

**STREAMLINE TRACING AND TIME OF FLIGHT DIAGNOSTICS FOR
WATERFLOODING OPTIMIZATION: THEORY AND APPLICATION**

A Thesis

by

RONGQIANG CHEN

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,	Akhil Datta-Gupta
Committee Members,	Michael J. King
	Debjyoti Banerjee
Head of Department,	A. Daniel Hill

December 2015

Major Subject: Petroleum Engineering

Copyright 2015 Rongqiang Chen

ABSTRACT

The objective of this work is to demonstrate the power and utility of streamline-based methods for flow diagnostics and waterflooding optimization. The robustness and high efficiency of the new post-simulation streamline tracing tool allows engineers to expand the use of streamline for flow diagnostics. Waterflooding is widely used for enhanced oil recovery, but the presence of high reservoir heterogeneity often leads to premature water breakthrough and low sweep efficiency. It eventually reduces field oil recovery. Streamline-based approaches have been widely used for reservoir management. Because of streamline method's ability to depict fluid front, it is very effective for waterflooding reservoir management. The new tracing tool is more robust in handling non-neighbor connection, and its fast computation makes flow diagnostic analyses applicable for large scale field cases. In this research, the application of streamline flow diagnostic plots in 2D synthetic case, 3D benchmark case, and 3D field case are demonstrated to evaluate reservoir and waterflooding performance. Furthermore, the use of flow diagnostic plots is extended to test and compare the performance of several streamline-based waterflooding optimization methods: streamline rate allocation using well allocation factors, streamline equalizing arrival time, and streamline-based NPV. The testing is conducted in 2D synthetic case and 3D benchmark case. All optimization cases tested support that streamline-based NPV method has the best performance in optimizing field NPV and is robust in all stages of the field operation.

DEDICATION

To my parents and my wife

For their love and support

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Datta-Gupta, and my committee members, Dr. King and Dr. Banerjee, for their guidance and support throughout the course of this research.

Thanks also go to MCERI alumni and current students for their mentorship, partnership and friendship throughout my study life.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks go to my mother and father for their encouragement and to my wife for her patience and love.

NOMENCLATURE

A	=	area, ft^2
b_α	=	reciprocal of formation volume factor of phase α , $\alpha = o$ or w
B_α	=	formation volume factor of phase α , $\alpha = o$ or w , bbl/STB
c	=	divergence of the velocity field, $c = \nabla \cdot u_t$
e_{ip}	=	efficiency between injector i to producer p , dimensionless
F_α	=	convective fractional flow of phase α , $\alpha = o, g, w$ dimensionless
$k_{r\alpha}$	=	relative permeability of phase α , dimensionless
L	=	length, ft
N_{st}	=	number of streamlines traced
Q_α	=	volumetric rate of phase α at surface condition, bbl/day
S_α	=	saturation of phase α , dimensionless
t	=	time, day
Ψ_{i+}	=	intercell transmissibility of geometric part, $bbl.md.ft^2/cp/psi$
u_α	=	velocity of phase α , ft/day
v_α	=	interstitial velocity of phase α , ft/day
V	=	volume, ft^3
ϕ	=	porosity, dimensionless
λ_α	=	relative phase mobility of phase α , cp^{-1}
μ_α	=	viscosity of phase, cp

τ	=	time-of-flight, <i>day</i>
τ_{ip}	=	average fluid arrival time from injector <i>i</i> to producer <i>p</i> , <i>day</i>
τ_{fm}	=	average field fluid arrival time, <i>day</i>
Λ_{ip}	=	discounted value between injector <i>i</i> to producer <i>p</i> , dimensionless
η	=	net to gross ratio, dimensionless
ψ, χ	=	bi-streamfunctions
ξ	=	streamline trajectory
WAF	=	well allocation factor
$EqAT$	=	equalizing arrival time
$SLNPV$	=	streamline net present value

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
NOMENCLATURE	v
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
LIST OF TABLES	xii
CHAPTER I INTRODUCTION	1
1.1 Background and Literature Review	2
1.2 Objective and Thesis Outline	6
CHAPTER II STREAMLINE TRACING AND TIME OF FLIGHT CALCULATION	7
2.1 Time of Flight Coordinate	7
2.2 Streamline for Compressible Fluid	10
2.3 Single Cell Streamline Tracing	11
2.4 1D Equation and Fractional Flow	16
CHAPTER III STREAMLINE TRACING AND FLOW DIAGNOSTIC	18
3.1 Streamline Tracing Workflow	18
3.2 Tracing Performance Comparison	21
3.3 Application of Streamline Flow Diagnostic	28
3.3.1 2D Five Spot Case	28
3.3.2 Brugge Benchmark Case	30
3.3.3 Bijupira Field Case	33
3.3.4 Husky Field Case	39
CHAPTER IV STREAMLINE-BASED WATERFLOODING OPTIMIZATION	44
4.1 Background and Theory	44
4.1.1 Streamline-based Well Allocation Factor Optimization	46
4.1.2 Streamline-based Equalizing Arrival Time Optimization	49

4.1.3 Streamline-based NPV Optimization	50
4.2 Application	53
4.2.1 2D Five Spot Synthetic Case.....	54
4.2.2 2D Areal Multi-well Case	59
4.2.3 Brugge Benchmark Case	62
CHAPTER V CONCLUSION AND RECOMMENDATIONS.....	65
5.1 Conclusions	65
5.2 Recommendations	66
REFERENCES	68
APPENDIX ANALYSIS OF DESTINY6 NNC CALCULATION	71
A.1 Uni-directional NNC Flow	71
A.1.1 Destiny 6 Algorithm	71
A.1.2 Local Boundary Layering.....	73
A.1.3 Stream Function Solutions	74
A.2 Reversal NNC Flow	77
A.2.1 Destiny 6 Algorithm	78
A.2.2 Stream Function Solutions	80

LIST OF FIGURES

	Page
Fig. 1 - Relationship between TOF and streamline trajectory.	8
Fig. 2 - Streamline pore volume.	9
Fig. 3 - Streamline tracing workflow.	18
Fig. 4 - The comparison between cell center tracing method (left) and resampling tracing method (right).	21
Fig. 5 - Brugge streamline tracing results for old (left) and new (right) tracing algorithms.	22
Fig. 6 - Brugge streamline tracing trajectory near a fault (1).	23
Fig. 7 - Brugge streamline tracing trajectory near a fault (2).	24
Fig. 8 - Brugge streamline tracing trajectory near a fault (3) – cross-section view (i = 75).	25
Fig. 9 - Husky streamline tracing results for old (left) and new (right) tracing algorithms.	26
Fig. 10 - Workflow of non-neighbor connection streamline tracing.	26
Fig. 11 - 2D 5 spot waterflooding case permeability (left) and porosity (right).	29
Fig. 12 - 2D 5 spot waterflooding flow diagnostic plots.	30
Fig. 13 - Brugge permeability, porosity, and initial water saturation field.	31
Fig. 14 - Flow diagnostic plots for the Brugge benchmark case in line view (left) and grid view (right).	32
Fig. 15 - Bijupira permeability and porosity field.	33
Fig. 16 - Bijupira flow diagnostic plots for reservoir characterization (line view).	34
Fig. 17 - Bijupira flow diagnostic plots for reservoir characterization (grid view).	35
Fig. 18 - Bijupira flow diagnostic plots for reservoir surveillance showing well partition in line view (top) and grid view (bottom).	36

Fig. 19 - Combing an injector partition and a producer partition to obtain a well pair (line view).....	37
Fig. 20 - Bijupira injector/producer well allocation map.	39
Fig. 21 – Husky permeability, porosity, and initial water saturation field.....	40
Fig. 22 - Husky flow diagnostic plots for reservoir characterization (line view).	40
Fig. 23 - Husky flow diagnostic plots for reservoir characterization (grid view).....	41
Fig. 24 - Husky flow diagnostic plots for reservoir surveillance showing well partition.....	42
Fig. 25 - Husky injector/producer well allocation map.....	43
Fig. 26 - Sensitivity of α on weight factor.....	48
Fig. 27 - The schematic of five spot waterflooding configuration (base case).	55
Fig. 28 - Permeability (left), porosity (middle), and water saturation (right) distribution.....	56
Fig. 29 - The history of injection rate of 4 strategies: (a) base (b) WAF (c) SLNPV (d) EqAT.....	57
Fig. 30 - The comparisons of net present value (left) and recovery factor (right) for 4 different strategies (2D 5 Spot).....	58
Fig. 31 - Streamline and water saturation distribution for 4 different scenarios at 300 days (5SP).....	58
Fig. 32 – Multi-well case well configuration and streamline producer partition.	59
Fig. 33 – Multi-well case permeability, porosity, and initial water saturation field.	60
Fig. 34 – Multi-well case NPV efficiency plot of base case (left) and SLNPV case (right).	61
Fig. 35 - The comparisons of net present value (left) and recovery factor (right) for 4 different strategies (2D Multi-well).....	62
Fig. 36 – Brugge NPV efficiency plot of base case (left) and SLNPV case (right).....	63
Fig. 37 - The comparisons of net present value (left) and recovery factor (right) for 4 different strategies (Brugge).....	63

Fig. 38 - Streamline and water saturation distribution for 4 different scenarios at 4,500 days (Brugge).....	64
Fig. 39 – A streamline leaving a cell with multiple NNCs	72
Fig. 40 - A streamline entering a cell with multiple NNCs.....	73
Fig. 41 - A zoom-in view of a local unit.	76
Fig. 42 - NNC reversal problem for limitations of Destiny 6 algorithm.....	78
Fig. 43 - NNC reversal problem solved by local boundary layering.	79
Fig. 44 – NNC Reversal flow problem for non-uniqueness in stream function solutions.....	80

LIST OF TABLES

	Page
Table 1 - Comparison between line view and grid view streamline diagnostic plots.	20
Table 2 - Tracing computational time comparison (Brugge).	22
Table 3 - Tracing computational time comparison (Husky).	27
Table 4 - Parameters used for waterflooding optimization.	54

CHAPTER I

INTRODUCTION

Streamline simulation is a fast flow simulation technique for modeling fluid flow in high resolution reservoirs. It has been proved to be complementary to conventional finite difference simulation. Streamline simulation provides great computational efficiency over the finite difference simulation because it decouples the multi-dimensional flow equations into a series of 1D calculations along streamlines. It has been successfully implemented for waterflooding optimization (Grinestaff 1999, Alhuthali et al. 2000, Thiele and Batycky 2003 & 2006, Giordano 2007, Datta-Gupta 2007, Moyner et al. 2014) and assisted history matching (Wang and Kovscek 2000, Agarwal and Blunt 2001, Milliken et al. 2001, Wu and Datta-Gupta 2002, Stenerud and Lie 2004, Stenerud et al. 2008, Yin et al. 2010). Currently, numerical finite difference simulators are still the most widely used simulation tools for reservoir engineers because of its capability of integrating all available data and solving all the complex physics. The main objective of reservoir engineers is to integrate all available data to provide a development strategy that will optimize oil and gas recovery while trying to reduce the risks and uncertainties associated with the approach. While finite difference simulation makes a great tool for integrating all data, streamline simulation can be used as a post-simulation tool to evaluate reservoir performance and to generate an optimal future development strategy.

In addition to the regular reservoir engineering uses, one of the added features for streamlines is to help engineers visualize reservoir heterogeneity as well as its

performance. It can be used to capture the swept and drainage volume, establish injector-producer well pair relationship, understand waterflooding allocation, predict water breakthrough, validate upscaled models, rank stochastic geo-models, as well as to optimize field wide water injection strategy (Gilman et al. 2002, Thiele and Batycky 2003, Al-Khalifa 2004, Datta-Gupta and King 2007).

In this research, a newer version of post-simulation tool (“Destiny 6”) was used. It has a better streamline tracing algorithm and flux-based non-neighbor connection (NNC) tracing which make all flow diagnostic plots feasible in a timely manner. While it reduces the computational time significantly, “Destiny 6” still uses very flexible algorithms for streamline tracing. “Destiny 6” will be used to generate flow diagnostic plots for reservoir characterization, reservoir surveillance, and waterflooding optimization.

1.1 Background and Literature Review

Streamline simulation has been used as a fast simulation technique to model convection dominated fluid flow for a long time and it can be implemented in almost all stages of a field development. Besides its fast simulation, flow visualization is another important feature of streamlines. The literature on streamline application is voluminous. Streamline application ranges from validating and ranking upscaling models, identifying swept and drainage volume, and waterflooding management and optimization. Due to the development of the new tracing algorithm, we can generate all the flow diagnostic plots within a timely manner and utilized those plots for optimization purposes.

At least four technologies have been developed before streamline to model convection dominated flow. There are line-source/sink, streamtube, particle tracking, and stream-function (Datta-Gupta and King 2007).

A very important development in streamline is the concept of time of flight introduced by Datta-Gupta and King (2007). With this concept, the 3D problem of saturation calculations can be decoupled to 1D transport equations along the streamline within the time of flight coordinate. This coordinate transformation makes the solution not restricted by the geologic grid-based CFL criteria (Lax and Wendroff 1960, Coats 2003), allowing larger time step and leading to faster simulation.

Tracing streamlines within a grid cell is based on Pollock's method (1988), which suggests piece-wise linear interpolation of the velocity field within a grid block. Pollock's method is reliable because it conserves the material balance equation. Although the original Pollock's method is implemented only in orthogonal grids, it was later extended to more general corner point grids.

Streamlines have two main applications, which are:

- 1) Fast convective flow forward simulation
- 2) Post-simulation reservoir management and assisted history matching

In this research, the focus will be on the second application: post-simulation reservoir management, and we will compare the results for existing methods for waterflooding optimization.

Many streamline projects are undertaken by researchers and industrial experts, and some of these flow diagnostic projects are listed below:

- 1) Swept volume and drainage volume calculation: as time of flight is directly related to fluid movement and reservoir heterogeneity, mapping TOF on the streamline or grid provides an intuitive representation of the swept volume and drainage volume (Datta-Gupta 2000 & 2007). Even though time of flight has a unit of time, it can be viewed as a distance. Time of flight is used as a spatial coordinate in streamline simulation. It is the distance along streamline to where the reservoir will be contacted. If a specific value is provided to filter or cutoff time of flight, it will capture where the reservoir has been contacted by production or injection. It is important to note that this swept or drainage area is not uniform because TOF is calculated by using the fluxes, which already take reservoir heterogeneity into account.
- 2) Reservoir partitioning: for post-simulation purposes, streamline tracing and TOF calculation can be conducted for each individual grid cell. It can be used to depict reservoir flow partitioning. While finite difference simulation focuses on where the fluid is and its composition, streamline can gather information to identify for each grid cell where the fluid is coming from and where the fluid is going to. The information provided by streamlines is more intuitive, and this information can be plotted in injector and producer partition plots. It will help reservoir engineers to perform production rate or waterflooding optimization.
- 3) Rate allocation and waterflooding optimization: several approaches have been tested for rate allocation and waterflooding optimization. TOF can be calculated for each grid cell, and since TOF is a good property to assess reservoir

heterogeneity, incorporated with Lorenz coefficient, one can get a good correlation between these heterogeneity and oil recovery (Moyner, Krostad, and Lie 2014). Thus, rate can be adjusted based on Lorenz coefficient. Some other approaches include streamline-based rate allocation optimization, equalizing arrival time, and streamline-based NPV method (Thiele and Batycky 2003 et al. 2007, Tanaka 2014).

