# A NEW APPROACH TO STREAMLINE SIMULATION ON THERMAL ENHANCED OIL RECOVERY BY HOT WATERFLOODING

A Thesis

by

ALFINO HARUN LUBIS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

| | |
|---|---|
| Chair of Committee, | John E. Killough |
| Committee Members, | Eduardo Gildin |
| | Maria A. Barrufet |
| Head of Department, | A. Daniel Hill |

August 2016

Major Subject: Petroleum Engineering

# ABSTRACT

Hot waterflooding is a well-known method of thermal enhanced oil recovery. Prediction of the reservoir behavior is generally obtained through numerical simulation. The finite volume simulator is the most common tool to simulate the thermal reservoir. However, the computational cost of modeling large-scale, fine-grid, heterogeneous reservoirs is high due to the nonlinearity and complexity of the non-isothermal problem. The streamline-based simulator complements the finite volume simulator in generating faster results by decoupling the 3D flow equation into several 1D problems along the streamlines. Implementation of the streamline simulator in the thermal model has been introduced by several researchers. As of now, the heat conduction part is mostly solved using the operator splitting technique at the end of a global time step. This technique can introduce the operator splitting error and may not be a unique solution.

This work proposes a new approach to handle heat conduction in a streamline simulation by solving it simultaneously with the pressure equation along the gridblocks. A 2D, quarter five-spot well pattern with one injector and one producer was tested on the new streamline simulator, and the results are in agreement with the commercial finite volume simulator. Compared to the operator splitting method, the new approach is slightly slower due to a larger set of linear equation systems. However, the better accuracy compensates for the longer computational time, and it is still much faster than the commercial thermal simulator.

# DEDICATION

This thesis is dedicated to my parents, sisters, and friends who have given their full support during my time as a graduate student.

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. John Killough for his continuous support throughout my research. Without his guidance, this thesis would not have been completed. One simply could not wish for a better or friendlier advisor. I am also grateful to Dr. Maria Barrufet and Dr. Eduardo Gildin for their willingness to serve as my committee members.

I would like to extend my gratitude to all of the members of Indonesian Student Association who have made my time in College Station more pleasant. To all students in Killough Research Group, thank you for listening, offering advice, and supporting me through the entire process. Furthermore, I am grateful to all of the staff and faculty members of Texas A&M University who have provided an excellent study environment to me for the last two years.

Last but not least, I am thankful for all of the support and motivation that my family has given me throughout my life.

# NOMENCLATURE

| | |
|---|---|
| 1D | One-Dimensional |
| 2D | Two-Dimensional |
| 3D | Three-Dimensional |
| API | American Petroleum Institute |
| CFL | Courant-Friedrichs-Lewy |
| CMG | Computer Modelling Group |
| CPU | Central Processing Unit |
| FIM | Fully Implicit Method |
| EOR | Enhanced Oil Recovery |
| GB | Gigabyte |
| IMPES | Implicit Pressure Explicit Saturation |
| NSPCG | Non-Symmetric Preconditioned Conjugate Gradient |
| PDE | Partial Differential Equation |
| RAM | Random Access Memory |
| SAGD | Steam Assisted Gravity Drainage |
| SPU | Single Point Upstream |
| STARS | Commercial Thermal Reservoir Simulator |

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

# CHAPTER I

# INTRODUCTION AND LITERATURE REVIEW

## 1.1 Introduction

A huge part of the world oil resources comes from the heavy oil, which can be described as oil with viscosity larger than 100 cp and API gravity less than 22 (Pasarai 2007). Current total worldwide heavy oil and bitumen resources are estimated around 9 trillion bbls with the largest heavy oil resources located at Orinoco Belt in Venezuela (Alboudwarej et al. 2006). The high viscosity of heavy oil leads to flow difficulties. For example, typical recovery factors under primary recovery range from 1 to 10% (Curtis et al. 2002). This is considered inefficient. Thermal recovery methods by hot fluid injection have enhanced the recovery of viscous oils by reducing the viscosity with increasing temperature, causing more oil to be recovered. According to the data from U.S. Department of Energy, thermal recovery methods take up to 50% of local Enhanced Oil Recovery (EOR) production (Zhu, Gerritsen, and Thiele 2010). Several thermal methods such as hot waterflooding, steamflooding, and cyclic steam stimulation are widely used, as well as in-situ combustion and the currently popular steam-assisted gravity drainage (SAGD) to produce heavy crude oil. During a heavy oil project, forecasts of reservoir behavior to thermal fluid injection are necessary. Numerical reservoir simulation is the most common way to yield predictions.

The standard industry practice of thermal reservoir simulation has been to use the traditional finite volume (FV) technique that applies a fully implicit (FIM) time stepping

method (Aziz and Settari 1979). This method has been found to be stable to handle rapid changes in properties and can utilize larger time steps compared to the explicit IMPES method. However, high computational cost is encountered during the simulation of real field-scale models with fine grid resolution due to the complexity of thermal problem and strong nonlinearity. Moreover, the method could predict erroneous results if the numerical scheme choice is not controlled. Since then, several techniques of improving the simulator efficiency have been proposed (Tamim, Abou-Kassem, and Farouq Ali 1999). A simple method with a balanced thermodynamic detail and computational efficiency was still not available until the mid-decade of the 2000s where thermal streamline-based simulators were growing.

Streamline-based simulator solves the pressure equation implicitly to calculate a set of streamlines that illustrate the flow in reservoir. The displacement process is captured in one-dimensional (1D) solution along each streamline. Solving transport phenomena such as fluid mass or heat along the streamline grid rather than Eulerian gridblocks provides several advantages. The solutions will be grid insensitive and less affected by the grid orientation effects. It also provides better indication of flow features which is very helpful in dynamic reservoir characterization. Also, we can use large global time step because it does not need to be under the grid-based stability limitations. In steady-state isothermal case, streamline method is an ideal way to quantify the relationship between injector and producer, determine flow pattern, and assess various field development scenarios.

Based on the success of streamline application in steady-state isothermal problems, we try to implement the method to unsteady-state non-isothermal case that typically occurs in thermal recovery process. The developed 2D thermal streamline simulator will solve the pressure, saturation, and temperature. A different approach will be made when taking account the heat conduction effect. Instead of using operator splitting, heat conduction will be solved simultaneously with pressure. It is believed that this approach will generate better results. The simulator is implemented in FORTRAN 90 codes with hot waterflooding as the base case in simulating the performance of thermal recovery method. Hot waterflooding is a great starting case for streamline simulation research before expanding to a more complicated process such as steamflooding or SAGD. Results from the developed simulator will be validated with the data from CMG STARS, a commercial thermal finite volume simulator.

## 1.2 Literature Review

### *1.2.1 Streamline Simulation*

Streamline itself is defined as the curve that is instantaneously tangent to the velocity vector of the flow (Bear 1972). Streamline-based simulation offers a very different technique of calculating variables over the ordinary grid-based finite volume simulation. In this method, we use streamlines instead of Eulerian grids to solve conservation equations. Two different time steps and grids are employed. Pressure is updated with larger time steps (or global time steps) on gridblocks, while saturation and temperature are updated with smaller time steps (or local time steps) on 1D grids along

each streamline using the time-of-flight variable instead of arc-length coordinate. Streamline method is faster than finite volume method because it can take large global time steps with fewer pressure solutions (Osako, Datta-Gupta, and King 2003). This method is also insensitive to grid orientation effects commonly associated with finite volume method. Many advantages of streamline simulation have led its application to various commercial software (King and Datta-Gupta 1998).

Contrast to the early streamtube simulation, streamline simulation does not require explicit computation of volume elements. Instead, it replaces them with calculation along lines. Each streamline can be thought as the center of streamtube where the velocity is obtained from a previous numerical finite volume calculation. In comparison, velocity in streamtube simulation must be determined from the volumetric flux per unit of explicit streamtube area. With streamlines, geometry is already implicit making it simpler to calculate in three-dimension (3D) (Pasarai 2007). Application of streamtube is also limited to 2D pattern flood whereas streamline application covers full range of compositional and multiphase displacements in 3D (Batycky 1997).

The majority of streamline simulators make use of Pollock's algorithm for tracing streamlines (1988). This is a semi-analytical method that points out the exit point of a streamline and its time-of-flight inside of a gridblock by assuming a linear interpolation of the velocity field from the fluxes of gridblock faces (Pasarai 2007). The tracing technique was further developed by introducing the variable time-of-flight to portray transport phenomena in a porous medium (Datta-Gupta and King 1995). Time-of-flight is the time required for a trace to reach a distance, s, along a streamline (Batycky 1997).

Current streamline simulators employ the concept of time-of-flight because of the simplicity and its ability to decouple a 3D problem into series of 1D problem. The discovery of time-of-flight variable is the most significant contribution in streamline simulation (Pasarai 2007).

In solving the conservation equations along streamlines, analytical or numerical techniques can be used. Analytical solution is restricted to fixed streamline paths and only applicable to uniform initial conditions. The numerical solution is more general and appropriate for changing well and mobility conditions (Pasarai 2007). The built simulator in this project will only utilize the numerical technique. Batcyky solved the 1D saturation equation numerically with a single point upstream (SPU) weighting finite difference explicit in time and spatially discretized the equation in time-of-flight ($\tau$) variable (1997). Based on his observation, there is a refinement of streamlines in the high-velocity area and reduced resolution in the low-velocity area.

Before solving the equation numerically, the irregular-spaced of $\tau$ grid must be converted to regular-spaced $\tau$ grid because the local time step is controlled by the local Courant-Friedrichs-Lewy (CFL) stability criterion in explicit time scheme. It was further found that implicit SPU is more efficient than explicit SPU, but produces larger numerical dispersion (Yan et al. 2004). Numerical stability can also be improved by choosing the right time step size for pressure updates (Osako, Datta-Gupta, and King 2003). The proposed guideline of time step size by Osako et al. pretty much eliminates the subjectivity of streamline simulation by introducing the automatic control of global time step.

Updated properties are mapped from streamlines back to gridblocks at the end of each global time step. The mapping procedure is based on the assumption that the flux is constant along each streamline (Batycky 1997). However, this assumption is only valid for incompressible flow because compressible flow creates flux divergence along the streamline. Therefore, this variation of flux has to be taken into account when averaging the properties from streamlines to gridblocks for compressible flow. For compressible flow case, we also need to add a source/sink term to the 1D transport equation along the streamline. By doing this, the fluid volume changes with pressure is captured, and compressibility effect is well represented (Cheng et al. 2005).

One of the most important advances in streamline simulation is the introduction of operator splitting method to decouple convective and diffusion parts in the transport equation. This method has been implemented in streamline simulators and shows reasonable accuracy and efficiency (Crane et al. 2000). Streamline works best for flow that is dominated by convection process and less suited for the diffusion process. Thus, operator splitting is needed to solve both processes separately. First, convective part is solved along 1D streamlines by finite difference method. The updated properties from streamlines are then used to compute new values of properties due to diffusion in the gridblocks. Various diffusion mechanisms such as capillary, gravity, and heat conduction effects can be modeled using the operator splitting technique (Pasarai 2007). A general workflow of the streamline-based simulator for a global time step can be seen in **Figure 1** (Zhu, Gerritsen, and Thiele 2010).

