000') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T11

status12 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};
```

```
A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F11 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F11(2:45,3) = cell2mat(E);
F11(:,6)=a*((F11(:,2)./(F11(:,1))));

j = 11*i+i;
Time= F11(:,3)==j;
w = F11(Time, 6);
F11(:,4)=(F11(:,1)./(F11(:,2)-F11(1,1))).^(w);
CofF11 = F11(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF11);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF11);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF11);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF11);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF11);
fclose(fid);
```

```
fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF11);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF11);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 1200') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T12

status13 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));


ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);
```

```
F12 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F12(2:45,3) = cell2mat(E);
F12(:,6)=a*((F12(:,2)./(F12(:,1))));

j = 13*i+i;
Time= F12(:,3)==j;
w = F12(Time, 6);
F12(:,4)=(F12(:,1)./(F12(:,2)-F12(1,1))).^(w);
CofF12 = F12(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF12);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF12);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF12);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF12);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF12);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF12);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF12);
```

```
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 1400') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T13

status14 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F13 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);
```

```
F13(2:45,3) = cell2mat(E);
F13(:,6)=a*((F13(:,2)./(F13(:,1))));

j = 15*i+i;
Time= F13(:,3)==j;
w = F13(Time, 6);
F13(:,4)=(F13(:,1)./(F13(:,2)-F13(1,1))).^(w);
CofF13 = F13(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF13);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF13);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF13);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF13);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF13);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF13);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF13);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
```

```
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 1600') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T14

status15 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));

ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F14 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F14(2:45,3) = cell2mat(E);
F14(:,6)=a*((F14(:,2)./(F14(:,1))));

j = 17*i+i;
Time= F14(:,3)==j;
w = F14(Time, 6);
F14(:,4)=(F14(:,1)./(F14(:,2)-F14(1,1))).^(w);
CofF14 = F14(Time, 4);
```

```
fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF14);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF14);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF14);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF14);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF14);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF14);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF14);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 1800') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
```

```
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);


%T15

status16 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));

ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F15 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F15(2:45,3) = cell2mat(E);
F15(:,6)=a*((F15(:,2)./(F15(:,1))));


j = 19*i+i;
Time= F15(:,3)==j;
w = F15(Time, 6);
F15(:,4)=(F15(:,1)./(F15(:,2)-F15(1,1))).^(w);
CofF15 = F15(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF15);
fclose(fid);
```

```
fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF15);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF15);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF15);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF15);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF15);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF15);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 2000') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);
```

```
%T16

status17 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F16 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F16(2:45,3) = cell2mat(E);
F16(:,6)=a*((F16(:,2)./(F16(:,1))));

j = 21*i+i;
Time= F16(:,3)==j;
w = F16(Time, 6);
F16(:,4)=(F16(:,1)./(F16(:,2)-F16(1,1))).^(w);
CofF16 = F16(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF16);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF16);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF16);
fclose(fid);
```

```
fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF16);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF16);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF16);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF16);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 2200') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T17

status18 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
```

```
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F17 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F17(2:45,3) = cell2mat(E);
F17(:,6)=a*((F17(:,2)./(F17(:,1))));

j = 23*i+i;
Time= F17(:,3)==j;
w = F17(Time, 6);
F17(:,4)=(F17(:,1)./(F17(:,2)-F17(1,1))).^(w);
CofF17 = F17(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF17);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF17);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF17);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF17);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF17);
fclose(fid);
```

```
fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF17);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF17);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 2400') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T18

status19 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);
```

```
F18 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F18(2:45,3) = cell2mat(E);
F18(:,6)=a*((F18(:,2)./(F18(:,1))));

j = 25*i+i;
Time= F18(:,3)==j;
w = F18(Time, 6);
F18(:,4)=(F18(:,1)./(F18(:,2)-F18(1,1))).^(w);
CofF18 = F18(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF18);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF18);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF18);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF18);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF18);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF18);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF18);
fclose(fid);
```

```
%# read & write to other file

bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 2600') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T19

status20 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F19 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F19(2:45,3) = cell2mat(E);
```