- 4) History Matching incorporated with dynamic production data: reservoir engineers forecast production based on static reservoir models which come from well logs, cores, and seismic analyses. History matching is a process to integrate static data with dynamic data, which consists of production history, well testing results as well as production logging results. History matching approaches can be divided into two main categories: one is deterministic approach, and another is stochastic approach. Stochastic method typically requires multiple initial static models and history matching updates these models to find global minimum of the solution space (Tanaka 2014). Streamline-based method is a deterministic method that utilizes a gradient based approach. Streamline tracing and time of flight calculation not only help engineers visualize reservoir performance, but also these properties are directly influenced by the permeability field. This direct relationship can be an effective tool for assisted history matching, and lots of previous work about streamline-based assisted history matching can be found (Agarwal and Blunt 2001, Emanuel and Chakravarty 2001, Milliken et al. 2004, Cheng et al. 2007, Tanaka 2014). Streamline-based history matching relies on the sensitivity of static

data, in our case permeability to the production data and the water breakthrough time. By matching the water breakthrough time, one can achieve a better reservoir model.

1.2 Objective and Thesis Outline

The objective of this study is to develop flow diagnostic plots for reservoir characterization and reservoir surveillance using the new version of post-simulation streamline tracing tool (Destiny 6). Because Destiny 6 is very robust in terms of non-neighbor connection streamline tracing, being able to generate all flow diagnostic plots using this semi-analytical solution is meaningful. Also, its computational efficiency makes it applicable for any field cases, especially for high resolution field cases. After all the plots are successfully implemented in Destiny 6, several waterflooding optimization approaches will be tested using the available diagnostic plots. We will compare the performance among different optimization approaches, including well allocation and streamline NPV methods.

This work includes an overview of the literature and previous work. Chapter II of this thesis discusses the theory and concept of streamline tracing and time of flight calculation. Chapter III provides details about streamline application to reservoir characterization, which includes TOF, drainage and swept volume, and streamline application to reservoir surveillance, which consists of well allocation and injector efficiency plots. Chapter IV will compare the performance among different waterflooding optimization methods.

CHAPTER II

STREAMLINE TRACING AND TIME OF FLIGHT CALCULATION

2.1 Time of Flight Coordinate

The fundamental theory of streamline methods is the coordinate transformation from physical space to coordinate system following the flow direction, which is also called the time of flight coordinate. Before going into details about streamline flow equations, one needs to review the transformation based on bi-streamfunctions. Introducing bi-streamfunctions, ψ , and χ , we construct a velocity field, \vec{u} , as:

$$\vec{u} = \nabla \psi \times \nabla \chi \dots\dots\dots(2.1)$$

It is assumed that there is no velocity divergence on streamline in conventional streamline methods, in which no velocity divergence can be expressed by the vector identity $\nabla \cdot (\nabla \psi \times \nabla \chi) = 0$. Thus, a streamline is defined by the intersection of a constant ψ with a constant χ . For 2D application, $\psi = \psi(x, y)$, $\chi = z$, and ψ becomes the streamfunction. The time of flight, τ , is defined as the travel time of a neutral tracer along the streamlines, which is shown as

$$\tau(x, y, z) = \int \frac{\phi}{|\vec{u}|} ds \dots\dots\dots(2.2)$$

The schematic for the relationship between time of flight and streamline trajectory is shown in Fig. 1.

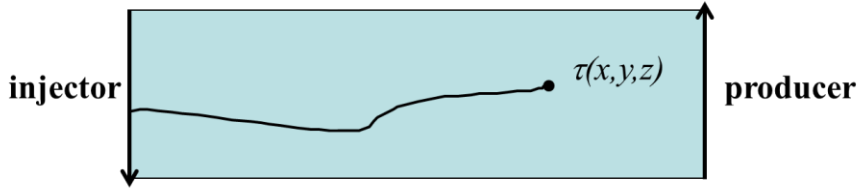


Fig. 1 - Relationship between TOF and streamline trajectory.

This coordinate transformation from physical space to time of flight coordinate can be obtained from streamline in which lines are traced through grid blocks based on an underlying velocity field (Tanaka 2014). Eq. 2.2 can be rewritten as $\vec{u} \cdot \nabla \tau = \phi$, and the following equation can be derived (Datta-Gupta and King 2007)

$$\left\| \frac{\partial(\tau, \psi, \chi)}{\partial(x, y, z)} \right\| = \nabla \tau \cdot (\nabla \psi \times \nabla \chi) = \nabla \tau \cdot \vec{u} = \phi \dots\dots\dots(2.3)$$

The relationship between the physical space and the time of flight coordinate can be shown as

$$d\tau d\psi d\chi = \phi dx dy dz \dots\dots\dots(2.4)$$

Using the above relationship, one can successfully perform the coordinate transformation.

A streamline is often described and visualized as a line in the reservoir. However, it does have its associated pore volume and the derivation is shown below. The derivation is under the assumption of incompressible flow system, and the flow rate is constant along a streamline. We first define q_{sl} as the total flow rate of a streamline and $u_t(s)$ as the total velocity at an arbitrary location, s , along a streamline. Therefore, the cross-sectional area of a streamline at location s can be written as

$$A_{sl}(s) = \frac{q_{sl}}{|u_t(s)|} \dots\dots\dots(2.5)$$

Multiplying porosity to both sides of Eq. 2.5, the equation becomes

$$A_{sl}(s)\phi(s) = q_{sl} \frac{\phi(s)}{|u_t(s)|} \dots\dots\dots(2.6)$$

Integrating Eq. 2.6 with its location s, we will get

$$\int_0^s A_{sl}(s)\phi(s)ds = \int_0^s q_{sl} \frac{\phi(s)}{|u_t(s)|} ds = q_{sl} \int_0^s \frac{\phi(s)}{|u_t(s)|} ds = q_{sl}\tau \dots\dots\dots(2.7)$$

The left side of Eq. 2.7 shows the integrated value of the cross-sectional area from 0 to s, which is the pore volume of a streamline. Thus, the pore volume of a streamline can be shown in Fig. 2 and written as

$$PV_{sl} = q_{sl}\tau \dots\dots\dots(2.8)$$

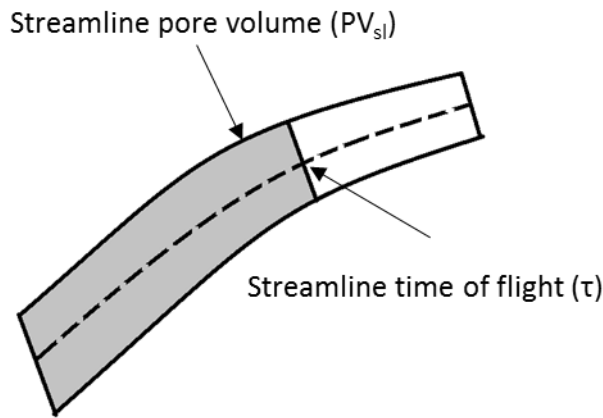


Fig. 2 - Streamline pore volume.

2.2 Streamline for Compressible Fluid

The concept of effective density was introduced to handle changes of fluid volumes along streamline (Cheng et al. 2006). This concept makes streamline applicable for compressible fluids simulation. Effective density is defined by modifying Eq. 2. 1 and redefining the streamfunctions

$$\rho \vec{u} = \nabla \psi \times \nabla \chi \dots\dots\dots(2.9)$$

$$\nabla \cdot (\rho \vec{u}_t) = 0 = \phi \frac{\partial \rho}{\partial \tau} + \rho \nabla \cdot \vec{u} \dots\dots\dots(2.10)$$

Thus, $\nabla \cdot u_t$ can be rewritten as

$$\nabla \cdot \vec{u}_t = -\frac{\phi}{\rho} \frac{\partial \rho}{\partial \tau} \dots\dots\dots(2.11)$$

If ρ is a constant, $\nabla \cdot u_t = 0$ recovers the standard formulation for incompressible fluids.

For compressible fluids, ρ is not a constant value. Using Pollock's assumption, velocities within a cell vary linearly in their respective directions, so the divergent of u_t can be expressed as follows

$$\nabla \cdot \vec{u}_t = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} \dots\dots\dots(2.12)$$

For compressible flow, there is no requirement for $\nabla \cdot u_t = 0$; however, with Pollock's approach, $\nabla \cdot u_t$ is a constant value. The equation for compressible fluids can be rewritten as

$$c = -\frac{\phi}{\rho} \frac{\partial \rho}{\partial \tau} \dots\dots\dots(2.13)$$

This equation can be easily integrated into the following form

$$\rho = \rho_0 e^{-c\tau/\phi} \dots\dots\dots(2.14)$$

where ρ_0 is the reference effective density, which is usually set at the injector well grid.

By setting $\rho_0 = 1.0$, relative fluid volumes can be calculated along streamline as follows

$$\rho_i = \rho_0 e^{-c_i \tau_i / \phi_i} \dots\dots\dots(2.15)$$

$$q_i = q_0 \cdot \frac{1}{\rho_i} \dots\dots\dots(2.16)$$

2.3 Single Cell Streamline Tracing

Streamlines are defined as instantaneous curves locally tangential to the fluid velocity vector. It is important to note that there is no restriction on whether the fluid is compressible or incompressible. Tracing streamlines within a single grid cell is based on the analytical solution introduced by Pollock (1988). The assumption Pollock made is that the velocity in each direction within a grid cell is linearly distributed and is independent of the velocities of other directions within this cell. This velocity model can be expressed using the numerical solutions shown as follows

$$\begin{aligned} u_x &= u_{x1} + c_x(x - x_1) \\ u_y &= u_{y1} + c_y(y - y_1) \\ u_z &= u_{z1} + c_z(z - z_1) \dots\dots\dots(2.17) \end{aligned}$$

The relationship between time of flight and velocity within a grid block can be expressed as

$$\frac{d\tau}{\phi} = \frac{dx}{u_x} = \frac{dy}{u_y} = \frac{dz}{u_z} \dots\dots\dots(2.18)$$

The time of flight can be calculated explicitly by integrating Eq. 2.18, and each direction can be calculated independently. The integral solution in the x-direction from x_0 can be written as

$$\frac{\Delta\tau_{xi}}{\phi} = \int_{x_0}^{x_i} \frac{dx}{u_{x0} + c_x(x - x_0)} = \frac{1}{c_x} \ln\left(\frac{u_{xi}}{u_{x0}}\right) \text{ when } c_x \neq 0 \dots\dots\dots(2.19)$$

$$\frac{\Delta\tau_{xi}}{\phi} = \frac{x_i - x_0}{u_{x0}} \text{ when } c_x = 0 \dots\dots\dots(2.20)$$

where index $i = 1, 2$ indicates the two faces in the x-direction. The time of flight in the y- and z-directions can be solved using the same construction, and the actual time of flight in a single cell is given by the minimum positive value shown as

$$\Delta\tau = \text{Min Positive} (\Delta\tau_{x1}, \Delta\tau_{x2}, \Delta\tau_{y1}, \Delta\tau_{y2}, \Delta\tau_{z1}, \Delta\tau_{z2}) \dots\dots\dots(2.21)$$

Once the time of flight is known, the streamline exit coordinate can be obtained by rearranging Eq. 2.19 and 2.20

$$x = x_0 + u_{x0} \left(\frac{e^{c_x \Delta\tau / \phi} - 1}{c_x} \right) \text{ when } c_x \neq 0 \dots\dots\dots(2.22)$$

$$x = x_0 + u_{x0} \Delta\tau \text{ when } c_x = 0 \dots\dots\dots(2.23)$$

The original Pollock's method is limited to only orthogonal grid, but later this approach is extended to corner point grids (Cordes and Kinzelbach 1992, Prevost,

Edwards, and Blunt 2002). To trace streamlines in corner point grids, a dimensionless grid system needs to be introduced, in which the relationship between dimensionless coordinate and physical coordinate system can be expressed as

$$\alpha = x/DX \quad \beta = y/DY \quad \gamma = z/DZ \quad \dots\dots\dots(2.24)$$

where α, β, γ stands for dimensionless coordinate, and x, y, z stands for the physical ones.

Meanwhile, the directional Darcy velocities are converted to volumetric fluxes by multiplying the cross-sectional areas shown as

$$\begin{aligned} Q_x &= u_x \cdot DY \cdot DZ \\ Q_y &= u_y \cdot DX \cdot DZ \\ Q_z &= u_z \cdot DX \cdot DY \quad \dots\dots\dots(2.25) \end{aligned}$$

The volumetric flux in each direction within a grid cell is linearly distributed using a simple linear interpolation for velocity.

Dividing Eq. 2.18 by the volume of the cell, we obtain

$$\frac{d\tau}{\phi \cdot DX \cdot DY \cdot DZ} = \frac{d\alpha}{Q_x(\alpha)} = \frac{d\beta}{Q_y(\beta)} = \frac{d\gamma}{Q_z(\gamma)} \quad \dots\dots\dots(2.26)$$

Cordes and Kinzelbach provided a simple and elegant generalization of Eq. 2.26 for computing streamline trajectory and the time of flight in corner point grids based on two assumptions:

1. Linearly interpolate volumetric flux instead of velocity
2. Use the Jacobian instead of cell volume to relate flux and velocity

$$\frac{d\tau}{\phi \cdot J(\alpha, \beta, \gamma)} = \frac{d\alpha}{Q_1(\alpha)} = \frac{d\beta}{Q_2(\beta)} = \frac{d\gamma}{Q_3(\gamma)} \quad \dots\dots\dots(2.27)$$

This above equation can be re-written as

$$\begin{aligned} \phi \frac{d\alpha}{d\tau} &= \frac{Q_1(\alpha)}{J(\alpha, \beta, \gamma)} \\ \phi \frac{d\beta}{d\tau} &= \frac{Q_2(\beta)}{J(\alpha, \beta, \gamma)} \\ \phi \frac{d\gamma}{d\tau} &= \frac{Q_3(\gamma)}{J(\alpha, \beta, \gamma)} \end{aligned} \dots\dots\dots(2.28)$$

The volumetric fluxes are linearly interpolated between the corresponding face fluxes as follows

$$\begin{aligned} Q_j(\alpha_j) &= a_j + c_j \cdot \alpha_j \quad j=1,2,3 \\ a_j &= Q_{j1} \quad c_j = Q_{j2} - Q_{j1} \quad j=1,2,3 \end{aligned} \dots\dots\dots(2.29)$$

The simplifying notations will be used $\{Q_j | j=1,2,3\} = \{Q_x, Q_y, Q_z\}$ and $\{\alpha_j | j=1,2,3\} = \{\alpha, \beta, \gamma\}$.

Ideally, Eq. 2.18 can be integrated to compute the time of flight and trajectories, but unfortunately, it is much more difficult to integrate than for the rectangular cells because all three parameters are coupled in the Jacobian. Jimenez et al. (2010) simplified this process by introducing a time-like parameter T, called the pseudo-time of flight.

$$dT = \frac{1}{\phi} \frac{d\tau}{J(\alpha, \beta, \gamma)} = \frac{d\alpha}{Q_1(\alpha)} = \frac{d\beta}{Q_2(\beta)} = \frac{d\gamma}{Q_3(\gamma)} \dots\dots\dots(2.30)$$

Similar to Pollock's method, these equations can be integrated explicitly, and independently in each direction. The integral solution in the α -direction can be shown as

$$\int_0^{T_E} dT = \int_{\alpha_0}^{\alpha} \frac{d\alpha}{Q_1(\alpha)} = \int_{\alpha_0}^{\alpha} \frac{d\alpha}{a_1 + c_1 \cdot \alpha} = \frac{1}{c_1} \ln \left[\frac{a_1 + \alpha \cdot c_1}{a_1 + \alpha_0 \cdot c_1} \right] \text{ when } c_1 \neq 0 \dots\dots\dots(2.31)$$

$$\int_0^{T_E} dT = \int_{\alpha_0}^{\alpha} \frac{d\alpha}{Q_1(\alpha)} = \int_{\alpha_0}^{\alpha} \frac{d\alpha}{a_1 + c_1 \cdot \alpha} = \frac{\alpha - \alpha_0}{a_1} \text{ when } c_1 = 0 \dots\dots\dots(2.32)$$

The pseudo-time of flight in the β - and γ -directions can be solved using the same construction, and the actual time of flight in a single cell is given by the minimum positive value shown as

$$\Delta T = \text{Min Positive}(\Delta T_{x1}, \Delta T_{x2}, \Delta T_{y1}, \Delta T_{y2}, \Delta T_{z1}, \Delta T_{z2}) \dots\dots\dots(2.33)$$

Once the pseudo-time of flight T is calculated, the streamline exit coordinate can be calculated as

$$\alpha_e = \alpha_0 + (a_1 + \alpha_0 c_1) \left(\frac{e^{c_1 T} - 1}{c_1} \right) \text{ when } c_1 \neq 0 \dots\dots\dots(2.34)$$

$$\alpha_e = \alpha_0 + a_1 T \text{ when } c_1 = 0 \dots\dots\dots(2.35)$$

After knowing the unit space coordinates (α, β, γ) , we can use tri-linear interpolation to transform the unit coordinates to the physical coordinate (x, y, z) .