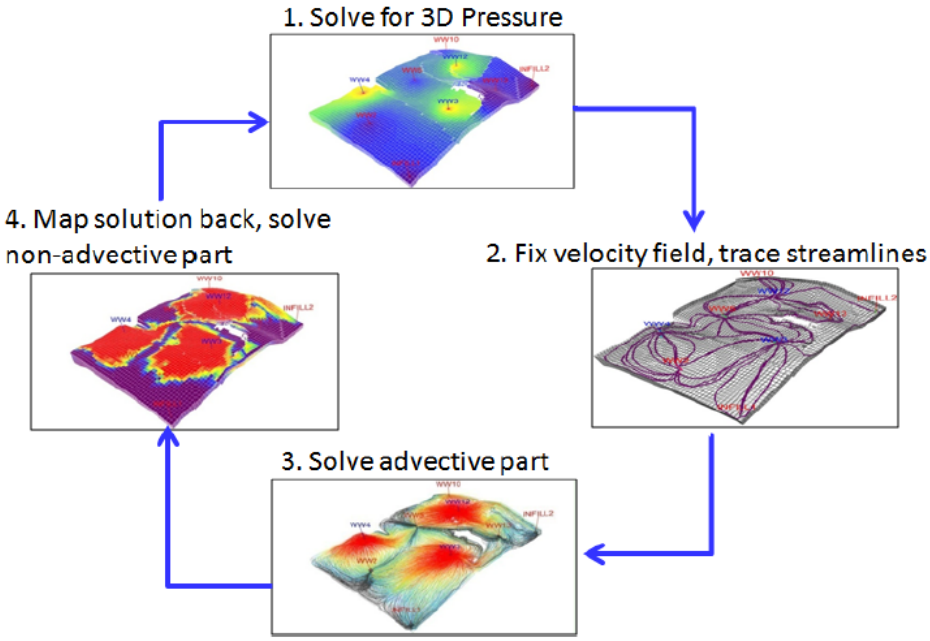



**Figure 1: General workflow of streamline-based simulation for a global time step (Zhu, Gerritsen, and Thiele 2010)**

Streamline simulation is well known to be efficient and complementary to the traditional finite volume simulation in large, geologically complex and heterogeneous systems. Most of the computational work in streamline simulation comes from solving the transport equations along the independent 1D streamlines. This creates a great opportunity for parallelization in streamline simulation framework. Recent research shows promising results in parallel speed-up (Lof, Gerritsen, and Thiele 2008). The high computational efficiency of streamline simulation makes it suitable for cases requiring many simulations such as history matching, ranking and optimization problems. One apparent disadvantage of this simulation technique is that it is not mass conservative due to the errors produced while mapping the properties from Eulerian grid to 1D streamline grid. Improved methods

of mapping have been proposed to minimize the mass balance errors (Mallison, Gerritsen, and Matringe 2004). Regardless, studies show that the mass balance errors are relatively small, and streamline method still predicts the global sweep in heterogeneous reservoirs effectively (Zhu, Gerritsen, and Thiele 2010).

### *1.2.2 Hot Waterflooding*

In a simple case, hot waterflooding involves the flow of water and oil phases only. Water that has been heated to a temperature higher than the reservoir condition but lower than the vaporization temperature is injected. The hot water will flow into cooler region, and some of the heat will be lost to surroundings until it is cooled back to initial temperature of the reservoir. This process will create oil bank that continues to grow ahead of the expanding heated zone. The primary role of heated water is to reduce oil viscosity so that the local displacement efficiency will be improved. Hot waterflooding is well suited for reservoir with low to moderate in-situ oil viscosity and various oil density. This type of reservoir is usually shallow where permeability of the order of 10's of Darcies is common (Pasarai 2007).

Field applications proved that hot waterflooding is able to recover additional volumes of oil and extend the economic life of a heavy oil reservoir (Kasrale, Sammon, and Jespersen 1993). Hot water injection is also the least expensive thermal enhanced oil recovery method (Farouq Ali 1974). Although steamflooding has three times more heat capacity than hot waterflooding method, it is costly to generate, and the field implementation is intensive. Another advantage of hot waterflooding is its capability of

providing higher injection pressure than steamflooding which is very efficient in maintaining reservoir pressure.

Streamline simulation application to hot waterflooding was first proposed by Pasarai and Arihara. Their simulator solved both oil and water mass transport equations on the gridblocks and the energy equation along the 1D streamlines to take account of unsteady-state flow (Pasarai 2007). Later, Zhu et al. improved the simulation by fully solving both mass and energy transport along streamlines while rigorously taking account for compressibility and non-advective effects (2010). The developed simulator was tested in 2D case of a simple quarter five-spot with one injector and one producer. Based on their observation, Zhu et al. found that operator splitting may not be the unique solution to efficiently incorporate heat conduction effect into thermal streamline simulation. Most recently, the simulation was performed on a 3D real, field-scale reservoir with complex heterogeneity distribution and multiple wells (Siavashi et al. 2014). Their simulator produced results that were in agreement with commercial simulator. All of the previous streamline-based simulators for hot waterflooding utilize the operator splitting to model the heat conduction effect.

# CHAPTER II

# RESEARCH STATEMENT

Forecast of reservoir performance in a thermal enhanced oil recovery process by hot waterflooding is essential. The typical finite volume simulator requires high computational cost due to the complexity and nonlinearity of the thermal problem. Streamline-based simulator provides an alternative fast approximate solution and has been tested in the hot waterflooding case by several researchers. As of now, heat conduction effect is always incorporated in streamline-based simulator by using the operator splitting method which may not be the unique solution.

This study evaluates a new approach to handle the heat conduction effect in a 2D streamline-based simulator. Instead of solving the heat conduction with operator splitting at the end of global time step, it is solved simultaneously with pressure equation on Eulerian grid at the beginning of global time step. Comparison between the two methods is assessed. Furthermore, the speed and accuracy of the developed simulator are also compared with a commercial finite volume thermal simulator.

# CHAPTER III

# GOVERNING EQUATIONS

This chapter presents a mathematical model for thermal simulation of hot waterflooding using the streamline method. Starting with the general mass and energy balance equations, we further reduce the equations to a system of PDE by making key assumptions related to the hot waterflooding case. The mathematical derivation of streamline equations is also discussed, ranging from the coordinate transformation, streamline tracing, to the transport equation along the streamlines.

## 3.1 General Mass Conservation

Transport mechanism of a particular fluid component can be carried out by bulk flow and molecular diffusion. Darcy's Law governs the bulk volumetric flux which is the volume flow rate per unit cross-sectional area. The generalized form of Darcy's Law can be written as,

$$\vec{u}_j = -\frac{\bar{\bar{k}}k_{rj}}{\mu_j}\left(\nabla P_j - \rho_j g \nabla D\right) \quad (3.1)$$

where $\bar{\bar{k}}$ is the permeability tensor, $k_{rj}$ is the phase relative permeability, $P_j$ is the phase pressure, $\rho_j$ is the phase density, $\mu_j$ is the phase viscosity, $g$ is gravitational constant, and $D$ is the reservoir depth. Here, the phase $j$ can be water, oil, or gas phase. On the other hand, diffusive flux per unit cross-sectional area is given by Fick's Law,

$$\vec{u}_{D_{ij}} = -\bar{\bar{D}}_{ij} \cdot \nabla \omega_{ij} \quad (3.2)$$

where $\overline{\overline{D}}_{ij}$ is dispersivity tensor and $\omega_{ij}$ is the mass fraction of component $i$ in phase $j$.

In an arbitrary control volume V within a permeable porous medium, a number of components are flowing through. The law of mass conservation declares that for a certain component $i$, the summation of net mass flux rate and sink/source rate must be equal to the mass accumulation rate. We can express this in the form of PDE,

$$\frac{\partial}{\partial t}\left(\emptyset \sum_{j=1}^{n_p} \rho_j S_j \omega_{ij}\right) + \nabla \cdot \left(\sum_{j=1}^{n_p}\left(\vec{u}_J \rho_j \omega_{ij} - S_j \emptyset \rho_j \overline{\overline{D}}_{ij} \cdot \nabla \omega_{ij}\right)\right) = \sum_{j=1}^{n_p} q_j \omega_{ij} \quad \text{.....(3.3)}$$

where $\emptyset$ is the porosity, $S_j$ is the phase saturation, and $q_j$ is the mass flow rate of a source or sink. The general form of Eq. (3.3) is applied to each component of interest. Later, we reduce the equation to a simpler form of continuity equation by making several assumptions related to the hot waterflooding model.

### 3.2 General Energy Conservation

Heat transportation mechanism can be due to conduction, convection, and radiation. Radiation seldom occurs in a reservoir, so it is safe to ignore the term in reservoir modeling. Conduction and convection process are analogous to the mass transport by diffusion and bulk flow, respectively. Heat conduction is caused by microscopic collisions of particles and electrons movement within a body. Fourier's Law describes the conduction heat flux as,

$$\vec{u}_K = -K_T \nabla T \quad \text{.....(3.4)}$$

$$K_T = (1 - \emptyset) K_r + \emptyset \left(\sum_{j=1}^{n_p} S_j K_j\right) \quad \text{.....(3.5)}$$

where $K_T$ is the total thermal conductivity, $K_r$ is the rock thermal conductivity, $K_j$ is the phase thermal conductivity, and $T$ is the temperature. In contrast to conduction, heat convection relates the process of heat transfer caused by fluid movement. The associated convective heat flux with negligible kinetic and potential energy is,

$$\vec{u}_T = \vec{u}_j \rho_j H_j \quad (3.6)$$

where $H_j$ is the phase enthalpy.

The main difference between non-isothermal and isothermal simulation is the addition of energy balance equation. Similar to mass balance equation, the summation of net heat flux rate and heat sink/source rate must be equal to the heat accumulation rate. For practical purposes, kinetic and potential energies will be negligible because of their small contribution in a typical thermal recovery process. Therefore, we can write the PDE of energy conservation as,

$$\frac{\partial}{\partial t}\left(\emptyset \sum_{j=1}^{n_p} \rho_j S_j U_j + (1-\emptyset)\rho_r C_r T\right) + \nabla \cdot \left(\sum_{j=1}^{n_p} \vec{u}_J \rho_j H_j\right) - \nabla \cdot (K_T \nabla T) = \sum_{j=1}^{n_p} q_j H_j \quad (3.7)$$

where $U_j$ is the internal energy of the phase and $C_r$ is the rock heat capacity. Unlike the mass conservation equation that requires one equation for each component, energy conservation only implements Eq. (3.7) for the whole system.