```
F19(:,6)=a*((F19(:,2)./(F19(:,1))));

j = 27*i+i;
Time= F19(:,3)==j;
w = F19(Time, 6);
F19(:,4)=(F19(:,1)./(F19(:,2)-F19(1,1))).^(w);
CofF19 = F19(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF19);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF19);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF19);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF19);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF19);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF19);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF19);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
```

```
while ~strcmp(l, 'TIME 2800') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T20

status21 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F20 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F20(2:45,3) = cell2mat(E);
F20(:,6)=a*((F20(:,2)./(F20(:,1))));

j = 29*i+i;
Time= F20(:,3)==j;
w = F20(Time, 6);
F20(:,4)=(F20(:,1)./(F20(:,2)-F20(1,1))).^(w);
CofF20 = F20(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
```

```
fprintf(fid,'%f\n',CofF20);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF20);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF20);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF20);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF20);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF20);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF20);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 3000') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
```

```
fclose(infid1);

%T21

status22 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F21 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);
F21(2:45,3) = cell2mat(E);
F21(:,6)=a*((F21(:,2)./(F21(:,1))));

j = 38*i+i;
Time= F21(:,3)==j;
w = F21(Time, 6);
F21(:,4)=(F21(:,1)./(F21(:,2)-F21(1,1))).^(w);
CofF21 = F21(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF21);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF21);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF21);
fclose(fid);
```

```
fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF21);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF21);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF21);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF21);
fclose(fid);



%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 3900') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

%T22

status23 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);

fid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.out','r');

C = textscan(fid, '%s','Delimiter','');
```

```
fclose(fid);
C = C{:};

A = ~cellfun(@isempty, strfind(C,'Ave Pres POVO'));
ind = A;
B=C(ind);

D=cellfun(@(x) x(30:end), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F22 = cell2mat(E);

A = ~cellfun(@isempty, strfind(C,'The Time is'));
ind = find(A);
B=C(ind);

D=cellfun(@(x) x(14:25), B, 'UniformOutput', false);

E=cellfun(@str2num,D,'un',0);

F22(2:45,3) = cell2mat(E);
F22(:,6)=a*((F22(:,2)./(F22(:,1))));

j = 49*i+i;
Time= F22(:,3)==j;
w = F22(Time, 6);
F22(:,4)=(F22(:,1)./(F22(:,2)-F22(1,1))).^(w);
CofF22 = F22(Time, 4);

fid = fopen('Coff.txt','w', 'n');
fprintf(fid,'*TRANSMF CON  ');
fprintf(fid,'%f\n',CofF22);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSMULT *ACCUMULATE  ');
fprintf(fid,'%f\n','');
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF22);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSI *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF22);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF22);
fclose(fid);
```

```
fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSJ *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF22);
fclose(fid);


fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *MATRIX IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF22);
fclose(fid);

fid = fopen('Coff.txt','a+');
fprintf(fid,'*TRANSK *FRACTURE IJK 1:19 1:19 1:1  ');
fprintf(fid,'%f\n',CofF22);
fclose(fid);


%# read & write to other file


bigfid = fopen('TSF_Kazemi_w_Vari_Time_Step_Run.dat', 'rt+'); %open the
big file in read/write text mode
infid1 = fopen('Coff.txt', 'rt'); %open the other file in read text
mode
%read lines until you get to the insertion point:
l = fgetl(bigfid); %read first line
while ~strcmp(l, 'TIME 5000') %or other comparison functions
  l = fgetl(bigfid);
end
insertpos = ftell(bigfid); %memorise insertion point
remainder = fread(bigfid); %read the rest of the file to rewrite after
insertion
fseek(bigfid, insertpos, 'bof'); %rewind to insertion point
fwrite(bigfid, fread(infid1)); %copy content of other file at insertion
point
fwrite(bigfid, remainder); %and write back the rest of the big file
fclose(bigfid);
fclose(infid1);

status24 = dos(['"C:\Program Files
(x86)\CMG\STARS\2013.10\Win_x64\EXE\st201310.exe" -f "' pwd '\'
'TSF_Kazemi_w_Vari_Time_Step_Run.dat" -log -parasol 4']);
```