In corner point grids, T is a more convenient parameter for determining streamline trajectories than τ . To within constant scaling factors, the equations for $\alpha(T)$, $\beta(T)$, $\gamma(T)$ are now identical to Pollock's equations in a three dimensional rectangular cell. Thus, we can calculate τ from the integral

$$\tau = \phi \int_0^T J(\alpha(T), \beta(T), \gamma(T)) dT \dots\dots\dots(2.36)$$

where α , β , and γ are all known functions of T . Each parameter will depend on T through $(e^{cT} - 1)/c$, and the Jacobian is a polynomial in α , β , and γ . The result is a sum of exponentials and constants, which can be integrated analytically.

2.4 1D Equation and Fractional Flow

Streamline simulation uses the transformation of Buckley-Leverett concept to solve for saturation along the streamline (Bratvedt et al. 1992, Datta-Gupta and King 2007, Tanaka 2014) after it solves the pressure implicitly. For 2-phase oil water incompressible flow, neglecting the effects of gravity and capillarity, the saturation can be solved easily using the following equation

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (F_w \vec{u}_t) = 0 \dots\dots\dots(2.37)$$

where F_w is the fractional flow term and will be discussed later in detail. The equation can be expanded into following form

$$\nabla \cdot (F_w \vec{u}_t) = \vec{u}_t \cdot \nabla F_w + F_w \nabla \cdot \vec{u}_t \dots\dots\dots(2.38)$$

while the system is incompressible ($\nabla \cdot \vec{u}_t = 0$), the equation can be further simplified and rewritten as (Datta-Gupta and King 2007)

$$\nabla \cdot (F_w \vec{u}_t) = \vec{u}_t \cdot \nabla F_w = \phi \frac{\partial F_w}{\partial \tau} \dots\dots\dots(2.39)$$

Thus, the material balance equation can be written as

$$\frac{\partial S_w}{\partial t} + \frac{\partial F_w}{\partial \tau} = 0 \dots\dots\dots(2.40)$$

However, for large scale field cases, some of these assumptions may not be valid, and the 1D solution usually involves operator splitting, anti-diffusive correction, or orthogonal projection methods to take into account the gravity, and capillarity effects (Bratvedt et al. 1996, Datta-Gupta and King 2007, Tanaka 2014). For the use of Destiny 6, since we do not perform any forward modeling, there is no need to get involved into

this complex saturation solver. Instead, we map the saturation solved by commercial simulators on streamlines. However, in order to perform waterflooding optimization, it is required to calculate the phase flow rate within each streamline. To achieve this goal, fractional flow needs to be calculated. We use the saturation mapped from the commercial simulators to calculate the fractional flow as

$$F_w(S_w) = \frac{\lambda_w}{\lambda_t} = \frac{\frac{k_{rw}}{\mu_w}}{\frac{k_{rw}}{\mu_w} + \frac{k_{ro}}{\mu_o}} \dots\dots\dots(2.41)$$

With the fractional flow, we are able to obtain the phase flow rate along a streamline using

$$q_w = F_w \cdot q_t \dots\dots\dots(2.42)$$

Destiny 6 uses this information to generate flow diagnostic plots such as flood efficiency and streamline-based NPV plots as discussed later.

CHAPTER III

STREAMLINE TRACING AND FLOW DIAGNOSTIC

3.1 Streamline Tracing Workflow

As mentioned above, streamline tracing is used as a post-simulation processing tool to evaluate reservoir and waterflooding performance in this research. The schematic workflow is shown in Fig. 3.

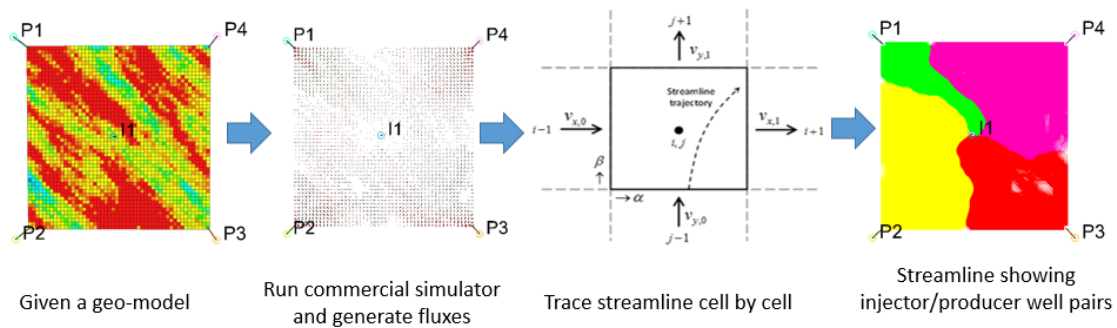


Fig. 3 - Streamline tracing workflow.

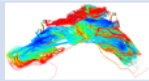
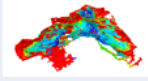
Given a reservoir geo-model, we first run a reservoir simulation using commercial simulation software (Eclipse). This process will solve for pressure, saturation, as well as flux field at each time step. The fluxes are properties at the cell faces, as shown in the second figure from the left in Fig. 3. It is important to note that all the fluxes will be available after this step. In other words, we have individual phase flux (oil / water / gas) fields after this step. Summing up fluxes for all phases, we will have the total fluid flux, which is also most commonly used for streamline tracing and visualization. With flux information, streamlines are traced based on Pollock's assumption from a source to a sink

or from a sink to a source depending on user's need. Meanwhile, time of flight can be calculated. Flow diagnostic plots can provide valuable visual understanding of reservoir performance and fluid movement for decision makers. Several types of diagnostic plots are currently available, and they are divided into two main categories: reservoir characterization and reservoir surveillance. Reservoir characterization focuses on reservoir heterogeneity. It includes TOF to producer, TOF to injector, total TOF, as well as swept and drainage volume. On the other side, reservoir surveillance focuses on the connections between wells and waterflooding performance. It includes well pairs, well allocation, and injector efficiency. Besides these plots above which uses total flux, one can also trace individual phase fluid movement. It can help identify individual phase fluid movement without the presence of saturation information. The detail for each plot will be shown and demonstrated later in this chapter.

One thing to make Destiny 6 more advanced than its previous versions is the computational efficiency. In the previous version of Destiny, for high resolution cases, streamlines and diagnostic plots are only available in line view, meaning that streamline tracing can only starts from a sink or a source. With this approach, it is possible that some parts of the reservoir do not have any streamline passed through them, and this part of the reservoir cannot be seen even though they are active. Because of the highly efficient tracing algorithm and its robustness in handling Non-neighbor Connections (NNCs), current Destiny 6 extends streamline diagnostic plots to grid view, especially for high resolution field cases. A detailed comparison between line view and grid view streamline diagnostic plots are shown in Table 1. For highly heterogeneous reservoirs, streamline

distribution is very sparse. If only line view diagnostic plots are available, engineers may be misled by the visualization effects. The current Destiny allows users to trace streamlines from the center of the cells. With this approach, properties can be calculated for every active grid in the model. Grid view diagnostic plots make sure engineers see the whole picture, not just the high fluid flow/movement regions.

Table 1 - Comparison between line view and grid view streamline diagnostic plots.

Streamline View	Properties stored on	Streamline tracing starts from	Coverage	Resolution	Advantages	Example
Line View	Streamline	Injector/producer	Majority of the cells	Number of streamline	Relative fast	
Grid View	Corner point grid	Center of each grid	All active cells	Grid resolution	Ready to use, wider coverage	

As the size of simulation models gets bigger, tracing from each individual cells becomes very challenging. An approximation method is used to calculate time of flight on each cell, which is called the resampling method. The idea is that if a cell has been passed through by a streamline, it records the TOF, and we will not start a new streamline tracing from this cell. The comparison between the cell center tracing method and this resampling method can be seen in Fig. 4. After streamline 1 has been successfully traced, the cell center tracing method will start the second streamline on cell (3,1) while the resampling method will start on cell (4,1) because it knows cell (2,1) has been passed through by streamline 1 and its TOF has been calculated. It will not start a streamline from any cell

that has been passed by previous streamlines, so this method will significantly reduce the number of streamlines, making mapping TOF on multi-million cell field models possible.

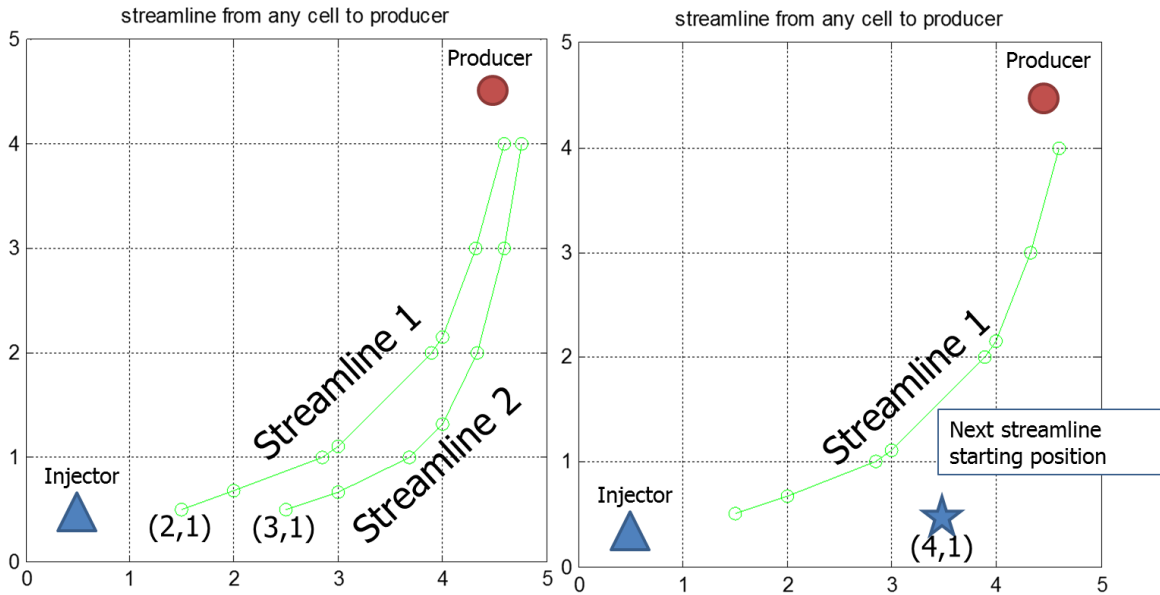


Fig. 4 - The comparison between cell center tracing method (left) and resampling tracing method (right).

3.2 Tracing Performance Comparison

As mentioned several times in the above sections, current Destiny 6 is a much faster post-simulation streamline tracing tool because it has a new tracing algorithm and a more efficient flux-based algorithm to solve streamline tracing in non-neighbor connections (NNCs). Several cases are tested to compare tracing performance.

The first case is the Brugge benchmark model. This model was presented at SPE Applied Technology Workshop in 2008. Its intent is to test waterflooding optimization and history matching methods in a closed loop workflow. For this research purpose, it

serves as a medium resolution pseudo-field case to test tracing performance for two versions of tracing tools. This model has a total of 30 wells: 20 producers and 10 injectors. This model has approximately 40,000 active cells and 155 NNCs. The number of streamlines is chosen to be 1,000, and the streamline tracing and the time of flight calculation results are shown in Fig. 5.

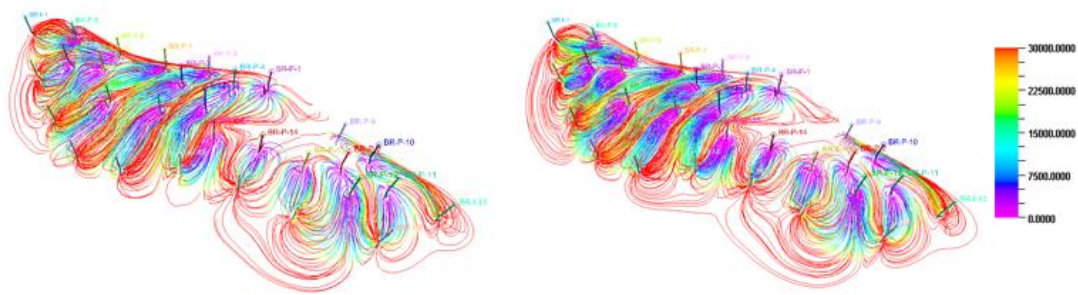


Fig. 5 - Brugge streamline tracing results for old (left) and new (right) tracing algorithms.

It is noted that there is no significant difference between the old and new streamline trajectory and time of flight, but there is a significant improvement in computational time for the new method as shown in Table 2.

Table 2 - Tracing computational time comparison (Brugge).

	Old Tracing	New Tracing
SL tracing computational time for each time step	20 sec	1 sec

To make sure streamlines in NNCs is accurately traced, we compared the Destiny 6's result with that of the local boundary layering method, which is a rigorous way to model streamline tracing in NNCs (Jimenez et al. 2010). The detailed analyses of two methods are discussed in Appendix A. Fig. 6 shows the tracing trajectory for Brugge at an early time step, and there is a transmissible fault in the red circle, which is our targeted study area.

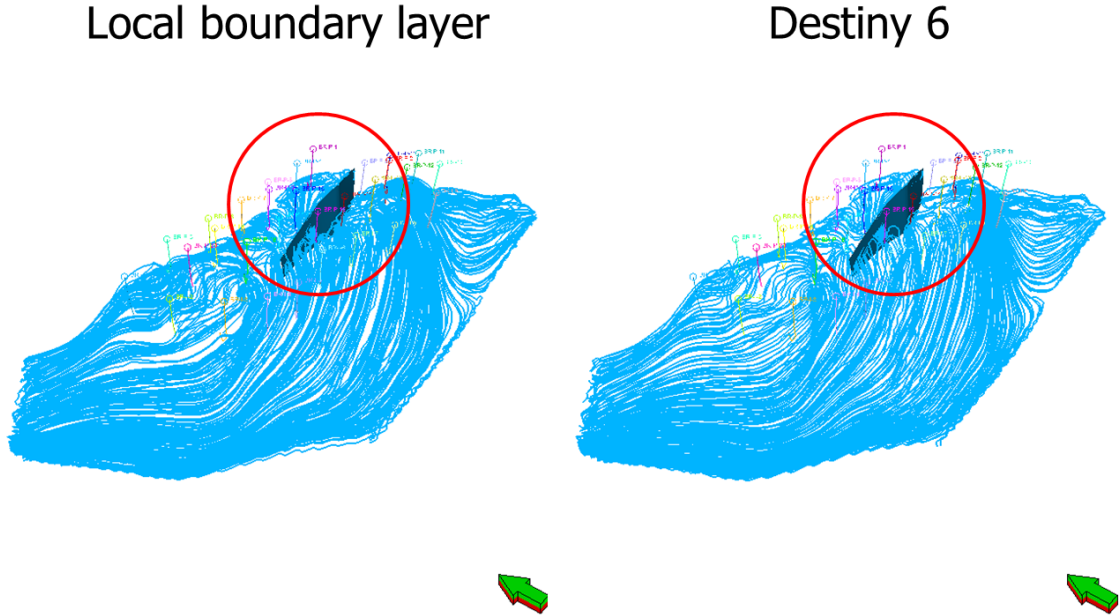


Fig. 6 - Brugge streamline tracing trajectory near a fault (1).