### 3.3 Pressure Equation

Streamline method employs a sequential procedure to model the reservoir, by separating the flow or pressure equation from the conservation equations. Several key assumptions are made to derive PDEs for our 2D hot waterflooding model such as: (1) it

only consists of two components: water and a pseudocomponent 'oil', (2) water and 'oil' components are immiscible which means that the 'oil' component is insoluble in water phase, and water component is insoluble in oil phase, (3) phase pressure of water and oil are always higher than bubble point pressure which indicates no formation of gas phase, (4) relative permeability is not a function of temperature, (5) gravity and capillary pressure are negligible, (6) heat loss to overburden and underburden does not occur, (7) contribution of diffusive flow is very small, (8) the rock is incompressible. Consequently, the mass conservation equations for water and oil components are reduced to,

$$\frac{\partial}{\partial t}(\emptyset \rho_w S_w) + \nabla \cdot (\rho_w \vec{u}_w) = q_w \quad (3.8)$$

$$\frac{\partial}{\partial t}(\emptyset \rho_o S_o) + \nabla \cdot (\rho_o \vec{u}_o) = q_o \quad (3.9)$$

Substitute Darcy's Law in Eq. (3.1) into above equations and eliminate the gravity and capillary term, resulting in,

$$\frac{\partial}{\partial t}(\emptyset \rho_w S_w) - \nabla \cdot \left(\rho_w \frac{\bar{\bar{k}} k_{rw}}{\mu_w} \nabla P\right) = q_w \quad (3.10)$$

$$\frac{\partial}{\partial t}(\emptyset \rho_o S_o) - \nabla \cdot \left(\rho_o \frac{\bar{\bar{k}} k_{ro}}{\mu_o} \nabla P\right) = q_o \quad (3.11)$$

Pressure equation is obtained by summing up Eq. (3.10) and (3.11),

$$\emptyset c_t \frac{\partial P}{\partial t} - \nabla \cdot \left(\frac{\bar{\bar{k}} k_{rw} \rho_w}{\mu_w} \nabla P\right) - \nabla \cdot \left(\frac{\bar{\bar{k}} k_{ro} \rho_o}{\mu_o} \nabla P\right) = q_w + q_o \quad (3.12)$$

with

$$c_t = S_w \rho_w c_w + S_o \rho_o c_o \quad (3.13)$$

where $c_t$ is the total compressibility, $c_w$ is the water compressibility, and $c_o$ is the oil compressibility. Note that the rock compressibility term is not involved because we assume that the rock is incompressible. Water and oil compressibility are defined as,

$$c_w = \frac{1}{\rho_w}\frac{\partial \rho_w}{\partial P} \quad (3.14)$$

$$c_o = \frac{1}{\rho_o}\frac{\partial \rho_o}{\partial P} \quad (3.15)$$

To solve Eq. (3.12), boundary conditions are defined at wells and no-flow boundaries. The mass phase flow rate per unit volume, $q_j$, is determined from Peaceman equation as follows,

$$q_j = WI\frac{k_{rj}\rho_j}{\mu_j}(P_{well} - P) \quad (3.16)$$

where $P_{well}$ is the wellbore pressure, $P$ is the gridblock pressure, and $WI$ is the well index which is defined as,

$$WI = \frac{2\pi k_e \Delta z}{\ln\frac{r_e}{r_w} + s} \quad (3.17)$$

where $s$ is the skin factor, $r_w$ is the wellbore radius, $\Delta z$ is the height of gridblock, $k_e$ is the effective absolute permeability, and $r_e$ is the effective wellbore radius. Both $k_e$ and $r_e$ can be calculated by,

$$k_e = \sqrt{k_x/k_y} \quad (3.18)$$

$$r_e = 0.28\left[\frac{\left[(k_y/k_x)^{1/2}\Delta x^2 + (k_x/k_y)^{1/2}\Delta y^2\right]^{1/2}}{(k_y/k_x)^{1/4}(k_x/k_y)^{1/4}}\right] \quad (3.19)$$

The unknown in Peaceman equation is either the mass phase flow rate for Dirichlet pressure-constrained well or the wellbore pressure for Neumann rate-constrained well

boundary condition. To fully solve the pressure equation, several auxiliary equations are needed as follows.

- Phase saturation equation, $S_w + S_o = 1$
- Relative permeability is only a function of phase saturation, $k_{rw} = {S_w}^2$ and $k_{ro} = {S_o}^2$
- Phase density is a function of both pressure and temperature,

$$\rho_j = \rho_j^{ref} \exp\left[c_j\left(P - P_{ref}\right) - \alpha_j\left(T - T_{ref}\right)\right] \quad (3.20)$$

- Phase viscosity is only a function of temperature,

$$\mu_j = A_j \exp\left(B_j/T\right) \quad (3.21)$$

- Liquid enthalpy is calculated using the average heat capacity of the phase,

$$H_j = \int_{T_{ref}}^{T} \left(C_p(T)\right)_j dT = \left(\bar{C}_p\right)_j \left(T - T_{ref}\right) \quad (3.22)$$

- Internal energy can be calculated as,

$$U_j = H_j - P/\rho_j \quad (3.23)$$

## 3.4 Streamline Equations

### *3.4.1 Incompressible Flow*

The concept of time-of-flight ($\tau$) is crucial in streamline simulation (Datta-Gupta and King 1995). The general 3D flow equation can be transformed by using $\tau$ as spatial coordinate into multiple 1D equations. By solving the 1D problems, the computational time and numerical dispersion are reduced (Batycky 1997). From the definition, time-of-flight is the time required for a certain tracer to reach a distance ($\zeta$) along the streamline. Although the unit of this variable is in time, time-of-flight acts as the spatial coordinate in

streamline simulation. Properties in a 3D domain such as porosity and permeability are honored along streamlines by the use of $\tau$ coordinate (Pasarai 2007). In mathematical form, time-of-flight can be written as,

$$\tau(s) = \int_0^s \frac{\emptyset(\zeta)}{|\vec{u}_t(\zeta)|} d\zeta \quad (3.24)$$

Eq. (3.24) can be stated in differential form as,

$$\vec{u}_t \cdot \nabla\tau = \emptyset \quad (3.25)$$

Coordinate transformation from the physical space in streamline simulation requires three new spatial coordinates. These are $\tau$ and two other bi-streamfunctions $\psi$ and $\chi$ (Datta-Gupta and King 1995). The following is the relationship,

$$\vec{u}_t = \nabla\psi \times \nabla\chi \quad (3.26)$$

Assumption of incompressible flow is implicit in this representation because of the vector identity,

$$\nabla \cdot (\nabla\psi \times \nabla\chi) = 0 \quad (3.27)$$

Therefore, $\nabla \cdot \vec{u}_t = 0$ is valid for incompressible flow. The coordinate transformation is facilitated by the simple Jacobian of coordinate transformation using Eq. (3.25) and (3.26),

$$\left|\frac{\partial(\tau,\psi,\chi)}{\partial(x,y,z)}\right| = (\nabla\psi \times \nabla\chi) \cdot \nabla\tau = \vec{u}_t \cdot \nabla\tau = \emptyset \quad (3.28)$$

The direct relationship between physical space and $\tau$ coordinate can be seen by rearranging Eq. (3.28),

$$\emptyset dxdydz = d\tau d\psi d\chi \quad (3.29)$$

From the above equation, the transformation preserves pore volume along the streamlines. This is essential to maintain material balance. Gradient operator is now expressed in the $(\tau, \psi, \chi)$ coordinates as,

$$\nabla = (\nabla\tau)\frac{\partial}{\partial\tau} + (\nabla\psi)\frac{\partial}{\partial\psi} + (\nabla\chi)\frac{\partial}{\partial\chi} \quad (3.30)$$

$$\vec{u}_t \cdot \nabla = \vec{u}_t \cdot (\nabla\tau)\frac{\partial}{\partial\tau} + \vec{u}_t \cdot (\nabla\psi)\frac{\partial}{\partial\psi} + \vec{u}_t \cdot (\nabla\chi)\frac{\partial}{\partial\chi} \quad (3.31)$$

The second and third terms on the right-hand side of Eq. (3.31) disappear because of the orthogonality between $\vec{u}_t$ and the bi-streamfunctions ($\psi$ and $\chi$). At last, the final form of coordinate transformation from physical space (x,y,z) to $\tau$ coordinate in incompressible flow is,

$$\vec{u}_t \cdot \nabla = \emptyset\frac{\partial}{\partial\tau} \quad (3.32)$$

### *3.4.2 Compressible Flow*

The flow along a streamline that is always tangent to the velocity vector is called longitudinal flux. In steady state condition, the fluid is only considered longitudinal, and the streamline does not move for a global time step. However, this is not valid for unsteady state or compressible flow because the fluid movement is not only tangent to the velocity vector but there is additional flow called transverse flux that moves across the streamline (Pasarai 2007). A clear illustration of these two fluxes is described in **Figure 2**.

Transverse flux triggers the volumetric flux variation along the streamlines. As a result, the term $\nabla \cdot \vec{u}_t$ is not zero anymore. The new mathematical form of $\nabla \cdot \vec{u}_t$ is derived by Cheng et al. using Pollock's algorithm for the case of compressible flow (2005). By

taking into account volumetric flux variation along streamlines, significant improvement of performance prediction can be achieved. To start off, Eq. (3.26) must be modified to include a scale factor,

$$\rho\vec{u}_t = \nabla\psi \times \nabla\chi \quad (3.33)$$

where $\rho$ is the effective density (equal to 1 for incompressible flow). The left-hand term of Eq, (3.3) can further be represented as a conserved flux,

$$\nabla \cdot (\rho\vec{u}_t) = \nabla \cdot (\nabla\psi \times \nabla\chi) = 0$$

$$\vec{u}_t \cdot \nabla\rho + \rho\nabla \cdot \vec{u}_t = 0$$

$$\emptyset\frac{\partial\rho}{\partial\tau} + \rho\nabla \cdot \vec{u}_t = 0 \quad (3.34)$$

The above equation is equivalent to Eq. (3.27).

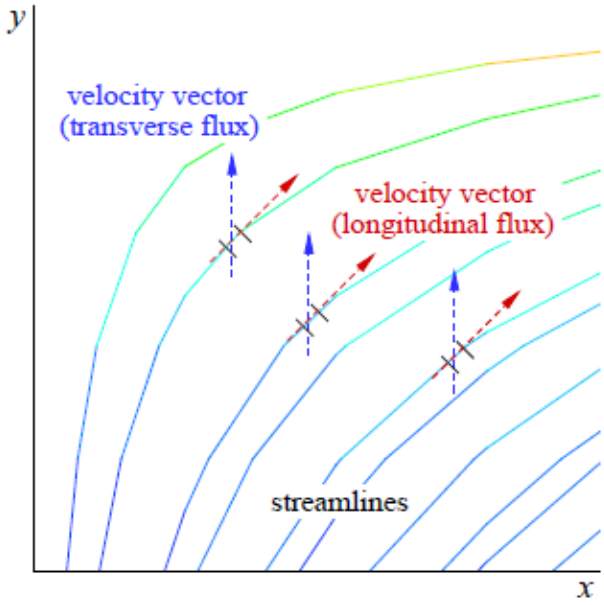



**Figure 2: Illustration of longitudinal and transverse fluxes along the streamlines (Pasarai 2007)**

As stated earlier, Pollock's algorithm is utilized to derive $\nabla \cdot \vec{u}_t$ for compressible flow. Pollock assumes a linear relationship between the values of gridblock interstitial velocity on cell faces. Interstitial velocity is obtained by dividing the total Darcy velocity by the porosity. A schematic of Pollock's application in 2D is represented in **Figure 3**. In mathematical form,

$$v_x = v_{x1} + a_x(x - x_1)$$

$$v_y = v_{y1} + a_y(y - y_1) \quad \text{.....(3.35)}$$

whereas the gradients are defined as,

$$a_x = (v_{x2} - v_{x1})/\Delta x$$

$$a_y = \left(v_{y2} - v_{y1}\right)/\Delta y \quad \text{.....(3.36)}$$

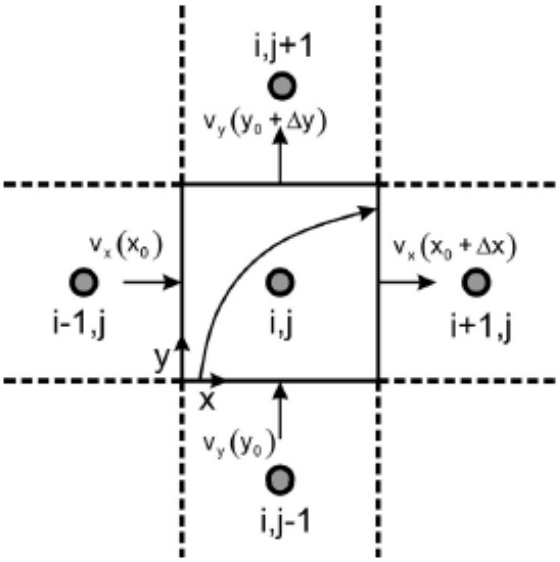



**Figure 3: Pollock's algorithm to trace streamline in 2D (Siavashi et al. 2014)**

According to Cheng et al., $\nabla \cdot \vec{u}_t$ is now defined in 2D for compressible flow as,

$$\nabla \cdot \vec{u}_t = a_x + a_y = a \quad (3.37)$$

If we substitute Eq. (3.37) into Eq. (3.34), and integrating within a cell along a streamline, we can obtain the formulation for effective density,

$$\rho = \rho_o e^{-(a\tau/\emptyset)} \quad (3.38)$$

At injector, the effective density $\rho_o$ is equal to 1. If we start tracing the streamline from injector, then the variation of volumetric flux along the streamline is directly related to the assigned flux at injector $Q_o$ in a way that,

$$Q_i = Q_o \frac{1}{\rho_i} = Q_o e^{(a\tau/\emptyset)} \quad (3.39)$$

It is easier to work with volumetric flux in Eq. (3.39) rather than the effective density in Eq. (3.38). In addition, Eq. (3.32) applies to both incompressible and compressible flow.

The Jacobian of coordinate transformation from physical space to $\tau$ coordinate is different in compressible flow. Correct procedure of this term is crucial to minimize mass balance errors during mapping from streamlines to gridblocks. Thus, Cheng et al. modified Eq. (3.28) and (3.29) into,

$$\left|\frac{\partial(\tau,\psi,\chi)}{\partial(x,y,z)}\right| = (\nabla\psi \times \nabla\chi) \cdot \nabla\tau = \rho\vec{u}_t \cdot \nabla\tau = \rho\emptyset \quad (3.40)$$

$$\emptyset dxdydz = \frac{1}{\rho} d\tau d\psi d\chi \quad (3.41)$$

### *3.4.3 Streamline Tracing*

Several works of literature have explained the procedure of streamline tracing and most of them use Pollock's semi-analytical tracing algorithm as the reference (1988). This

method is consistent with the material balance equation. The time it takes for a particle in a streamline to flow across a gridblock face can be determined by integrating Eq. (3.35) into,

$$\Delta t_{e,x} = \frac{1}{a_x} \ln \left[ \frac{v_{x,o} + a_x (x_e - x_o)}{v_{x,o} + a_x (x_i - x_o)} \right]$$

$$\Delta t_{e,y} = \frac{1}{a_y} \ln \left[ \frac{v_{y,o} + a_y (y_e - y_o)}{v_{y,o} + a_y (y_i - y_o)} \right] \quad \text{.....(3.43)}$$

where subscript i means inlet, e means exit, and o means origin. The true exit face is the one that generates minimum time in Eq. (3.43). Then, the exit location of the streamline in a gridblock is obtained by using the minimum time ($\Delta t_e$) and rearranging Eq. (3.43) as,

$$x_e = \frac{1}{a_x} \left( v_{x,i} \exp[a_x \Delta t_e] - v_{x,o} \right) + x_o$$

$$y_e = \frac{1}{a_y} \left( v_{y,i} \exp[a_y \Delta t_e] - v_{y,o} \right) + y_o \quad \text{.....(3.44)}$$

Calculations of the time and exit location are repeated to the next block until a sink is encountered. The time-of-flight is computed stepwise using the minimum time for each gridblock such that,

$$\tau = \sum_{i=1}^{n_{gb}} \Delta t_{e,i} \quad \text{.....(3.45)}$$

where $\Delta t_{e,i}$ is the current exit time of gridblock and $n_{gb}$ is the number of cells that has been passed by the streamline.

Most of the time, streamlines are launched from the gridblock faces of an injection well. Volumetric flux across each injection block face is uniform, and the streamlines are also uniformly distributed on the faces of injection block proportional to the assigned flux

across the block (Pasarai 2007). Therefore, the initial volumetric flux for each streamline is given by,

$$Q_{sl} = \frac{Q_f}{n_{sf}} \quad (3.46)$$

where $Q_f$ is the assigned flux of the block face and $n_{sf}$ is the number of streamlines launched from the block face. Careful consideration of the number of streamlines as a simulation input should be done so that each gridblock has at least one streamline passing through it. In the case where a certain active gridblock is not passed by a streamline, a new streamline is launched from the center of that block and traced backward to an injector. The process is continued until no missed cells are detected.

### *3.4.4 Transport Equations on Streamline*

After a thorough explanation of the basis of streamline, it is time to implement the mass and heat transport equations into $\tau$ coordinate along the streamline. The term fractional flow is needed for coordinate transformation, and it is defined as,

$$f_j = \frac{\vec{u}_j}{\vec{u}_t} = \frac{k_{rj}/\mu_j}{\sum k_{rj}/\mu_j} \quad (3.47)$$

Substitute Eq. (3.47) into Eq. (3.8) to obtain,

$$\emptyset \frac{\partial}{\partial t}(\rho_w S_w) + \nabla \cdot (\rho_w f_w \vec{u}_t) = q_w \quad (3.48)$$

Streamline works better for a convective-dominated problem. As a result, we will not take account heat conduction in the heat transport equation along the streamline. Eq. (3.7) reduces to,

$$\frac{\partial}{\partial t}(\emptyset[\rho_w S_w U_w + \rho_o S_o U_o] + (1-\emptyset)\rho_r C_r T) + \nabla \cdot (\vec{u}_w \rho_w H_w + \vec{u}_o \rho_o H_o) = q_w H_w^w + q_o H_o^w \quad (3.49)$$

where superscript $w$ indicates well condition.

Substitute fraction flow term from Eq. (3.47) to Eq. (3.49) and rearrange the equation to get the following,

$$\emptyset \frac{\partial E}{\partial t} + \nabla \cdot (F\vec{u}_t) = q_w H_w^w + q_o H_o^w \quad (3.50)$$

where $E$ is the total energy per unit volume defined as,

$$E = \rho_w S_w U_w + \rho_o S_o U_o + \frac{(1-\emptyset)}{\emptyset}\rho_r C_r T \quad (3.51)$$

and $F$ is convective energy written as,

$$F = \rho_w H_w f_w + \rho_o H_o f_o \quad (3.52)$$

In general, streamlines calculations are only performed outside the well block because all streamlines are launched from the face of the well block (Batycky 1997). If we expand the divergence operator, Eq. (3.48) and (3.50) will take form outside the well block as,

$$\frac{\partial}{\partial t}(\rho_w S_w) + \frac{\rho_w f_w}{\emptyset}\nabla \cdot \vec{u}_t + \frac{\vec{u}_t}{\emptyset} \cdot \nabla(\rho_w f_w) = 0 \quad (3.53)$$

$$\frac{\partial E}{\partial t} + \frac{F}{\emptyset}\nabla \cdot \vec{u}_t + \frac{\vec{u}_t}{\emptyset} \cdot \nabla F = 0 \quad (3.54)$$

To transform the coordinate, substitute Eq. (3.32) into the above equations and use $\nabla \cdot \vec{u}_t$ term from Eq. (3.37) to take account for fluid compressibility,

$$\frac{\partial}{\partial t}(\rho_w S_w) + \frac{\partial}{\partial \tau}(\rho_w f_w) = -\frac{a}{\emptyset}\rho_w f_w \quad (3.55)$$

$$\frac{\partial E}{\partial t} + \frac{\partial F}{\partial \tau} = -\frac{a}{\emptyset}F \quad (3.56)$$

Eq. (3.55) and (3.56) are mass and heat transport equation in $\tau$ coordinate, respectively.

### *3.4.5 Mapping Techniques*

The time-of-flight can vary between gridblocks and cause the fluid properties to be defined on irregular $\tau$ grid as shown in **Figure 4**. Before solving the transport equations along the streamlines, properties should be modified onto a regularly-spaced $\tau$ grid first. This will simplify the calculation of internode fluxes and provide stability in local time step. Property values can be assigned to regular $\tau$ grid using the following basis,

$$\left(\int_0^s X_j d\tau\right)_{regular} = \left(\int_0^s X_j d\tau\right)_{irregular} \quad (3.57)$$

where $X_j$ represents any property such as water saturation, pressure, or temperature.

For example, the value of water saturation in the third cell of regular grid that is shown in **Figure 5** can be computed by,

$$(S_{w3})_{regular} = \frac{(S_{w3})_{irregular}(\tau_3 - \tau_2') + (S_{w4})_{irregular}(\tau_3' - \tau_3)}{\tau_3' - \tau_2'} \quad (3.58)$$

By applying Eq. (3.57), mass and heat are conserved in the transformation process. However, the transformation from irregular to regular $\tau$ grid creates one source of numerical diffusion during the streamline simulation (Pasarai 2007).

After solving the transport equations along the streamline at the end of a global time step, water saturation and temperature are averaged back onto the Eulerian gridblocks. This action is highly dependent on the number of streamlines that pass through the cell. Because of the various volumetric flux along the streamline due to

compressibility, the averaging procedure must include the local volumetric flux and time-of-flight variable in such that,

$$\bar{S}_w = \frac{\sum_{i=1}^{n_s} S_{wi} q_i \Delta\tau_i}{\sum_{i=1}^{n_s} q_i \Delta\tau_i} \quad (3.59)$$

$$\bar{T} = \frac{\sum_{i=1}^{n_s} T_i q_i \Delta\tau_i}{\sum_{i=1}^{n_s} q_i \Delta\tau_i} \quad (3.60)$$

where $n_s$ is the number of streamlines in a gridblock and $q_i$ is the local volumetric flux computed by Eq. (3.39). Mapping is one of the main sources of mass balance error in streamline simulation, so it must be taken with care. Mapping error affects saturation more than the temperature because it may contain sharp gradients.

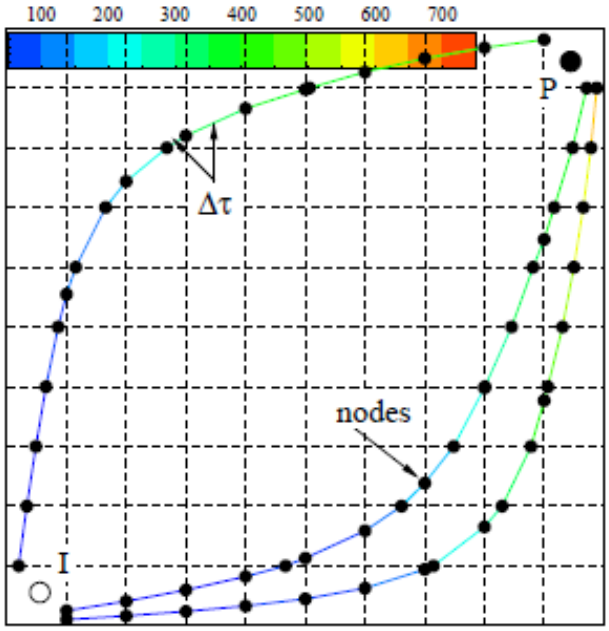



**Figure 4: Discretization in irregular $\tau$ coordinate of several streamlines with time-of-flight information in color (Pasarai 2007)**

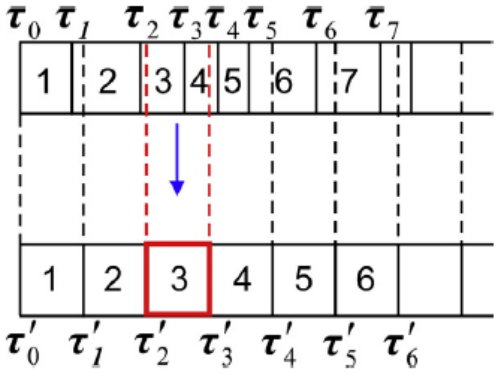



**Figure 5: Transformation of irregular to regular $\tau$ grid (Siavashi et al. 2014)**

### *3.4.6 Operator Splitting*

Operator splitting technique is commonly used to decouple convective and diffusive terms in streamline-based simulation. Convective part is solved along the streamlines in $\tau$ coordinate while diffusive part is solved using finite difference method on Eulerian grid. By implementing operator splitting, diffusion term can be solved independently and in regions where they are important (Pasarai 2007). For mass transport equation, diffusive term refers to gravity and capillary effect. Meanwhile for heat transport equation, diffusive term refers to heat conduction. This study will focus only on the heat conduction effect.