When we zoom in a little more, we notice both methods achieve reasonably similar results as shown in Fig. 7. Both methods preserve the fact that this fault is transmissible so streamlines can pass through this fault.

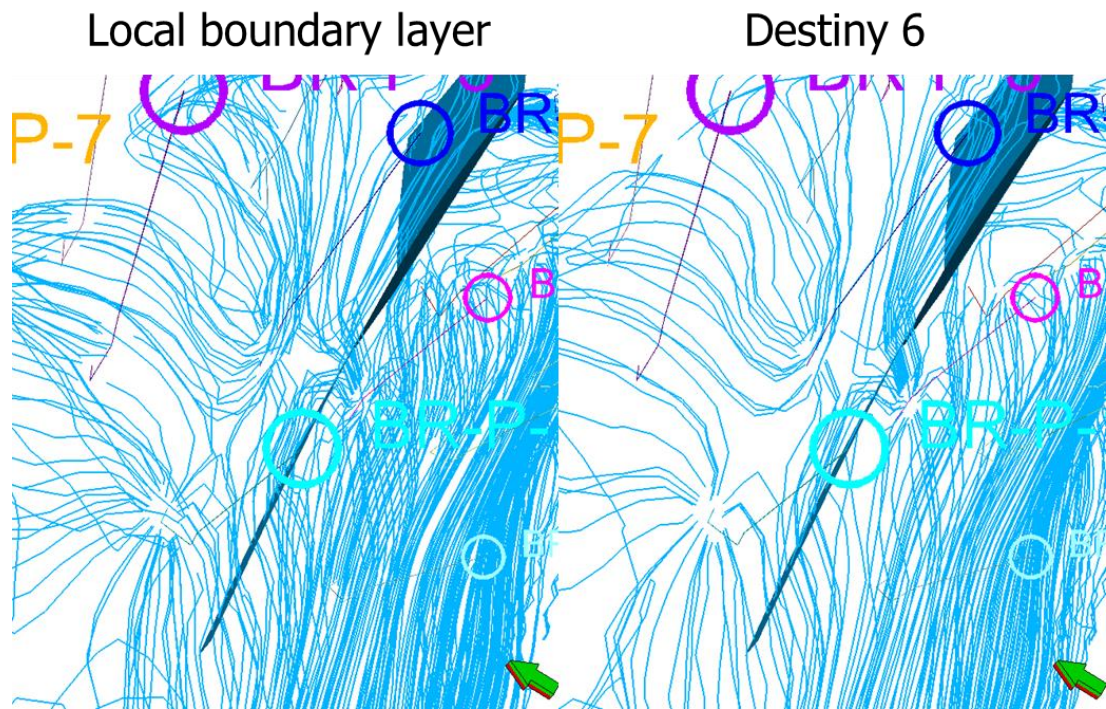


Fig. 7 - Brugge streamline tracing trajectory near a fault (2).

We further zoom in to make sure the streamlines are connecting to the right cells shown in Fig. 8. It is noted that some of the biggest NNC fluxes are in layer $I = 75$. The top view is the cross-section at $I=74$ and the bottom view is at $I = 76$. Even though the entering and exiting coordinates of two methods are not exactly the same, and the differences can be caused by the different starting positions of the streamlines chosen by two tools, both methods have fluxes entering and exiting at the correct cells based on the NNC information. As can be seen, there are streamlines connecting $(75,14,1)$ to $(75,13,7)$ and connecting $(75,14,2)$ to $(75,13,8)$ highlighted in red, and these streamlines are validated with NNC fluxes.

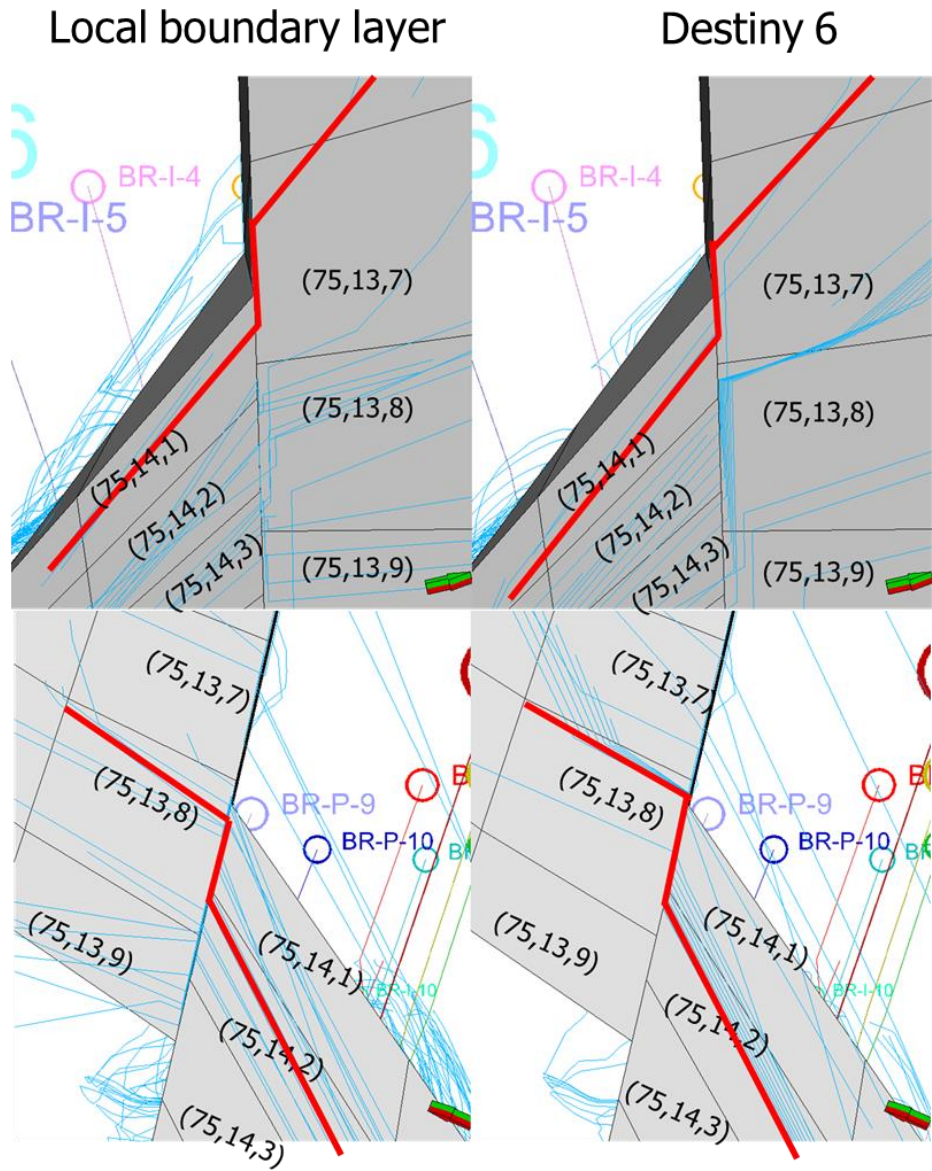


Fig. 8 - Brugge streamline tracing trajectory near a fault (3) – cross-section view (i = 75).

With the success of implementing new Destiny 6 on the Brugge benchmark case, we are confident to extend its use on a real field case, with higher grid resolution and more complicated grid geometry. The Husky model is the chosen model. It has 4,000,000 active

cells, and it is highly heterogeneous with 1,000,000 NNCs. The streamline tracing results for Husky are shown in Fig. 9.

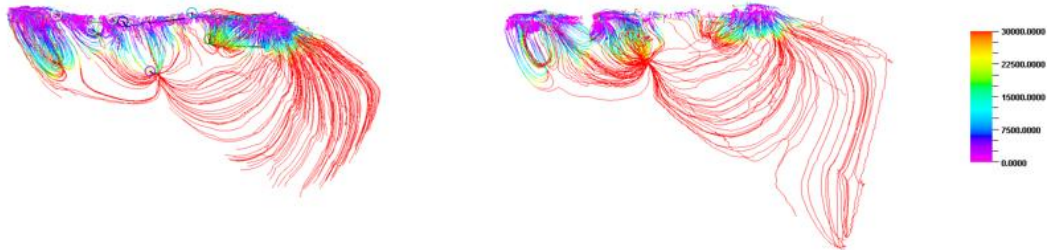


Fig. 9 - Husky streamline tracing results for old (left) and new (right) tracing algorithms.

As one can tell, there is quite a bit of difference on streamline distribution. This may be caused by the methods of handling non-neighbor connections. The workflow for new NNC streamline tracing is shown in Fig. 10.

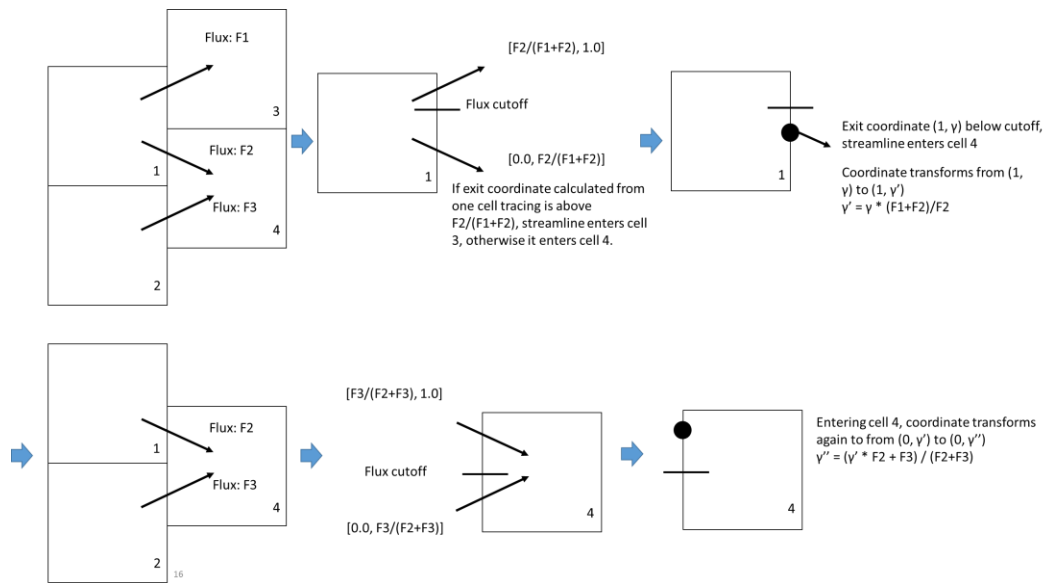


Fig. 10 - Workflow of non-neighbor connection streamline tracing.

If the software identifies there is a NNC in cell 1 with F1 amount of flux going into cell 3 and F2 amount of flux going into cell 4, it first calculates the flux cutoff which is the ratio of F1 and F2. After we perform single cell streamline tracing, if γ is below the cutoff $F1/(F1+F2)$, the streamline enters cell 4. Here, we need to perform a coordinate transformation to get proportional coordinate at the lower part of the cutoff, where $\gamma' = \gamma * (F1+F2)/F2$. Then, we notice F2 and F3 are going into cell 4 at the face where this streamline enters, and we calculate the flux cutoff for this two fluxes. Finally we do a coordinate transformation again to get the new entering coordinate, where $\gamma'' = (\gamma' * F2 + F3) / (F2+F3)$. It is noticed that we carry the unit cell exit coordinate to the new cell. While in the past, once a streamline leaves a cell, it calculates the physical coordinate and deletes the unit cell exit coordinate. Flux cutoff is only used before a streamline enters a cell and/or after a streamline leaves a cell, and within single cell tracing the flux is still assumed to be constant in a face, so this approximation does not violate Pollock's assumption.

In the past, the area cutoff is used instead of the flux cutoff. In extreme case, when zero flux going to large area and large flux going small area, streamlines are very likely to terminate because they easily enter cells with small or no flux.

Table 3 - Tracing computational time comparison (Husky).

	Old Tracing	New Tracing
SL Tracing Computational Time for each time step	~900 sec	~10 sec

Besides the difference in streamline trajectory, the computational efficiency for Husky improves significantly as shown in Table 3. The speedup is more appealing than Brugge due to the larger mesh size and higher number of non-neighbor connections. Thus, we expect to see more acceleration with larger number of grids and more NNCs.

3.3 Application of Streamline Flow Diagnostic

The post-simulation streamline tracing and flow diagnostic tool (Destiny 6) is tested with a series of synthetic examples as well as field examples. These case studies support that Destiny 6 is a reliable tool for streamline tracing and flow diagnostic plots, and it helps engineers better visualize and evaluate field performance and waterflooding efficiency.

3.3.1 2D Five Spot Case

Streamline tracing and diagnostic plots are first tested in a 2D areal five spot waterflooding case with 4 producers at the corners and 1 injector. Each producer is operating at a constant rate of 600 bbl/day, and the injection volume is the sum of all the production. This case has a heterogeneous permeability and homogeneous porosity shown in Fig. 11. The objective for this case is to test all the available flow diagnostic plots in a simple 2D case.

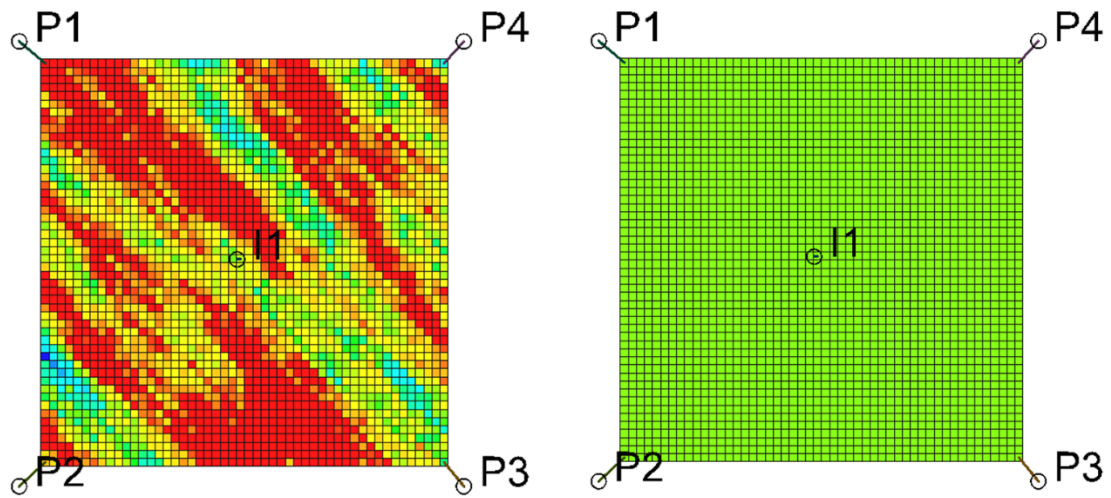


Fig. 11 - 2D 5 spot waterflooding case permeability (left) and porosity (right).

Because the case setup is very easy, with only one injector, not all the diagnostic plots are shown in Fig. 12. From the permeability field, it is observed that high permeability regions have a direction from northwest to southeast with most of the high permeability regions located at the west of the injector. From the injector TOF map, P1 and P2 have water breakthrough first. Then P3 will see fluid coming from the injector. It will take a long time for the water to reach P4. These results agree with the permeability field. Also, because there is a high permeability channel connecting P1 and I1 and each producer is producing at the same rate, most of the streamline in this well pair can go through this high permeability channel, resulting in a small volume of the reservoir being drained by P1.