After the energy transport equation in Eq. (3.56) is solved, we get the new temperature distribution $T_c$ along the streamlines. These $T_c$ values are mapped back to gridblock using Eq. (3.60) and operator splitting of heat conduction part,

$$\frac{\partial E}{\partial t} - \frac{1}{\emptyset}\nabla \cdot (K_T \nabla T) = 0 \quad (3.61)$$

is solved on Eulerian grid while holding other variables (pressure and saturation) constant and using $T_c$ as the initial condition. The utilization of operator splitting to the energy balance equation most likely introduces a splitting error and may not be the unique solution (Zhu, Gerritsen, and Thiele 2010). Thus, this study aims to provide a better alternative technique to capture the heat conduction effect in streamline simulation without operator splitting.

CHAPTER IV

SIMULATION PROCEDURE

## 4.1 General Workflow

A schematic workflow of the new approach to streamline simulation taken in this study is illustrated in **Figure 6**. The main difference between general streamline simulation and this work is located in the treatment of heat conduction effect. Instead of

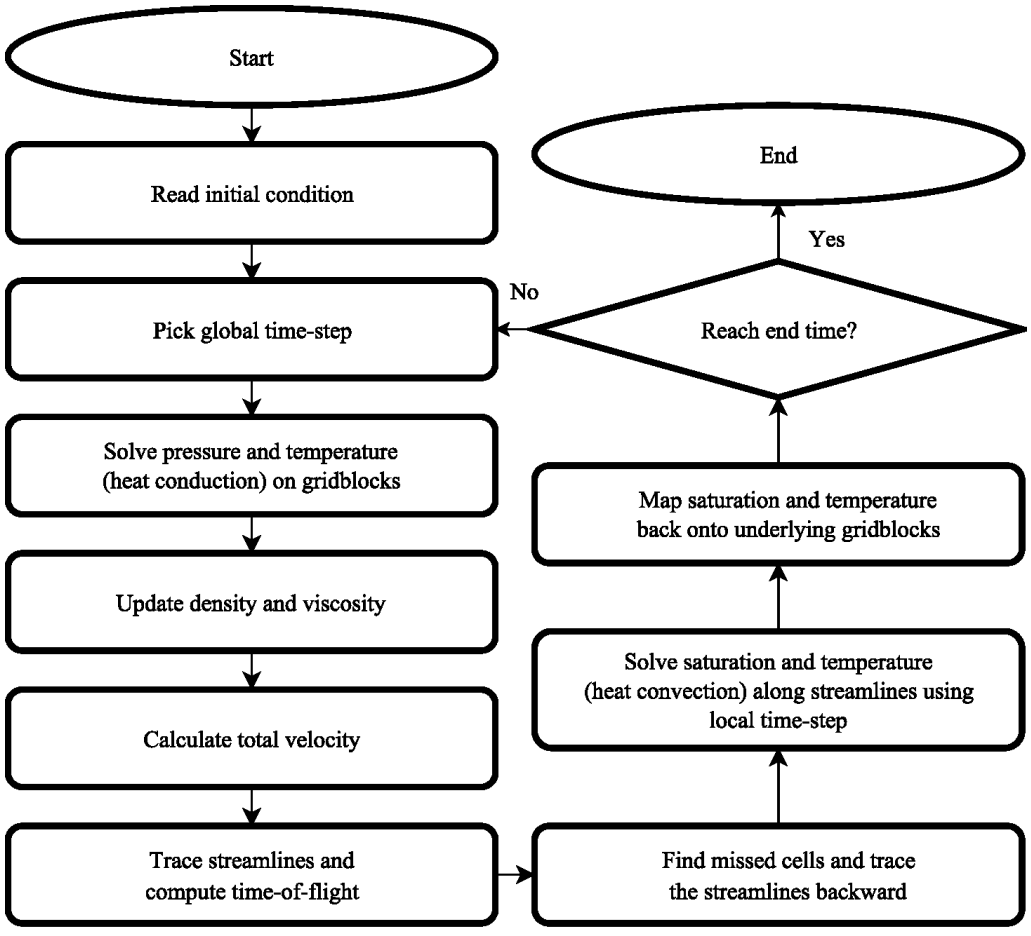



**Figure 6: General workflow of the streamline simulator using the new approach**

using operator splitting, the heat conduction equation is decoupled from the general energy balance equation in Eq. (3.7) and solved simultaneously with pressure equation on the 2D Eulerian grid,

$$-\nabla \cdot (K_T \nabla T) = q_w H_w^w + q_o H_o^w \quad \text{.....(4.1)}$$

The following is the general procedure to move the solution forward for a global time step:

1. Make sure that the initial condition for each variable is available on all active gridblocks.
2. Solve the pressure equation in Eq. (3.12) and heat conduction equation in Eq. (4.1) simultaneously for pressure and temperature. Use Newton-Raphson method to linearize the equations.
3. Use Darcy's Law in Eq. (3.31) to obtain the total velocity at gridblock faces. Interstitial velocity is determined by dividing Darcy's velocity with porosity.
4. Trace streamlines from injection block to producer as described in Chapter 3.4.3.
5. Store the current values of pressure, temperature, and water saturation from the gridblock to every streamline passing through. Compute variables $\tau$, $a$, and $Q_i$ for each gridblock. Transform all variables ($p$, $S_w$, $T$, $\tau$, $a$, $Q_i$) from irregular $\tau$ grid to regular $\tau$ grid.
6. Solve the mass transport equation in Eq. (3.55) and heat transport equation in Eq. (3.56) simultaneously along the streamlines for the updated values of saturation and temperature. Solution moves forward explicitly in time with small local time step until

the global time step is achieved. Transform back the updated variable values from regular $\tau$ grid to irregular $\tau$ grid.

7. Map the updated saturation and temperature from streamlines to gridblocks using Eq. (3.59) and (3.60). All properties and well data at the end of global time step can be printed.
8. Return to step 1 with the next global time step until simulation end time is reached.

### 4.2 Solution of the Simulation

The pressure equation in Eq. (3.12) and heat conduction equation in Eq. (4.1) are both solved simultaneously by the standard finite difference scheme on the Eulerian grid with a specified rate or pressure at the well and no-flow boundary conditions over the domain. Discretized form of pressure equation is highly nonlinear and is often solved using the Newton-Raphson method in an iterative sequence.

The components of Jacobian matrix for this method are computed at the current iteration level. The residual form of Eq. (3.12) in 2D, 5-point discretization at location i, j can be expressed as,

$$R_{p;i,j}^{v+1} = TX_{t;i-\frac{1}{2}}^{v+1}\left(P_{i-1,j}^{v+1} - P_{i,j}^{v+1}\right) + TX_{t;i+\frac{1}{2}}^{v+1}\left(P_{i+1,j}^{v+1} - P_{i,j}^{v+1}\right) + TY_{t;j-\frac{1}{2}}^{v+1}\left(P_{i,j-1}^{v+1} - P_{i,j}^{v+1}\right) +$$

$$TY_{t;j+\frac{1}{2}}^{v+1}\left(P_{i,j+1}^{v+1} - P_{i,j}^{v+1}\right) - \frac{(V_b \emptyset c_t)_{i,j}^{v+1}}{\Delta t_p}\left(P_{i,j}^{v+1} - P_{i,j}^{n}\right) + q_{w;i,j}^{v+1} + q_{o;i,j}^{v+1} = 0 \quad \text{.....(4.2)}$$

where the superscript $v$ is the iteration level and $n$ is the time level. As for the heat conduction effect, Eq. (4.1) takes residual form as,

$$R_{T;i,j}^{v+1} = TX_{K_T;i-\frac{1}{2}}^{v+1}\left(T_{i-1,j}^{v+1} - T_{i,j}^{v+1}\right) + TX_{K_T;i+\frac{1}{2}}^{v+1}\left(T_{i+1,j}^{v+1} - T_{i,j}^{v+1}\right) + TY_{K_T;j-\frac{1}{2}}^{v+1}\left(T_{i,j-1}^{v+1} - T_{i,j}^{v+1}\right) + TY_{K_T;j+\frac{1}{2}}^{v+1}\left(T_{i,j+1}^{v+1} - T_{i,j}^{v+1}\right) + q_{w;i,j}^{v+1}H_w^{w;v+1} + q_{o;i,j}^{v+1}H_o^{w,v+1} = 0 \quad \text{.....(4.3)}$$

Residual form in Eq. (4.2) requires boundary conditions that are imposed via the well equation. When the bottomhole pressure is specified, Dirichlet-type boundary condition applies and no additional equations are required in Eq. (4.2) because the flow rate can be defined by the existing variables. However, if the flow rate is specified, then Neumann-type boundary condition applies and an additional equation is required,

$$R_{w;i,j}^{v+1} = (T_w^w + T_o^w)_{i,j}^{v+1}\left(P^w - P_{i,j}^{v+1}\right) + q_{t;i,j}^{v+1} = 0 \quad \text{.....(4.4)}$$

Consider the flow in x-direction, the transmissibility terms in Eq. (4.2), (4.3) and (4.4) are defined as,

$$TX_{t;i\pm\frac{1}{2}} = \left(T_g T_t\right)_{i\pm\frac{1}{2}} \quad \text{.....(4.5)}$$

$$T_{g;i\pm\frac{1}{2}} = \frac{2}{\left(\frac{\Delta x}{kA}\right)_i + \left(\frac{\Delta x}{kA}\right)_{i\pm1}} \quad \text{.....(4.6)}$$

$$T_{t;i\pm\frac{1}{2}} = T_{w;i\pm\frac{1}{2}} + T_{o;i\pm\frac{1}{2}} = \left(\frac{k_{rw}\rho_w}{\mu_w}\right)_{i\pm\frac{1}{2}} + \left(\frac{k_{ro}\rho_o}{\mu_o}\right)_{i\pm\frac{1}{2}} \quad \text{.....(4.7)}$$

$$TX_{K_T;i\pm\frac{1}{2}} = \left(\frac{AK_T}{\Delta x}\right)_{i\pm\frac{1}{2}} \quad \text{.....(4.8)}$$

Total phase transmissibility in Eq. (4.7) is evaluated with upstream weighting while the thermal conductivity in Eq. (4.8) is obtained with harmonic averaging.

The set of linear equations in Newton-Raphson method can be expressed as,

$$J^v(\delta X)^v = -R^v \quad \text{.....(4.9)}$$

$$X^{v+1} = X^v + (\delta X)^v \quad \text{.....(4.10)}$$

where $J^v$ is the Jacobian matrix, $R^v$ is the residual matrix, and $X$ is either the unknown pressure or temperature. In this work, we use the linear solver NSPCG (Nonsymmetric Preconditioned Conjugate Gradient) pack to solve Eq. (4.9) (Kincaid et al. 1984). An example illustration of the matrix form in Eq. (4.9) is described as follows.

Assume a 2D reservoir with the uniform 3x3 gridblocks in column-ordered, as shown in **Figure 7**. An injector is placed at the 1st block and a producer is at the 9th block. Flow rate is specified for injector while bottomhole pressure is specified for producer. The resulting matrix based on Eq. (4.9) is developed in **Figure 8**. From the figure, Jacobian matrix structure can be divided into several blocks of 2x2 elements. Additional column and row are included to represent the well equation due to a rate-constrained well condition.

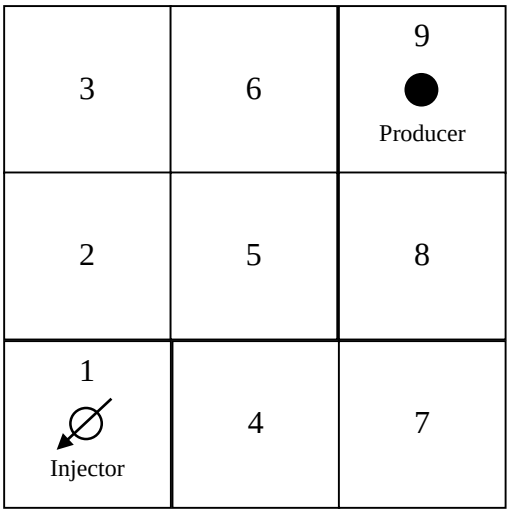



**Figure 7: A sample 3x3 reservoir grid in 2D ordered by columns**

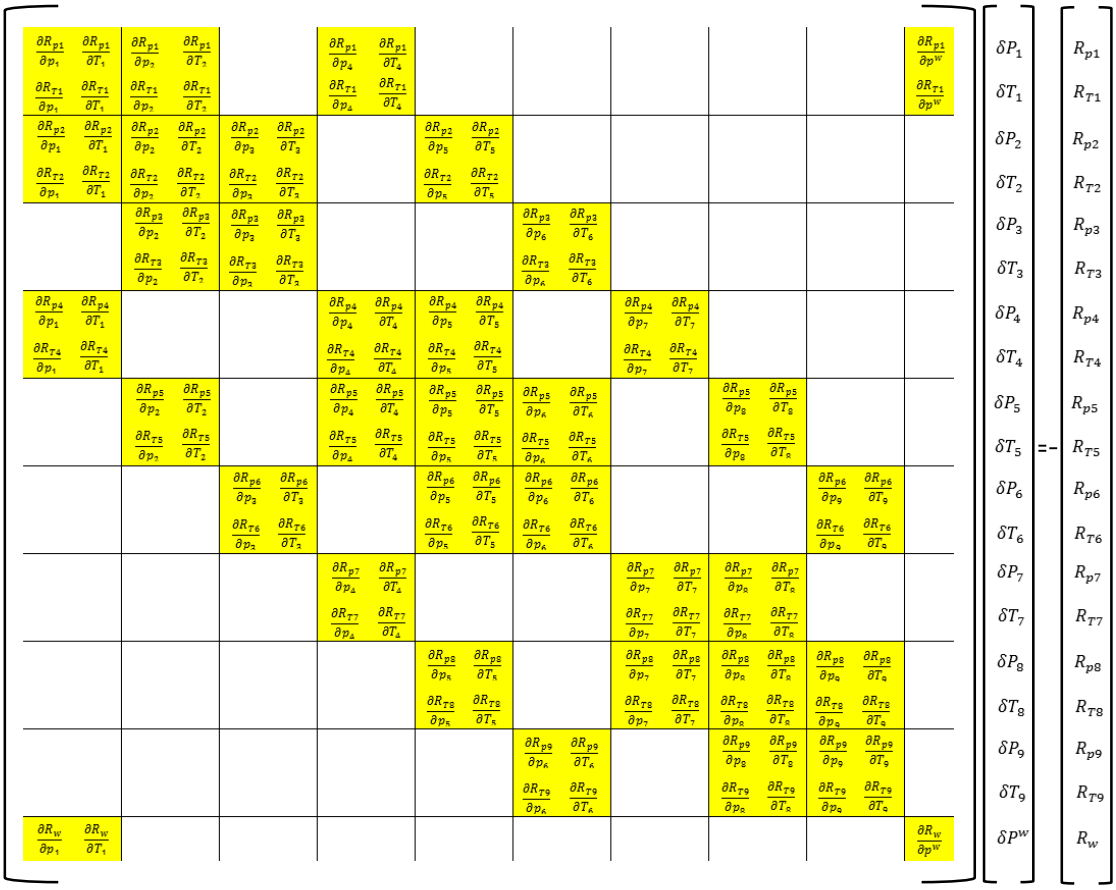


**Figure 8: Corresponding matrix structure of Newton-Raphson method based on the system shown in Figure 7**

The solution in time-of-flight grid is now discussed as follows. By applying the explicit term in time, the discretized forms of mass transport equation in Eq. (3.55) and heat transport equation in Eq. (3.56) can be written as,

$$\frac{1}{\Delta t}\left[(\rho_w S_w)_i^{n+1} - (\rho_w S_w)_i^n\right] + \frac{1}{\Delta \tau}\left[(\rho_w f_w)^n_{i+\frac{1}{2}} - (\rho_w f_w)^n_{i-\frac{1}{2}}\right] = -\left(\frac{a}{\emptyset}\rho_w f_w\right)_i^n \quad \text{.....(4.11)}$$

$$\frac{1}{\Delta t}\left[E_i^{n+1} - E_i^n\right] + \frac{1}{\Delta \tau}\left[F^n_{i+\frac{1}{2}} - F^n_{i-\frac{1}{2}}\right] = -\left(\frac{a}{\emptyset}F\right)_i^n \quad \text{.....(4.12)}$$

where subscript $i$ denotes the time-of-flight grid location, and $n$ is the time level. All interblock quantities in the above equations are evaluated using the upstream weighting method. Both Eq. (4.11) and Eq. (4.12) are solved simultaneously along the streamlines. Local time step that is smaller than global time step is used based on the Courant-Fredrich-Lewy stability criterion (Siavashi et al. 2014),

$$\Delta t = \frac{N_c \Delta \tau}{\max[(v_s^{max})_{sl}^{mass}, (v_s^{max})_{sl}^{heat}]} \quad (4.13)$$

where $N_c$ is the CFL number, $v_s^{max}$ is the maximum shock speed, subscript $sl$ means streamline, and the superscripts $mass$ and $heat$ refers to mass and energy transport equations respectively.

### 4.3 Simulation Data Input

The developed simulator is tested in a 2D heterogeneous reservoir model with a configuration of 200 x 200 x 1 gridblocks. The volume of each gridblock is 3.125 x 3.125 x 6 $m^3$. A quarter five-spot well pattern of hot waterflooding is simulated for 1000 days. An injector is placed at the bottom left (block 1,1,1) and a producer is at the top right (block 200,200,1). Hot water of 80$^o$C is injected with a constant rate of 300 $m^3$/day while the producing well is operated under the minimum bottomhole pressure constraint of 400 kPa. Fluid compressibility is honored in this simulator.

The rock is assumed to be incompressible with a constant porosity of 0.2. At the initial condition, oil saturation is equal to 1, temperature is set to 20$^o$C, and the pressure is at the value of 500 kPa throughout the reservoir. Permeability values are isotropic but non-

uniform, and its distribution can be seen in **Figure 9**. Properties of reservoir rock and fluid are summarized in **Table 1** and **Table 2**, respectively.

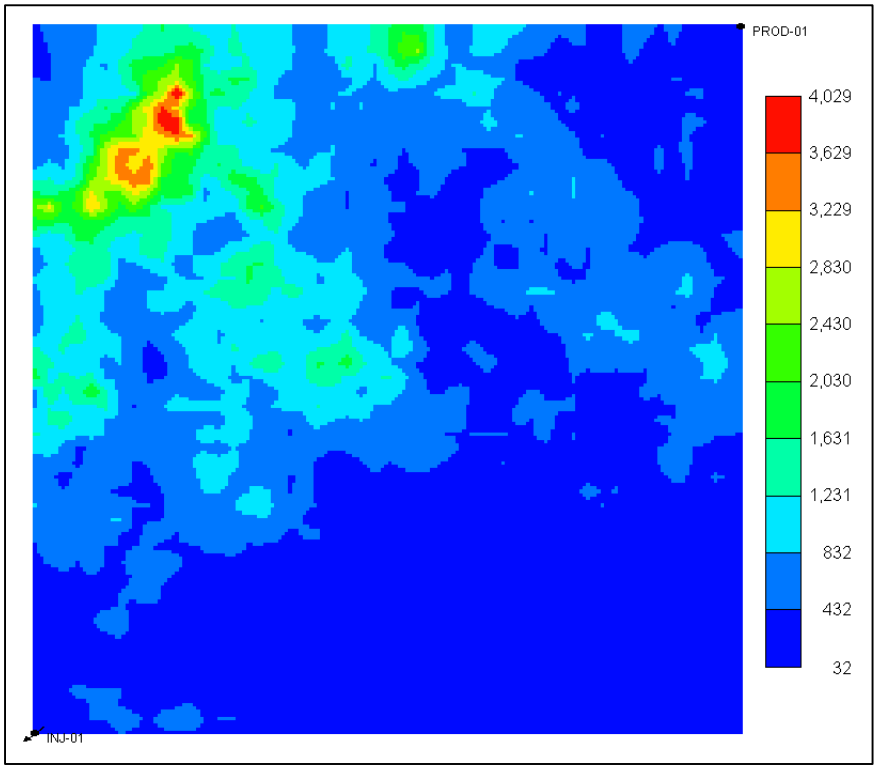



**Figure 9: Permeability distribution of the simulation model**

**Table 1: Reservoir rock properties**

| Property | Value |
|---|---|
| Reservoir area ($m^2$) | 390,625 |
| Thickness (m) | 6 |
| Porosity | 0.2 |
| Rock thermal capacity ($kJ/m^3.K$) | 2300 |
| Rock thermal conductivity (W/m.K) | 3.5 |

**Table 2: Reservoir fluid properties**

| Property | Water | Oil |
|---|---|---|
| **Viscosity Parameters** | | |
| $A_j$ (cp) | 4.735 x $10^{-3}$ | 9.554 x $10^{-3}$ |
| $B_j$ (K) | 1515.7 | 1868.1 |
| **Density Parameters** | | |
| $\rho_j^{ref}$ (kg/m$^3$) | 998 | 972 |
| $c_j$ (1/kPa) | 4.5 x $10^{-7}$ | 7.3 x $10^{-7}$ |
| $\alpha_j$ (1/K) | 3.0 x $10^{-4}$ | 4.0 x $10^{-4}$ |
| **Thermal Properties** | | |
| $K_j$ (W/m.K) | 0.619 | 0.133 |
| $\left(\bar{C}_p\right)_j$ (kJ/kg.K) | 4.19 | 2.02 |
| **Reference Condition** | | |
| $P_{ref}$ (kPa) | 100 | |
| $T_{ref}$ (K) | 293.15 | |

# CHAPTER V

# SIMULATION RESULTS

The reservoir model described in the previous chapter is employed to simulate hot waterflooding using the new approach of streamline simulation. CMG STARS, a commercial finite volume thermal simulator is used to validate the results from our developed simulator. Another important note is to compare the results with operator splitting technique, which is currently the general method to incorporate heat conduction effect in most streamline simulators. Streamlines are updated every 50 days until it reaches the simulation end time of 1000 days. From the injection well block, 100 streamlines are launched to solve the mass and heat transport equation as shown in **Figure 10**.