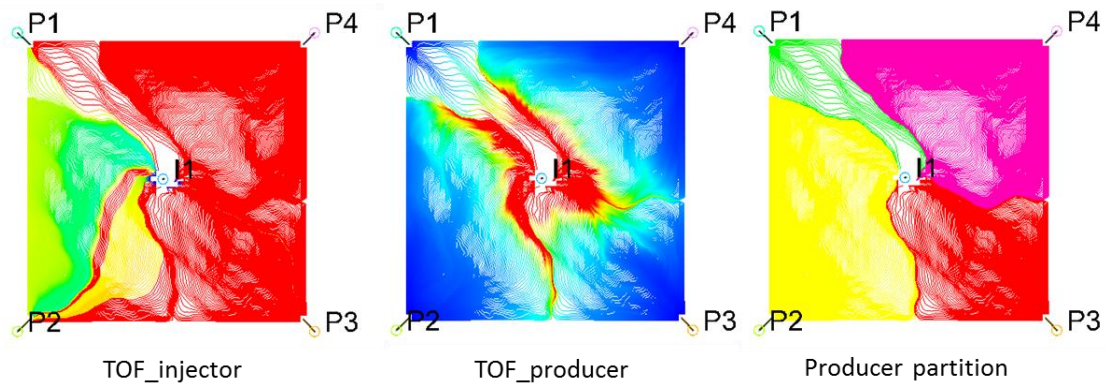


Fig. 12 - 2D 5 spot waterflooding flow diagnostic plots.

3.3.2 Brugge Benchmark Case

Flow diagnostic plots then are applied in the Brugge Benchmark case. As mentioned before, this is a 3D synthetic case. The model has a total of 30 wells: 20 producers and 10 injectors. This model has approximately 40,000 active cells and 155 NNCs. Permeability and porosity are shown in Fig. 13. As can be seen, 10 injectors are located at the high water saturation region to provide pressure support.

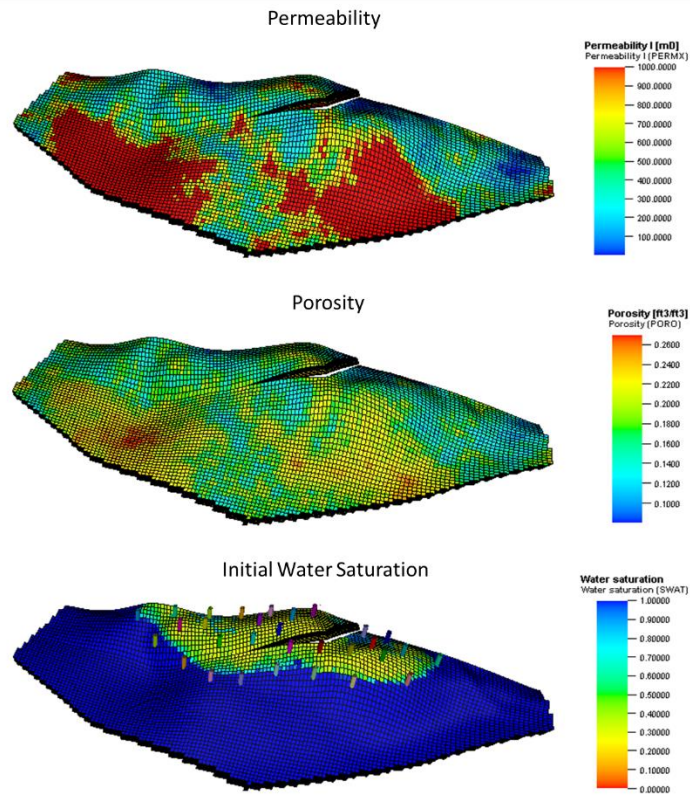


Fig. 13 - Brugge permeability, porosity, and initial water saturation field.

Using the flux information generated by the commercial simulator, we generate all 3D streamline flow diagnostic plots for Brugge shown in Fig. 14. It is noted that the results from grid view and line view are extremely similar for the Brugge case because the geometry of this model is relatively simple even though it is a 3D multi-well case. Not a lot of isolated regions exist for this field. Also, the injector and producer well locations are quite uniform with all producers at the center (low water saturation) area surrounded by injectors at the side (high water saturation) area. Even though this case shows no significant difference for line view and grid, we should not be discouraged to continue the use of grid view flow diagnostic plots when they can be generated in almost no time. When

the grid geometry gets more complicated and more non-neighbor connections appear, like most field cases do, it is possible to have different pictures for line view and grid view, and it will be demonstrated in the next application. Incorporating both views will make sure engineers get a comprehensive understanding of the reservoir and waterflooding performance.

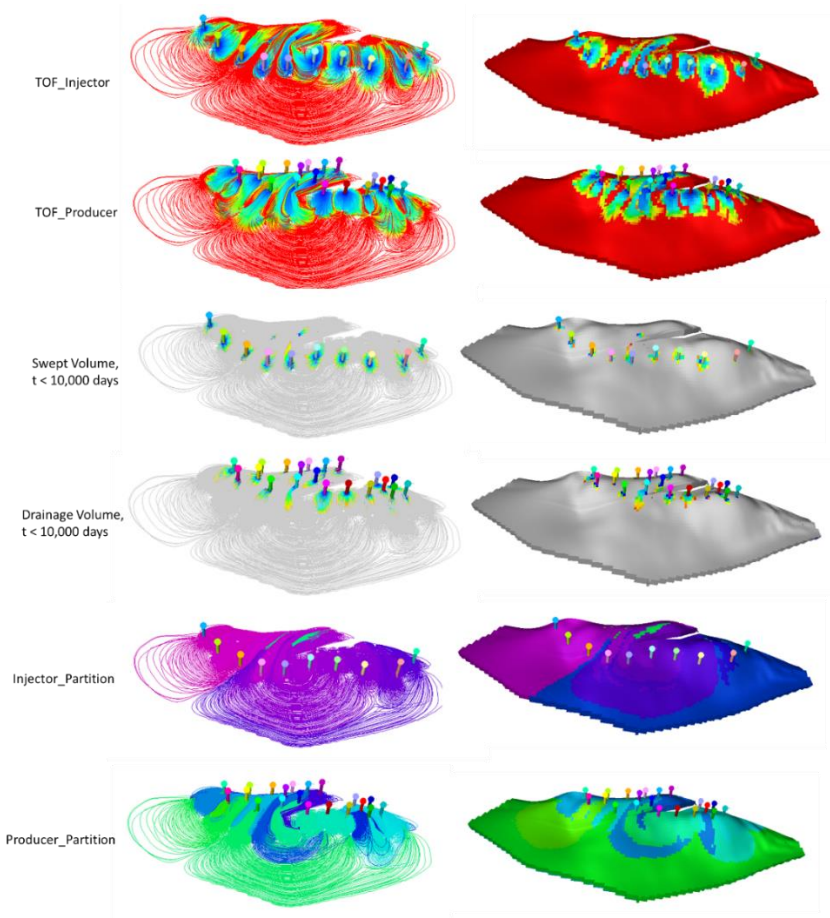


Fig. 14 - Flow diagnostic plots for the Brugge benchmark case in line view (left) and grid view (right).

3.3.3 Bijupira Field Case

The application of flow diagnostic plots is extended to the Bijupira field case. This field is located 250 km east of Rio de Janeiro, Brazil. It is a two-phase conventional reservoir, with permeability ranging from 0.1 md to 8,000 md and 30% average porosity. The production started in 1993, and it has 8 producers and 4 injectors. The permeability and porosity fields are shown in Fig. 15.

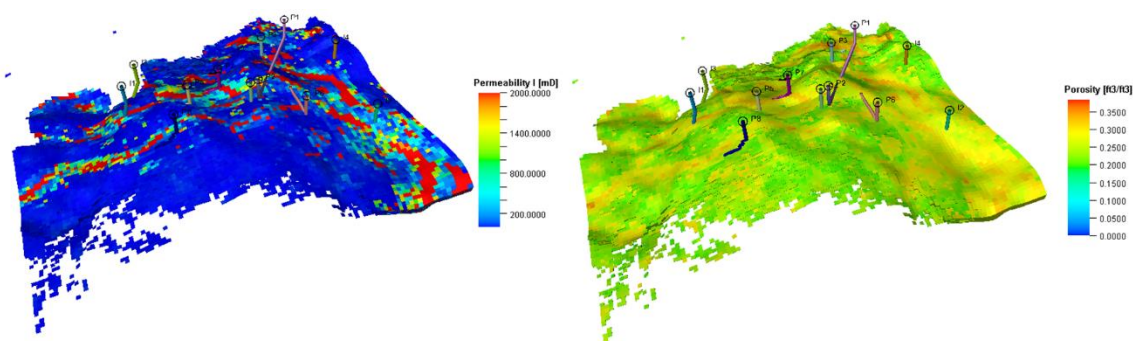


Fig. 15 - Bijupira permeability and porosity field.

As can be seen in the figure, this field is highly heterogeneous. Most of the producers are located at the high reservoir quality regions. This model has approximately 800,000 grids, and we trace a total number of 10,000 streamlines from the producers.

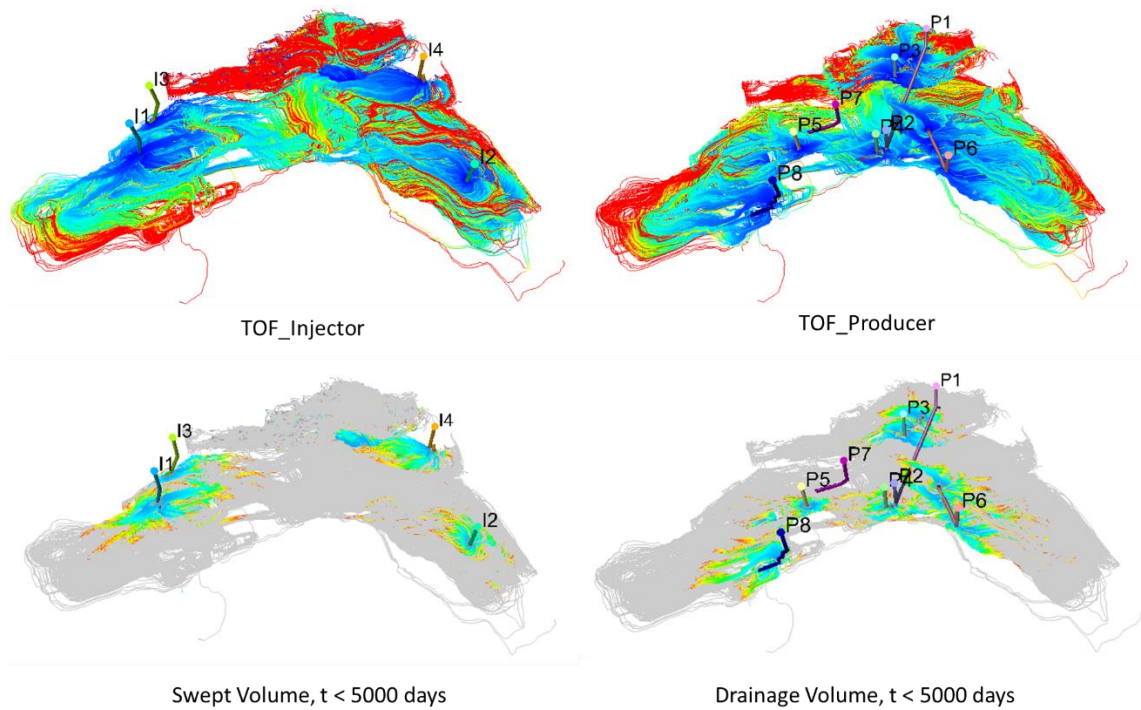


Fig. 16 - Bijupira flow diagnostic plots for reservoir characterization (line view).

The reservoir characterization plots are shown in Fig. 16. In the upper left plot, we have the TOF to injector, and in the upper right, we have the TOF to producer. Because the unit of mapping values (TOF) is in times, using a user chosen physical time $t = 5000$ days as a filter, we are able to track the fluid fronts, and it helps to identify the swept and drainage volume on the given time. Flow diagnostic plots are available in grid view shown in Fig. 17.

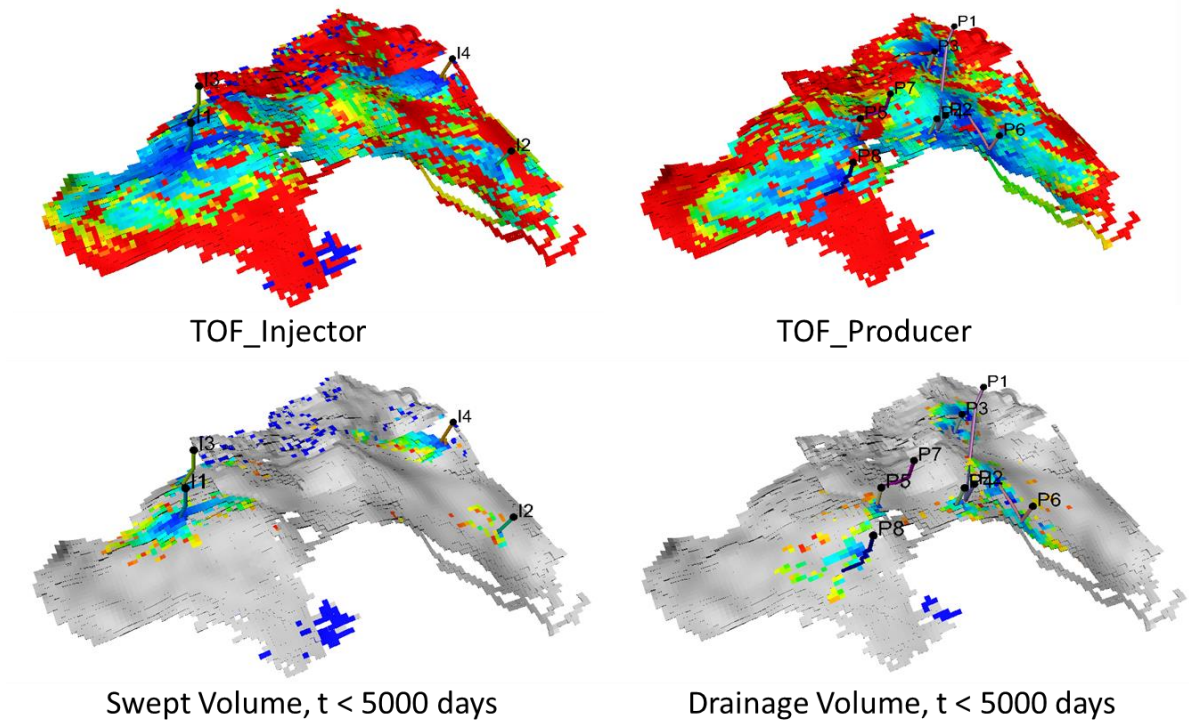


Fig. 17 - Bijupira flow diagnostic plots for reservoir characterization (grid view).

It is noted that these two views provide similar images, but there is a noticeable difference in the lower left side of the reservoir. While the line view has only one streamline passing through this region, there is a big area of reservoir showing in the grid view plot. The reason behind this is simple. Line view diagnostic plots always trace streamlines from a sink or a source. In order to show distribution of the fluxes, each streamline at the sink/source cell carries an equal amount of fluids. Thus, very few streamlines will pass through the stagnant regions. However, it does not mean that fluids at these stagnant regions do not contribute to the production. Diagnostic plots in grid view can provide a more intrusive understanding of the stagnant regions because it sees the

whole picture not just the regions where fluxes are relatively higher. Also, drainage and swept volume can be obtained in grid view.

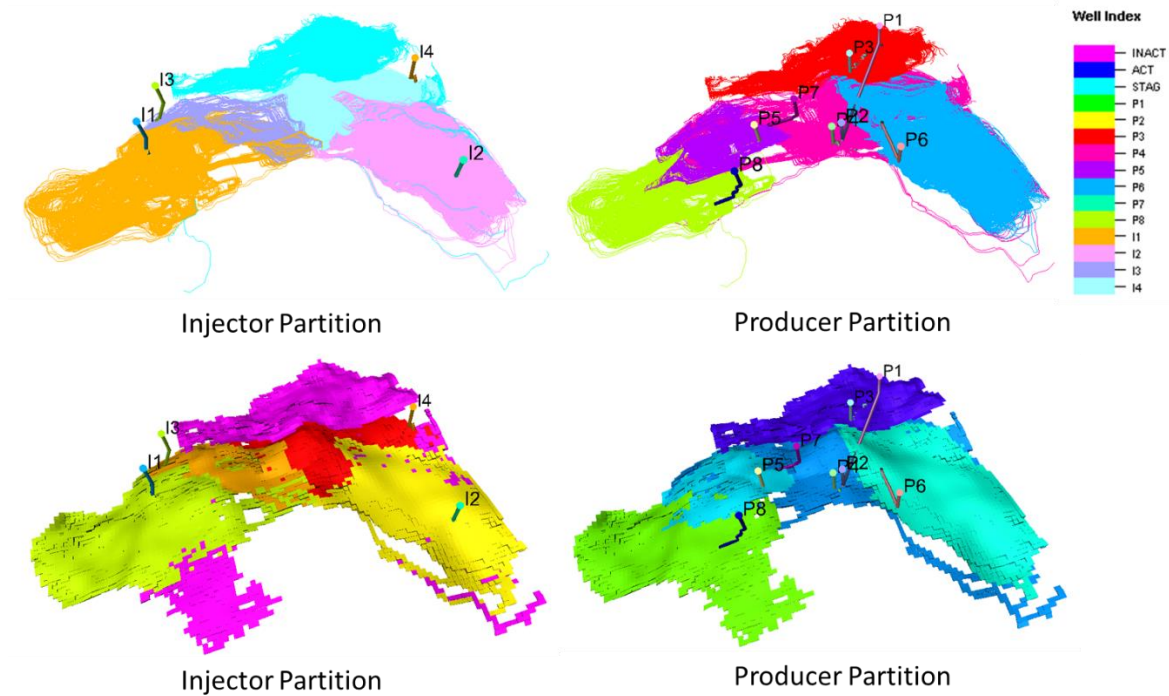


Fig. 18 - Bijupira flow diagnostic plots for reservoir surveillance showing well partition in line view (top) and grid view (bottom).

As mentioned, flow diagnostics can also be used for reservoir surveillance. Fig. 18 shows the partition of each producer and injector. Often time in a waterflooding project, engineers not only want to know where injected fluids go, but also like to know how big of an area is controlled by a specific injector or producer. This plot lets engineers visualize the information, and engineers can adjust the waterflooding strategy correspondingly. Again, if only line view diagnostic plots are available, it may lead engineers to underestimate the reservoir size, while if only grid view diagnostic plots are available,

engineers will not have an understanding of the contribution from each region. Thus, diagnostic plots in two views become complementary to each other.

These well partition plots can be further broken down into individual well partition. Combining a chosen injector partition and a producer partition, we have an injector/producer well pair, and a more detailed analysis within this well pair can be conducted. The individual well partition and injector/producer well pair can be seen in Fig. 19. Again, similar results have been achieved for both line view and grid view. The line view well pair is at the top, while the grid view is at the bottom. However, it is noticed that the grid view well pair plot is more scattered. The reason is that when doing cell center tracing, it is possible that multiple streamlines will pass through one single cell, but to keep things consistent, we only keep track of the first streamline which passes through the cell and use this streamline's sink and source as the cell's sink and source.

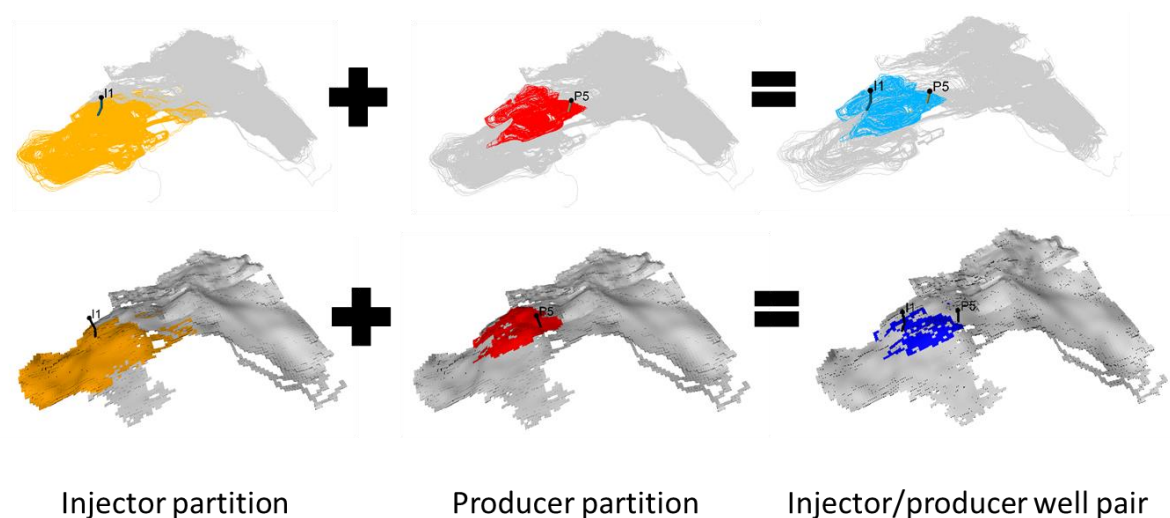


Fig. 19 - Combining an injector partition and a producer partition to obtain a well pair (line view).

Another type of flow diagnostic plot is a bubble map of injection or production distribution. Tracing either from a producer or an injector, we know how many streamlines will be connecting each well pair. Because each streamline carries an equal amount of fluids at the source/sink point, the number of streamlines directly reflects how much fluid is contributed by a specific well pair. As shown in Fig. 20, on the left-hand side, the injector allocation map shows that I2 is the biggest injector, and most of its injection is going into producer P6. On the right-hand side, the producer allocation map shows that P3 and P6 have the biggest production. While P6 production comes from injector I2 and I4, P3 is still under primary recovery. There is a lot of potential to add an injector to support production from P3 once its production declines. Compared with a 3D streamline flow diagnostic plot, the 2D well allocation map is more straightforward and easy to use because it is a summary of all the layers and it can give engineers a broad review of the reservoir performance, driving mechanism, as well as well connectivity.

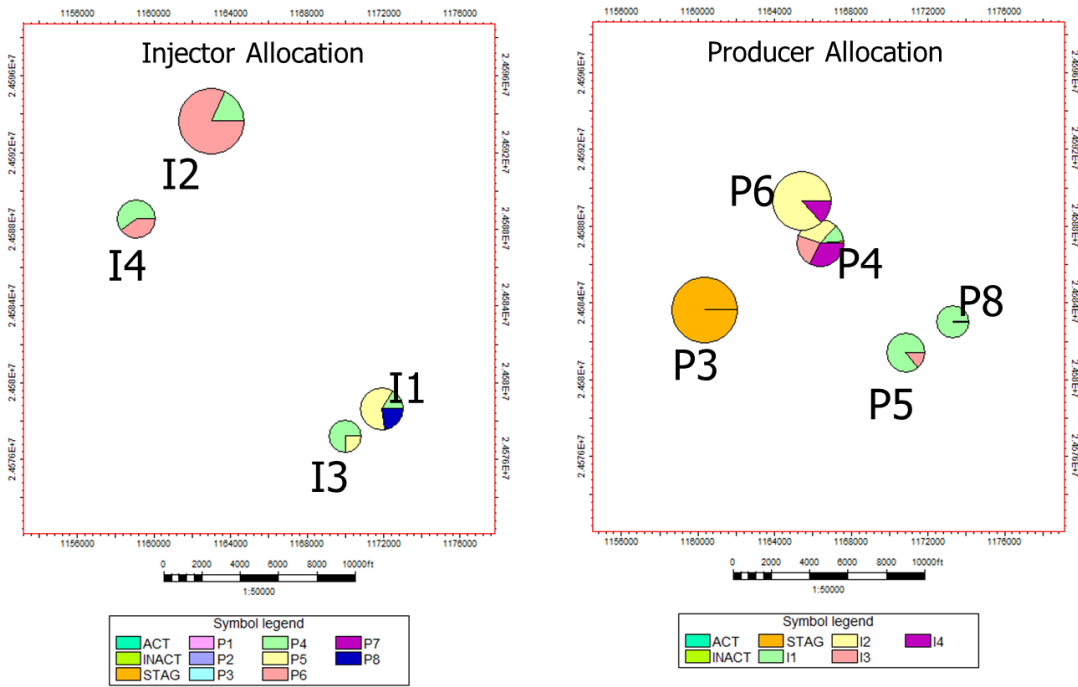


Fig. 20 - Bijupira injector/producer well allocation map.

3.3.4 Husky Field Case

The application of flow diagnostic plots is extended to the Husky field case. It has over 12 million grid cells and 1 million non-neighbor connections. There are a total of 4 producers and 4 injectors, and it has 4 year production history. The porosity of the field ranges from 0.0002 to 0.3 with an average of 0.177, and the permeability ranges from 0.0004 md to 464.8 md with an average of 162 md shown in Fig. 21.

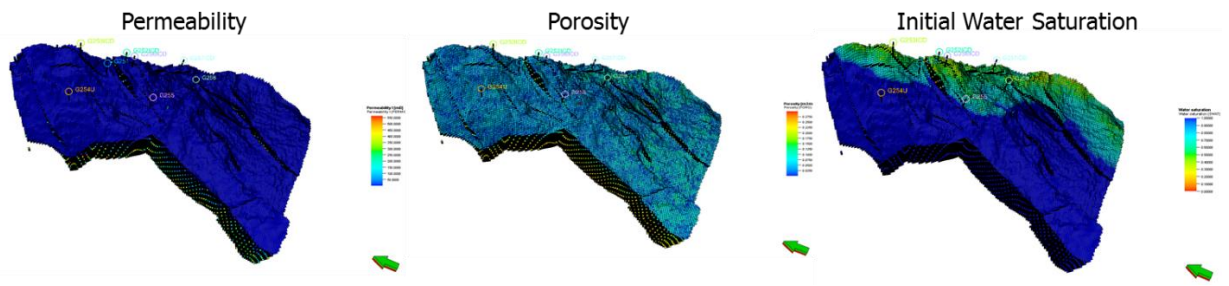


Fig. 21 – Husky permeability, porosity, and initial water saturation field.

This is a highly heterogeneous multi-million cell field case with very complex geologic and geometric settings. The simulation model has 729 faults with over 1 million NNCs, creating big challenges for streamline tracing and flow diagnostic analyses.

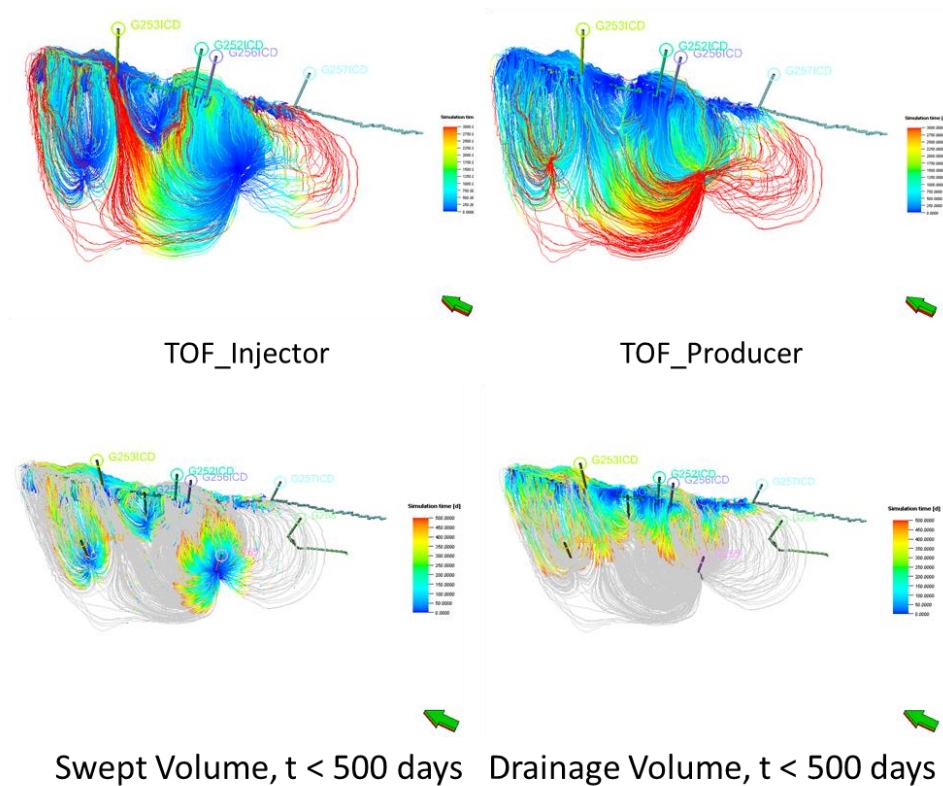


Fig. 22 - Husky flow diagnostic plots for reservoir characterization (line view).

Fig. 22 shows the reservoir characterization on line view. With the time of flight information, we are able to depict the swept and drainage volume at different times. The same analysis is also available on grid view shown in Fig. 23. It is clear that the top layer does not contribute a lot for the production because the time of flight is high. Instead, layer 110 shows a lot of fluid movement, and the swept and drainage volume at 500 days are shown.

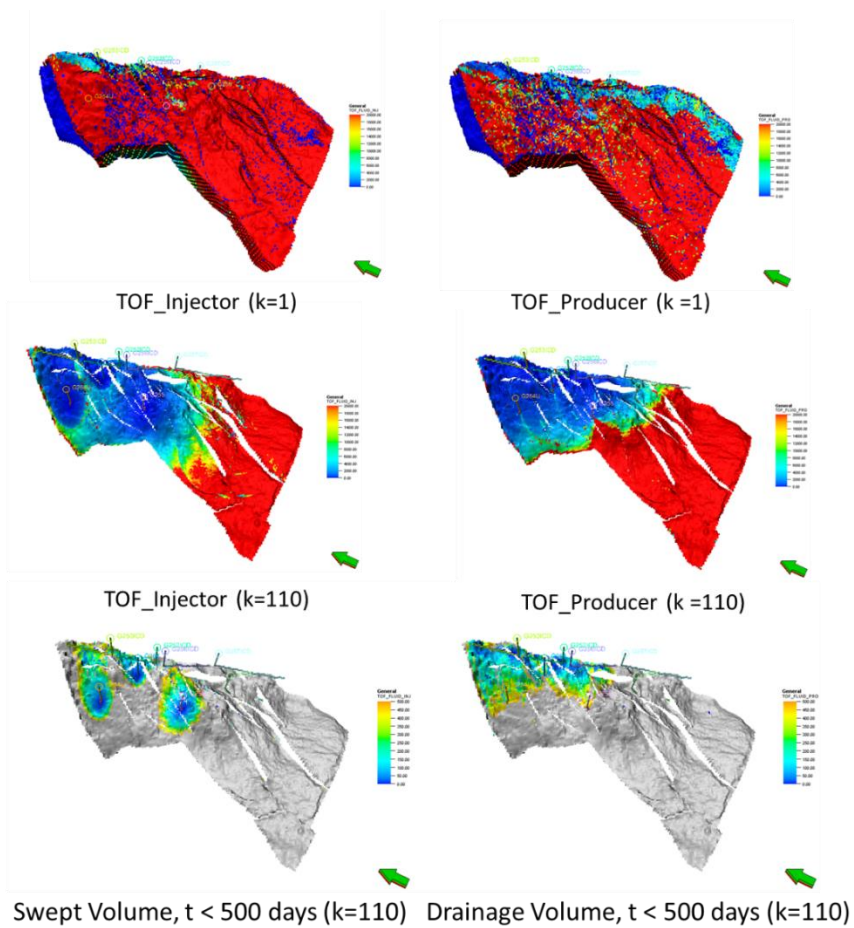


Fig. 23 - Husky flow diagnostic plots for reservoir characterization (grid view).