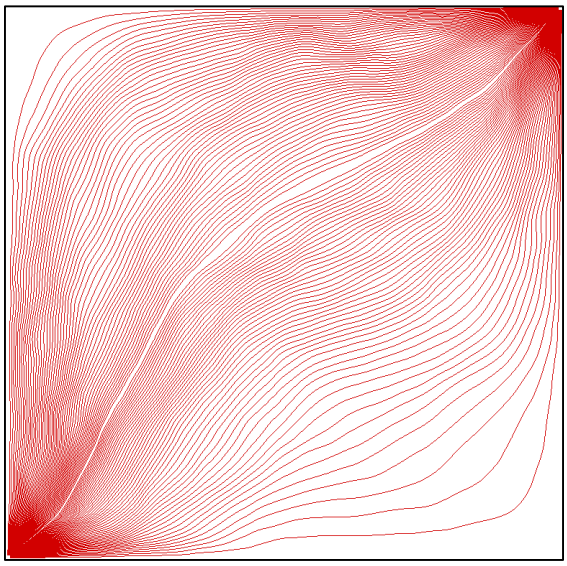
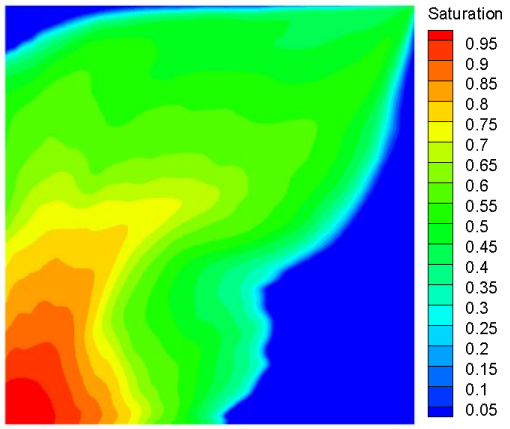
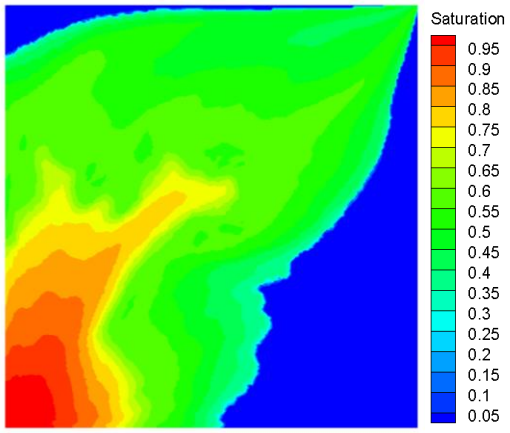


**Figure 10: Trace of streamlines from injector to producer**

CMG STARS




Streamline: New Approach




Streamline: Operator Splitting

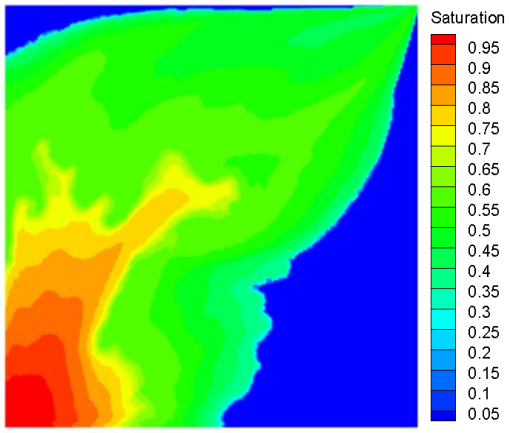
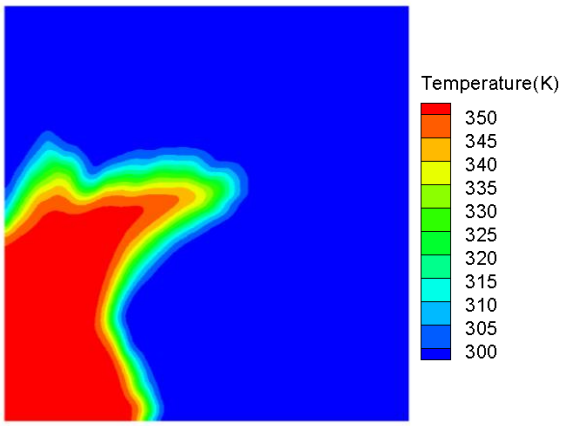
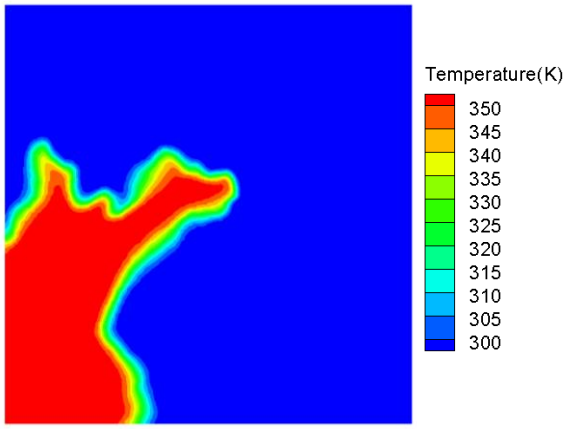



**Figure 11: Comparison of the computed water saturation between simulators after 1000 days**

CMG STARS




Streamline: New Approach




Streamline: Operator Splitting

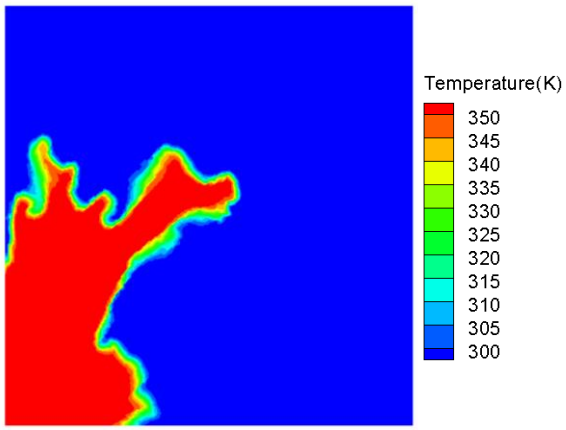



**Figure 12: Comparison of the computed temperature between simulators after 1000 days**

As can be seen in **Figure 10**, streamlines tend to cluster at the top right area where the permeability is relatively high. This is one of the typical behavior in streamline simulation where it provides higher resolution along preferential flow paths (Datta-Gupta and King 2007). **Figure 11** compares the water saturation distribution from all simulators. The streamline simulators produce a quite similar saturation pattern to STARS but with smeared front. This might be due to numerical dispersion from the mapping procedure in streamline method and high heterogeneity of permeability. From **Figure 12**, temperature front is seen to move slower than the saturation which is caused by the high thermal capacity of the rock.

Based on the temperature distribution, our approach matches better to STARS than the operator splitting method. Heat conduction is well known to make the temperature front retarded (Pasarai 2007). Our approach yields smooth progression of temperature front whereas the operator splitting generates sharp front with small "fingers". This indicates that the new method is more suitable to handle heat conduction than operator splitting.

Oil production rate and water cut are represented through **Figure 13** and **14** and are in agreement with the reference CMG STARS with error around 5.2% and 6.3%, respectively. Although the temperature front difference between our approach and operator splitting can be recognized in **Figure 12,** their difference in production profile is almost negligible. All simulators are run on Intel Xeon 3.7 GHz CPU with 32 GB RAM. The commercial simulator takes 24.2 minutes to finish the simulation. Run time and speedups of the streamline simulators are found in **Table 3**.