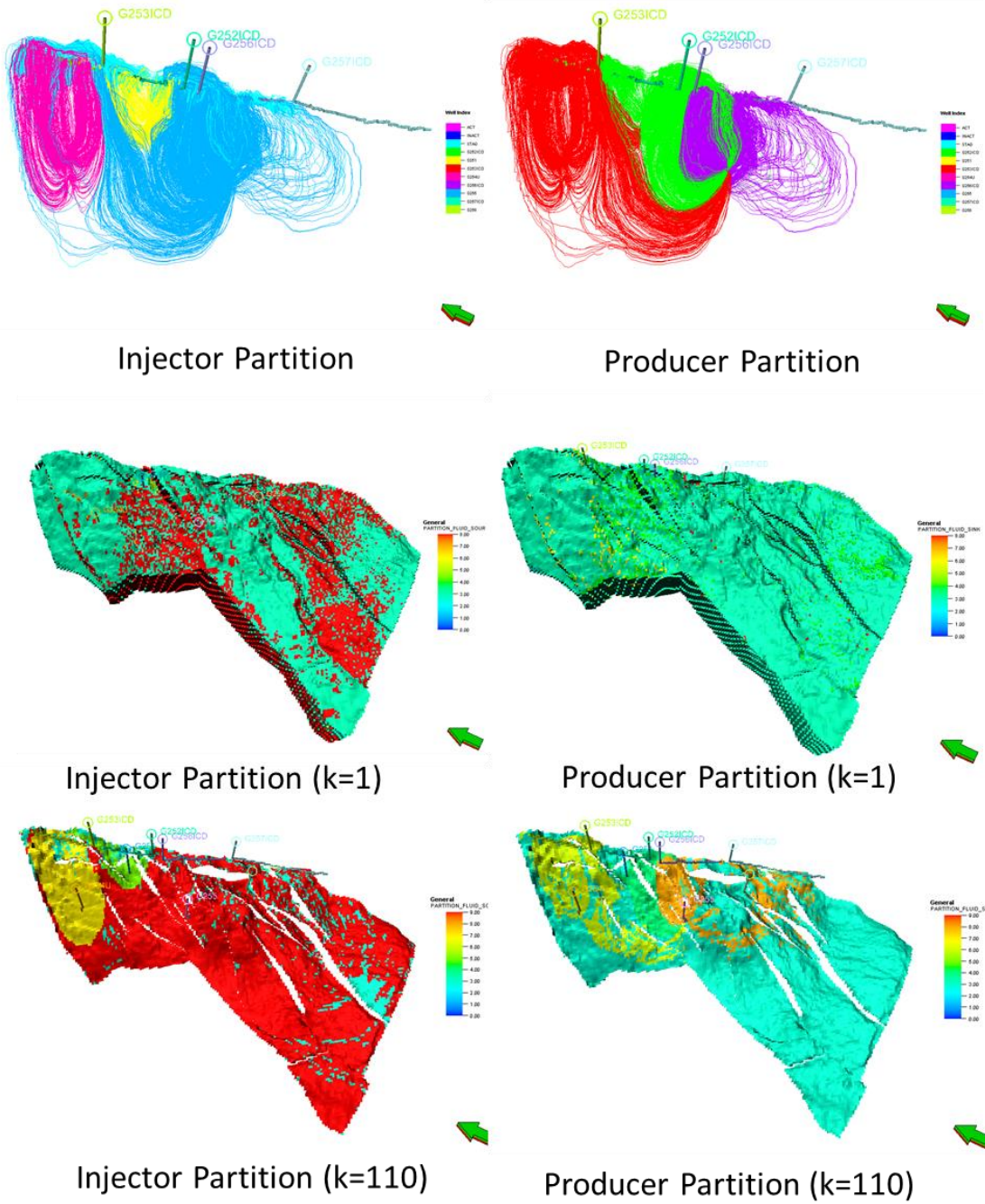


Fig. 24 - Husky flow diagnostic plots for reservoir surveillance showing well partition.

Fig. 24 shows injection and producer well partition for both line view and grid view. It is noted that if we only have the line view flow diagnostics, we miss the bottom right part of the reservoir information. We are also able to obtain well allocation information as shown in Fig. 25. It indicates at this given time step, G253ICD is the biggest producer, and most of its production comes from primary recovery. G255 is the biggest injector, and most of the injected water goes toward G256ICD.

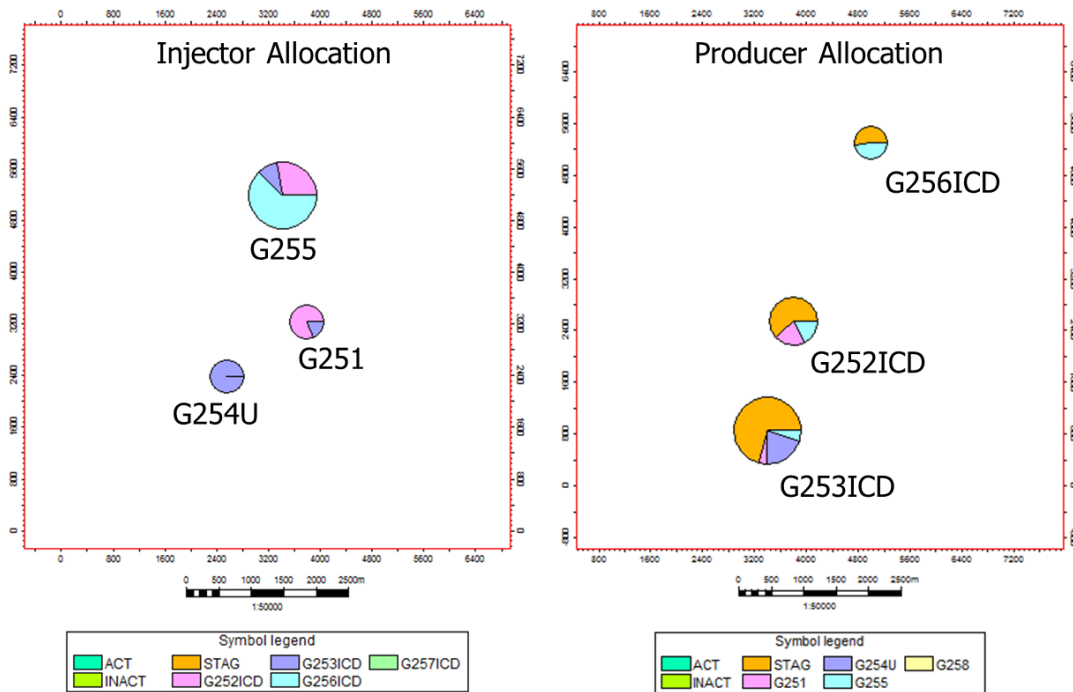


Fig. 25 - Husky injector/producer well allocation map.

CHAPTER IV

STREAMLINE-BASED WATERFLOODING OPTIMIZATION

4.1 Background and Theory

Streamline-based methods have shown their utility for waterflooding management due to streamline's ability to capture fluid front and its visualization, and several streamline-based waterflooding optimization methods have been developed. In this research, several streamline-based optimization techniques are implemented into Destiny 6 to test and compare their performances for waterflooding management problems.

One of the streamline optimization techniques is based on well allocation. It evaluates and ranks injector and producer rate performance by streamline time of flight and trajectories and adjusts rates correspondingly, and its goal is to maximize oil recovery. Another approach is based on fluid arrival time. Because TOF is a representation of reservoir heterogeneity, sweep efficiency will increase by equalizing TOF for each well pair. A final approach is based on the Net Present Value of a well pair. It still evaluates and ranks injector and producer performance by streamline time of flight and trajectories, but it further incorporates information to obtain the total monetary value of a well pair and its net present value.

Waterflooding optimization is a highly non-linear optimization problem with multiple well and field constraints. Similar to approaches to solve a history matching problem, the techniques to solve waterflooding optimization can be divided into 2 categories: gradient-based approaches (Suwartadi 2012) and non-gradient based

approaches (Spall 2005). Gradient-based methods calculate parameter sensitivity and iteratively approach an optimal solution, but often time, these approaches will be trapped at a local optimal solution. Non-gradient-based methods, or stochastic methods, are able to find the global optimum point by searching all solution space (Spall 2005). However, it is hard to implement in waterflooding optimization as the number of simulation increases exponentially when the number of control variables increases (Harding et al.1996).

The use of streamline and time of flight for waterflooding management is proved to be effective (Thiele and Batycky 2003, Alhuthali et al. 2007, Moyner et al. 2014) because streamline supplies information which finite difference simulators cannot provide. Again, it captures convective flow, provides time of flight for the flow and establishes well pair and well connection relationship as shown in Fig. 14. Several literatures can be found for streamline-based waterflooding optimizations. In this research, we focus on two approaches. One is to allocate the optimal injection rate using producer/injector well allocation factor, developed by Thiele and Batycky (2003). The second one is the use of equalizing arrival time (Alhuthali et al. 2007, Park and Datta-Gupta 2011), and the third one is streamline-based NPV method, developed by Tanaka (2014). The fundamentals and workflows of different streamline-based optimizations are similar. We first trace streamline and calculate time of flight. Once this process is finished, integrated with reservoir dynamic properties such as phase saturation and phase mobility, we can achieve detailed quantitative analyses such as injector efficiency, sensitivity of rate on TOF, and injector NPV efficiency if we further include financial information into

calculations. All algorithms are demonstrated in detail and their advantages and limitations are discussed.

4.1.1 Streamline-based Well Allocation Factor Optimization

Thiele and Batycky (2003) proposed a waterflooding optimization method using well allocation factors. Like every other waterflooding optimization objective, the objective of this method is to maximize oil production and recovery for every barrel of water injected while honoring production/injection constraints. A key concept in this optimization is injector efficiency, which is defined as follows in the original paper

$$I_{ip} = \frac{\text{offset oil production}}{\text{water injection rate}} \dots\dots\dots(4.1)$$

The injector efficiency is the oil cut for a specific well pair, and the equation can be rewritten as

$$e_{ip} = \frac{\text{offset oil production rate of producer } i}{\text{water injection rate of injector } i \text{ to producer } p} = \frac{\sum_l^{N_{sl}} q_{ip,l} F_{o,l}^{end}}{\sum_l^{N_{sl}} q_{ip,l}} \dots\dots\dots(4.2)$$

where subscript ip stands for a specific injector/producer well pair, N_{sl} stands for the total number of streamlines connecting this well pair, and F_o^{end} is the oil fractional flow at the producer for each streamline. $q_{ip} * F_o^{end}$ represents the oil production for each streamline within the well pair. Thus, by calculating e_{ip} for each injector/ producer well pair within the system, we have overall field flood efficiency plots. Ideally, for piston-like water injection cases, $F_o^{end} = 1$, and well pairs reach their highest injector efficiency.

Once the streamline tracing is completed and injector efficiency is calculated using a single step flux information, we then can update the water injection rate of an individual injector as

$$q_{ip}^{n+1} = (1 + w_{ip}) q_{ip}^n \times r \dots\dots\dots(4.3)$$

where w is the weight factor calculated from field efficiency and r is the factor to keep the constraints of total available water for injection. Assuming that the initial total injection rate is the total available water in a field, r is calculated as follows

$$r = \frac{\sum_i^{Ninj} q_i^n}{\sum_{ip}^{Nip} (1 + w_{ip}) q_{ip}^n} \dots\dots\dots(4.4)$$

and the weight factor w is calculated as (Thiele and Batycky 2003)

$$w_i = \begin{cases} \min \left[w_{\max}, w_{\max} \left(\frac{e_i - \bar{e}}{\bar{e} - e_{\min}} \right)^\alpha \right], & \text{if } e_i \geq \bar{e} \\ \min \left[w_{\min}, w_{\min} \left(\frac{\bar{e} - e_i}{\bar{e} - e_{\min}} \right)^\alpha \right], & \text{if } e_i \leq \bar{e} \end{cases} \dots\dots\dots(4.5)$$

where the exponent α and constraint values $w_{\max, \min}$ control the degree of rate adjustment allowed for a single run, and $e_{\max, \min}$ is the maximum/minimum efficiency for the field well pairs. The sensitivity of α on weight factor is shown in Fig. 26. $\alpha = 2.0$ will result in

the most smooth change for well pairs close to average field efficiency, and it will be used in applications.

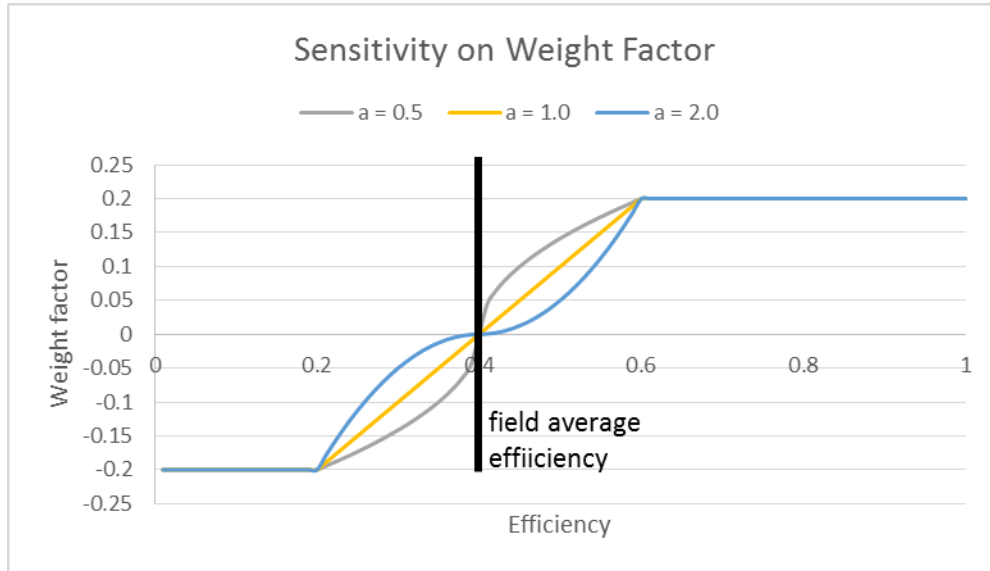


Fig. 26 - Sensitivity of α on weight factor.

The average efficiency \bar{e} is simply given by the field total production divided by the total water injection, calculated as

$$\bar{e} = \frac{q_{field,prd}^{oil}}{q_{field,inj}^{wat}} \dots\dots\dots(4.6)$$

After computing efficiency of each injector/producer well pair and field average efficiency, Eq. 4.3 is used to update water injection rate. This is a heuristic approach. Its underlining assumption is that field production will increase by equalizing injector

efficiency or oil cut for all injector/producer well pairs. In other words, this approach reduces the injection rate if the efficiency is lower than the field average, and vice versa.

The injection rate for the next iteration is calculated by summing up all the injector/producer well pair rates using

$$q_i^{n+1} = \sum_{ip}^{N_{ip,i}} q_{ip}^{n+1} \dots\dots\dots(4.7)$$

Rate allocation optimization uses a single time step result to perform optimization using Eq. 4.2-4.7, and multiple iterations can be conducted to equalize efficiency for each well pair. It is expected to generate higher oil production from iteration to iteration. The limitation of this approach is the dependence on clear water breakthrough information. Before water breakthrough happens, the efficiency is not very sensitive to the injection rate. Therefore, the proposed injection rate may not be reliable.