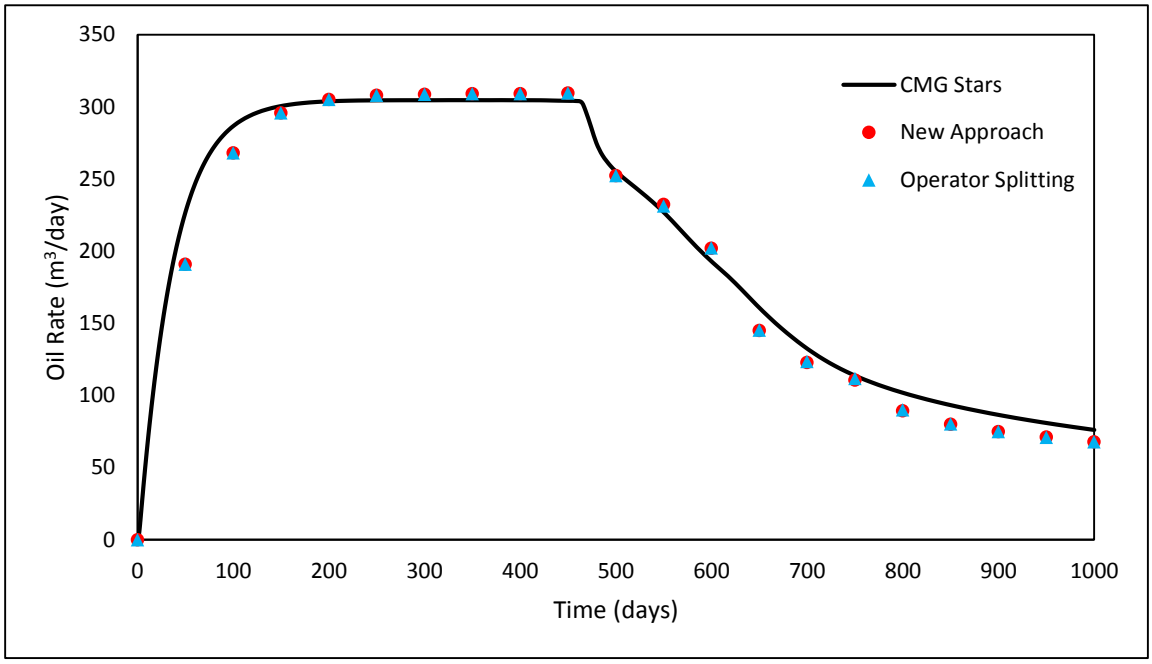



**Figure 13: Plot of the oil production rate over time**

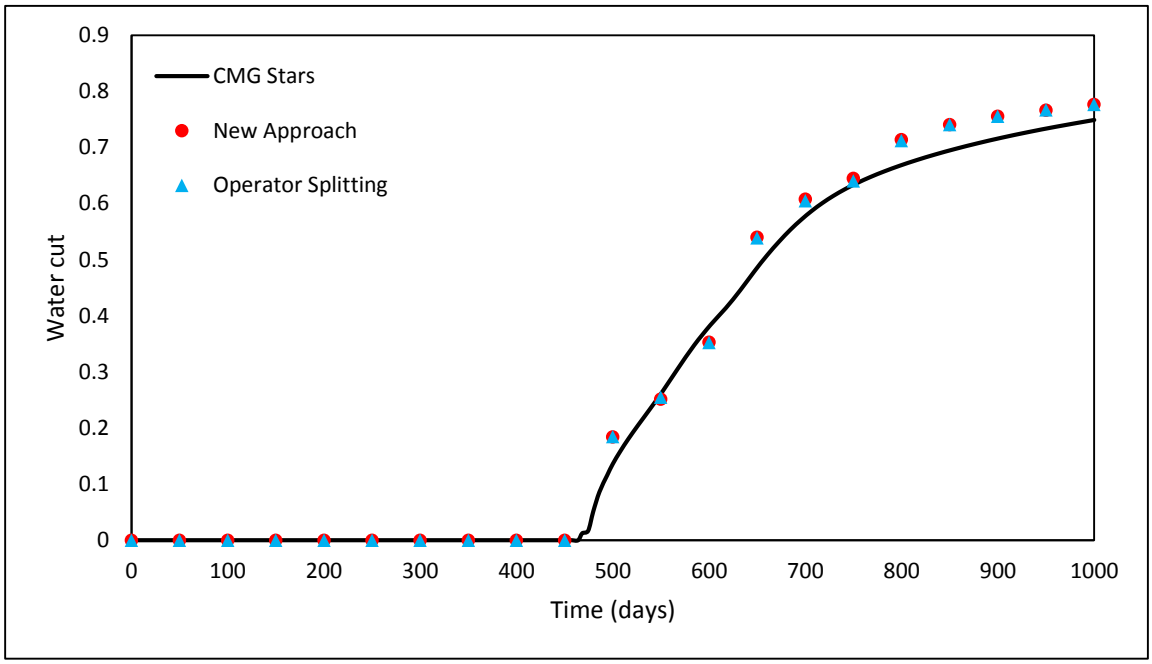



**Figure 14: Plot of the water cut over time**

Both streamline simulators run much faster than CMG STARS with a speedup factor around 2. However, our new approach is slightly slower than the operator splitting technique. Because we solve the heat conduction simultaneously with pressure equation, the set of linear equation system becomes large and more complex. This will require more computational effort than the operator splitting which decouples the heat conduction from pressure equation. Nevertheless, the higher computational cost is compensated by better accuracy. From Table 3, it can be seen that our new approach provides better result with less error than the operator splitting. The error term here is computed as the relative L2 norm. Thus, the trade-off is speed versus accuracy.

**Table 3: Simulator performance**

| Property | New Approach | Operator Splitting |
|---|---|---|
| Run time | 11.33 | 10.52 |
| Speedup factor | 2.1 | 2.3 |
| $S_w$ error at 1000 days (%) | 7.58 | 9.24 |
| $T$ error at 1000 days (%) | 1.66 | 2.51 |

Another thing to be considered here is the temperature error difference between the two streamline simulators. Even though the error is reduced in our new approach, it is relatively small decrement. This can be caused by the fact that hot waterflooding process relies mainly on the convective flow. Therefore, change of conduction treatment will not affect the overall system too much. In the future, this new approach can be implemented

to a case where heat conduction plays an important role such as reservoir with heat loss to overburden or the steamflooding case. It will be interesting to see the relationship between speed and accuracy of the new approach in these cases.

Speed and accuracy of the simulation solution also depend on the number of streamlines and pressure updates. **Table 4** gives an idea of how the number of streamlines affects the simulation. Run time of CPU is the lowest at 100 streamlines, and it increases if we try to reduce or add more streamlines. A large number of streamlines is undoubtedly producing high run time. On the other hand, a small number of streamlines is also time-consuming because more missed gridblocks are needed to be traced back. Thus, 100 streamlines is the optimum case for our study in terms of speed. Regarding accuracy, adding more streamlines tend to improve the solution even though the change is not that significant. For the case of 200 streamlines, the error in temperature is slightly higher than in the case of 150 streamlines. This is a consequence of the mapping error that comes with a large number of streamlines.

**Table 4: Simulator performance at various number of streamlines with 20 pressure updates**

| **Property** | **SL = 25** | **SL = 50** | **SL = 100** | **SL = 150** | **SL = 200** |
|---|---|---|---|---|---|
| Run time | 15.20 | 12.74 | 11.33 | 17.65 | 23.41 |
| $S_w$ error at 1000 days (%) | 7.84 | 7.67 | 7.58 | 7.50 | 7.46 |
| $T$ error at 1000 days (%) | 1.78 | 1.70 | 1.66 | 1.62 | 1.64 |

The sensitivity of simulation to the pressure updates is given in **Table 5**. By updating pressure more often, the CPU time is apparently rising, and the accuracy is improved since we are capturing the temporal velocity field better in each pressure update. On the other hand, frequent pressure updates can increase the error due to the frequent mapping. In this case, 20 pressure updates give the optimum speed and accuracy. Although 40 pressure updates give slightly better accuracy, it is not worth the double CPU run time.

**Table 5: Simulator performance at various pressure updates with 100 streamlines**

| **Property** | **PU = 2** | **PU = 5** | **PU = 10** | **PU = 20** | **PU = 40** |
|---|---|---|---|---|---|
| Run time | 5.90 | 8.84 | 10.62 | 11.33 | 23.57 |
| $S_w$ error at 1000 days (%) | 16.18 | 12.56 | 9.10 | 7.58 | 7.04 |
| $T$ error at 1000 days (%) | 5.01 | 3.19 | 2.45 | 1.66 | 1.54 |

# CHAPTER VI

# CONCLUSIONS AND FUTURE WORK

## 6.1 Conclusions

In this thesis, a new streamline simulator is developed using a new approach to heat conduction treatment. Instead of solving the heat conduction at the end of global time step using the common operator splitting technique, it is solved simultaneously with the pressure equation along the gridblocks. The simulator is used to model hot waterflooding process in a 2D heterogeneous reservoir.

Results from the new streamline simulator are in good agreement with commercial thermal finite volume simulator, CMG-STARS. In addition, the new approach is more accurate than the operator splitting method. From the map of water saturation and temperature, the new approach generates smoother front compared to operator splitting which creates some small artificial “fingers”. Therefore, our new approach captures the heat conduction effect better than the operator splitting method.

The run time of the new approach is slightly slower than the operator splitting. This is due to the fact that it has to solve a larger size of linear equation systems along the gridblocks for both pressure and heat conduction. However, the new approach is still two times faster than the commercial simulator. Thus, our new streamline simulator is still very efficient in simulating thermal reservoir model.

Evaluation of the number of streamlines and pressure updates is also discussed in this study. We can achieve higher accuracy with a large number of streamlines and more

frequent pressure updates. But, keep in mind that generating more streamlines or updating velocity field too often can lead to an increase in error from mapping. Additionally, higher speedup can be achieved with a moderate number of streamlines and less frequent pressure updates. In short, we need to consider the tradeoff between speed and accuracy to obtain the optimum number of streamlines and pressure updates.

### 6.2 Future Work

The improvement of accuracy in this new approach may not be too significant in the hot waterflooding process due to the convection-dominated condition. It will be even better to implement the new approach to the cases where heat conduction plays an important role such as a reservoir with heat losses to underburden and overburden. We expect higher accuracy of the new approach compared to operator splitting in this kind of case because our new approach captures the heat conduction effect better. The speedup of streamline simulator can also be higher when we perform code optimization.

The developed simulator in this study is based on several key assumptions. To generate more realistic results comparable to real reservoir condition, we need to extend it to 3D, 3-phase, multicomponent simulator with gravity and capillary pressure effect. The relative permeability can also be made dependent on temperature, and the porosity is assumed to be changing over time. Lastly, if we already enhance the simulator in the hot waterflooding case, then it is time to perform the new approach on other thermal enhanced oil recovery processes such as steamflooding, SAGD, in-situ combustion, and many more. Modifications to the new approach may be needed as it can become problem-dependent.

# REFERENCES


Alboudwarej, H., J. Felix, S. Taylor et al. 2006. Highlighting Heavy Oil (in *Oilfield Review Summer 2006* **18** (2): 34-53.

Aziz, K., A. Settari. 1979. *Petroleum Reservoir Simulation*, 1st edition, 164-165. London, England, Applied Science Publishers Ltd. (Reprint).

Batycky, R. P. 1997. A Three-Dimensional Two-Phase Field Scale Streamline Simulator. Ph.D. Dissertation, Stanford University.

Bear, J. 1972. *Dynamics of Fluids in Porous Media*, 1st edition. New York, USA, Elsevier (Reprint).

Cheng, H., I. Osako, A. Datta-Gupta et al. 2005. A Rigorous Compressible Streamline Formulation for Two- and Three-Phase Black Oil Simulation. Proc., SPE Annual Technical Conference and Exhibition, Dallas, Texas.

Crane, M., F. Bratvedt, K. Bratvedt et al. 2000. A Fully Compositional Streamline Simulator. Proc., SPE Annual Technical Conference and Exhibition, Dallas, Texas.

Curtis, C., R. Kopper, E. Decoster et al. 2002. Heavy-Oil Reservoirs (in *Oilfield Review Autumn 2002* **14** (3): 30-51.

Datta-Gupta, A., M. J. King. 1995. A Semianalytical Approach to Tracer Flow Modeling in Heterogeneous Permeable Media (in *Advances in Water Resources* **18**: 9-24.

Datta-Gupta, A., M. J. King. 2007. *Streamline Simulation: Theory and Practice*, 1st edition. Richardson, Texas: SPE Textbook Series, Society of Petroleum Engineers (Reprint).

Farouq Ali, S. M. 1974. Heavy Oil Recovery - Principles, Practicality, Potential, and Problems. Proc., SPE Rocky Mountain Regional Meeting, Billings, Montana.

Kasrale, M., P. H. Sammon, P. J. Jespersen. 1993. Field Development Options for a Waterflooded Heavy-Oil Reservoir (in *Journal of Petroleum Technology* **45**: 888-894.

Kincaid, D., T. Oppe, J. Respess et al. 1984. *ITPACKV 2C User's Guide* (Reprint).

King, M. J., A. Datta-Gupta. 1998. Streamline Simulation: A Current Perspective (in *In Situ* (22): 91-140.

Lof, H., M. Gerritsen, M. Thiele. 2008. Parallel Streamline Simulation. Proc., EAGE Conference and Exhibition.

Mallison, B. T., M. Gerritsen, S. F. Matringe. 2004. Improved Mappings for Streamline-Based Simulation. Proc., SPE 14th Symposium on Improved Oil Recovery, Tulsa, Oklahoma.

Osako, I., A. Datta-Gupta, M. J. King. 2003. Timestep Selection during Streamline Simulation via Transverse Flux Correction. Proc., SPE Reservoir Simulation Symposium, Houston, Texas.

Pasarai, U. 2007. Development of Streamline-Based Simulators for Evaluation of Heavy Oil Recovery. Ph.D. Dissertation, Waseda University, Tokyo.

Pollock, D. W. 1988. Semianalytical Computation of Path Lines for Finite Difference Methods (in *Ground Water* **26**: 743-750.

Siavashi, M., M. J. Blunt, M. Raisee et al. 2014. Three-Dimensional Streamline-Based Simulation of Non-Isothermal Two-Phase Flow in Heterogeneous Porous Media (in *Computers & Fluids* **103**: 116-131.

Tamim, M., K. H. Abou-Kassem, S. M. Farouq Ali. 1999. Recent Developments in Numerical Simulation Techniques of Thermal Recovery Processes. Proc., SPE International Thermal Operations and Heavy Oil Symposium, Bakersfield, California.

Yan, W., M. L. Michelsen, E. H. Stenby et al. 2004. Three-Phase Compositional Streamline Simulation and Its Application to WAG. Proc., SPE 14th Symposium on Improved Oil Recovery, Tulsa, Oklahoma.

Zhu, Z., M. Gerritsen, M. Thiele. 2010. Thermal Streamline Simulation for Hot Waterflooding (in *SPE Reservoir Evaluation & Engineering* **13** (03): 372-382.