4.1.2 Streamline-based Equalizing Arrival Time Optimization

The objective of this approach is to maximize the waterflooding sweep efficiency. To achieve the goal, it reallocates individual well rate to equalize arrival time of water front to each producer. The equation for updating the rates is written as

$$S\delta\mathbf{q} = \mathbf{e} \dots\dots\dots(4.8)$$

where \mathbf{e} is a residual vector between individual well pair breakthrough time and field average breakthrough time. The matrix S is the partial derivative of the residual vector with respect to individual well rate. This matrix dimension is $N_{prd} \times N_{inj}$, representing all possible well pairs in a field. The sensitivity can be derived as follows under the

assumption that a small change in a well rate does not change the streamline shape and arrival time

$$\begin{aligned}
 \mathbf{S}_{ip} &= -\frac{\partial t_{ip}}{\partial q_i} = -\frac{\partial \tau_{ip}}{\partial q_i} / [df_w / dS_w] = -\frac{\partial \tau_{ip}}{\partial q_{sl}} \frac{\partial q_{sl}}{\partial q_i} / [df_w / dS_w] \\
 &= -\frac{\sum_{sl}^{N_{sl,ip}} \tau_{sl} / [df_w / dS_w]}{q_{sl} N_{sl,i}} \dots\dots\dots(4.9)
 \end{aligned}$$

where t_{ip} is the actual average break through time and τ_{ip} is the average time of flight of a

well pair. $\frac{\partial \tau_{sl}}{\partial q_{sl}}$ is approximated by $-\frac{\tau_{sl}}{q_{sl}}$ based on Darcy's law (Alhuthali, Oyerinde, and

Datta-Gupta 2007). $\frac{\partial q_{sl}}{\partial q_i}$ is approximated by one over the number of streamlines launched from injector i .

The change of well rate can be calculated solving the following matrix equation

$$\begin{bmatrix} S \\ \mathbf{I} \end{bmatrix} \delta \mathbf{q} = \begin{bmatrix} e \\ 0 \end{bmatrix} \dots\dots\dots(4.10)$$

where identity matrix \mathbf{I} is introduced to set total change of well rate to be zero.

This method heavily depends on the frontal propagation of injection fluid. Thus, its best performance is expected to be at the early time of an injection project, when the reservoir is saturated with oil and water breakthrough has not yet occurred.

4.1.3 Streamline-based NPV Optimization

Streamline-based Net Present Value (NPV) optimization was proposed by Tanaka (2014) to optimize field NPV by allocating injection and production well rate. It preserves

the advantages of streamline-based approaches and it is able to post processing simulation results from commercial simulators. More importantly, it makes good use of an important streamline simulation concept: time of flight (TOF). Because TOF stores important information about how much time it will take for a neutral tracer to travel to a location, using it with fluid phase velocity, we will be able to identify how much time it will take for a specific phase fluid to be recovered. This is exactly the information that is required for NPV calculations. In addition, it does not require clear water breakthrough information. Thus, it can be applied at any time during a waterflooding project.

We first need to determine the pore volume of a streamline. As discussed in Chapter II Eq. 2.7, a streamline is not only a line but it has an associated pore volume. We can extend Eq. 2.7 to compressible flow system, and the pore volume of a streamline can be calculated as follows

$$PV_{sl} = \int_0^s A(\xi)\phi(\xi)d\xi = \int_0^s \frac{q_{sl}\phi(\xi)}{\rho_{eff}(\xi)u_t(\xi)}d\xi = \sum_i^{node} \frac{q_{sl}}{\rho_{eff,i}} \Delta\tau_i \dots\dots\dots(4.11)$$

where ρ_{eff} is the effective density, and it varies along a streamline in compressible system. After obtaining the pore volume along a streamline, we are able to calculate the total

hydrocarbon monetary value along a streamline. The total hydrocarbon monetary value is calculated as

$$\Lambda_{sl} = q_{sl} \sum_i^{node} \frac{\Delta\tau_i}{\rho_{eff}} \sum_{\alpha=og} (S_{\alpha} b_{\alpha} R_{\alpha}) \dots\dots\dots(4.12)$$

where S_{α} is the phase saturation, b_{α} is the inverse of the phase formation volume factor, and R_{α} is the phase fluid price per unit volume. In other words, this equation is multiplying the hydrocarbon volume with its unit price.

The discounted hydrocarbon monetary value along a streamline is calculated as

$$r_{sl} = q_{sl} \sum_i^{node} \frac{\Delta\tau_i}{\rho_{eff}} \sum_{\alpha=og} (S_{\alpha} b_{\alpha} R_{\alpha}) \cdot (1+d)^{-\tau_{node} \cdot f_{\alpha}^{-1} / 365} \text{ } \notin \tau_{node} > t_{rsm} \dots\dots\dots(4.13)$$

where t_{rsm} is the reservoir remaining life or contract life, d is the discount rate, and f_{α}' is the derivative of the phase fractional flow, also defined as the phase velocity. Time of flight is the spatial coordinate in streamline system. The term $\tau_{node} \cdot f_{\alpha}^{-1}$ is the physical time it will take for α phase fluid to travel to a node. Thus, $(1+d)^{-\tau_{node} \cdot f_{\alpha}^{-1} / 365}$ provides the discount factor for α phase fluid to reach a producer. With the total and discounted hydrocarbon monetary value for each individual streamline, summing all streamlines within a well pair, the efficiency for a well pair can be calculated as

$$e_{ip,NPV} = \frac{\sum_l^{Nsl} r_{sl}}{\sum_l^{Nsl} \Lambda_{sl}} \dots\dots\dots(4.14)$$

In order to better visualize the efficiency, the total and discounted hydrocarbon monetary values are normalized as $\tilde{\Lambda} = \Lambda_{ip} \cdot 100 / \max(\Lambda)$ and $\tilde{r}_{ip} = r_{ip} \cdot 100 / \max(\Lambda)$

respectively. Following very similar methods as described in Eq. 4.3 – 4.5, we can update the production and injection rate.

In order to create a more general approach for waterflooding optimizations, we create a concept of an integrated efficiency, which is defined as

$$e_{ip} = \left(\frac{\tilde{r}_{ip}}{\tilde{\lambda}_{ip}} \right)^\alpha \cdot \left(\frac{\tau_{ip}}{\tau_{fm}} \right)^\beta \cdot \left(\frac{Q_{o,ip}}{Q_{t,ip}} \right)^\gamma \dots\dots\dots(4.15)$$

where α , β , and γ stands for the weight to the each objective, τ_{ip} is the average arrival time for a well pair, and τ_{fm} is the field average well pair arrival time. The first objective is maximizing NPV, the second one is equalizing the saturation front, and the third one is maximizing the production rate. By assigning different weights on different cases, one can achieve different objectives as needed. For example, to optimize production rate can be done by $(\alpha, \beta, \gamma) = (0, 0, 1)$, and to optimize NPV can be done by $(\alpha, \beta, \gamma) = (1, 0, 0)$. It is important to point out that weight factor for equalizing arrival time, β , can often be used to reduce the change of well rate. For instance, $(\alpha, \beta, \gamma) = (1, 0.25, 0)$ will result in smaller change in well rate compared to $(\alpha, \beta, \gamma) = (1, 0, 0)$.

4.2 Application

In this research, several synthetic cases including the Brugge benchmark case are used to test and compare the performance of different waterflooding optimization algorithms. It is important to note that Destiny 6 will be used for the tracing engine, and Eclipse will be used as the pressure and flux solver. Therefore, non-neighbor connections

will no longer be a limitation for this waterflooding optimization application. The general parameters for waterflooding optimization are shown in Table 4.

Table 4 - Parameters used for waterflooding optimization.

Parameter Name	Value
Relative oil price	1.0 [\$/bbl]
Relative water price (Produced)	-0.2 [\$/bbl]
Relative water price (Injected)	-0.2 [\$/bbl]
Relative gas price	0.0 [\$/bbl]
$e_{min}, e_{max}, W_{min}, W_{max}, \alpha$ (For WAFs)	0.0, 1.0, -, 0.1, 0.1, 2.0
Amount of SL use (For EqAT)	80%
α, β, γ (For SLNPV)	1.0, 0.2, 0.0

4.2.1 2D Five Spot Synthetic Case

Waterflooding optimization is tested in a 2D areal five spot case. This is a 2 phase oil-water case. It is designed to be slightly compressible and there is no capillarity between phases. The total injection rate is 100 bbl/day. As shown in Fig. 27, the initial injection rate for each injector is 25 bbl/day. This configuration also serves as the base case in waterflooding optimization application. The objective of this problem is to reallocate the injection rate to improve reservoir recovery and NPV.

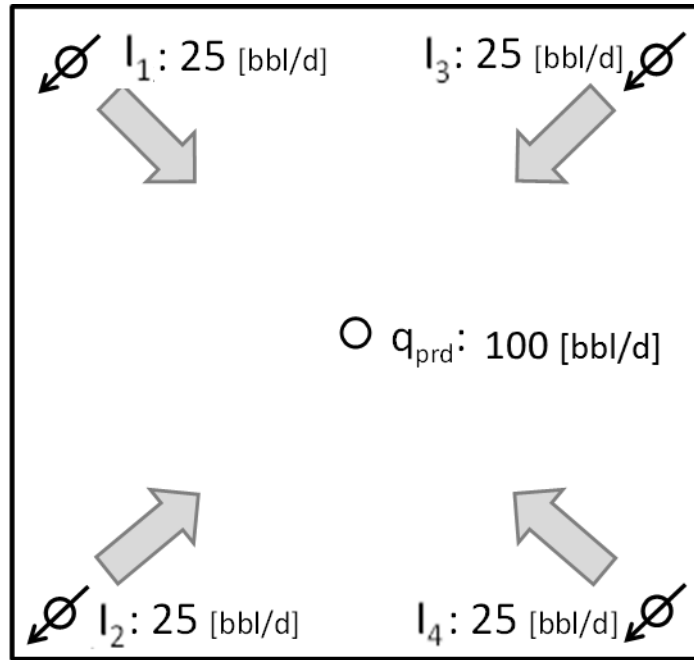


Fig. 27 - The schematic of five spot waterflooding configuration (base case).

Permeability, porosity, as well as initial water saturation distribution are shown in Fig. 28. The model is highly heterogeneous in permeability and porosity with a high permeability and porosity channel running from injector I1 to injector I4. The saturation map shows high oil saturation at the left side of the producer. Thus, in the early time, a heuristically preferable strategy is to increase the injection rate for I1 and I2 because they are located at the high oil saturation region. Furthermore, I1 should have the highest injection because it is located at the high permeability and porosity region.

In this research, 4 different strategies are tested for waterflooding optimization: base case (uniform injection), well allocation factor (WAF), equalizing arrival time (EqAT), and streamline NPV (SLNPV).

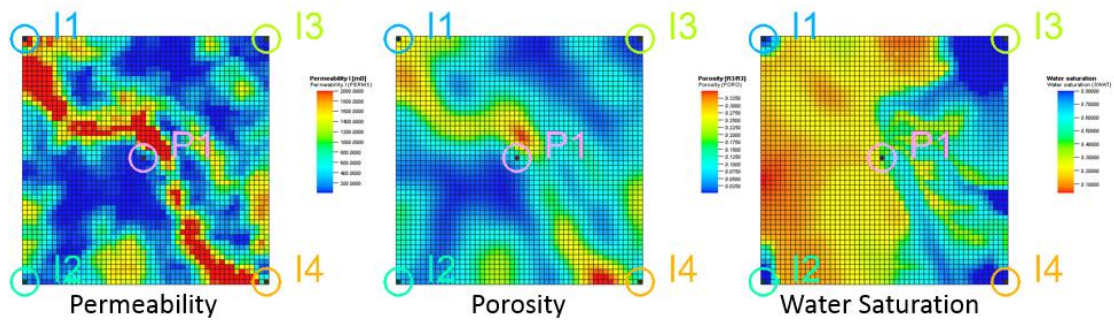


Fig. 28 - Permeability (left), porosity (middle), and water saturation (right) distribution.

Fig. 29 shows the injection rate history for 4 different strategies. Injection rate is updated at each time step, and all of these optimizations are fully automatic. Again, the field water injection rate constraint is 100 bbl/day. The base case shows uniform injection rate of 25 bbl/day per well throughout the history. Fig. 29 (b) shows the injection strategy from WAF. The initial change in injection rate is very smooth due to the fact that there is no clear difference in initial saturation around producer for each well pair. Because of the high permeability and porosity near I1, I1 has a higher injector efficiency so WAF increases I1 injection rate. Fig. 29 (c) shows the result for SLNPV. I1 and I2 have high injection rate in the early time because they located at the high oil saturation region, while I1 has the highest injection because of the high permeability and porosity. Fig. 29 (d) shows the result for EqAT. The injection rates remain constant after the early change because of the small compressibility. The steady state condition maintains the streamline distribution, so the injection does not vary much after the change in the early stage. It also shows that both I1 and I4 have the high injection rate because they are at the high permeability as well as high porosity regions.

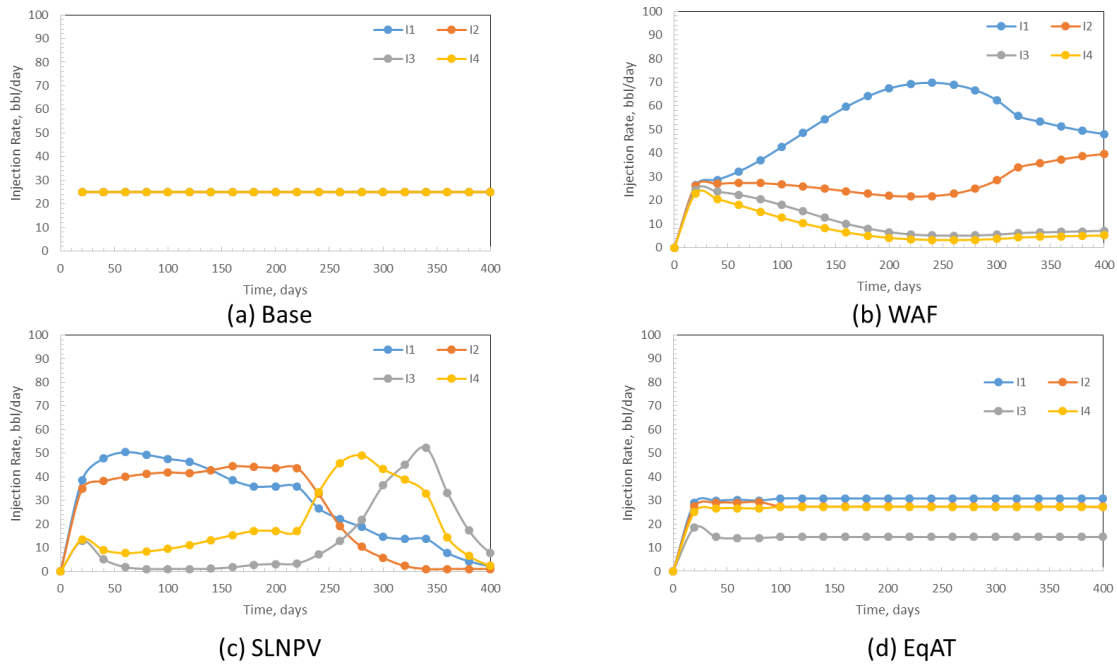


Fig. 29 - The history of injection rate of 4 strategies: (a) base (b) WAF (c) SLNPV (d) EqAT.

Fig. 30 shows the comparisons of NPV and recovery factor. It is clear that SLNPV has the highest NPV throughout the simulation period. Also, at the later stage after around 350 days of production, SLNPV reduces the total injection in order to maintain an increasing NPV. However, EqAT focuses on sweep efficiency and WAF focuses on production rate optimization, and they do not see NPV decreases so they do not adjust the total injection rate. On the other hand, SLNPV results in a relatively lower recovery compared to WAF and EqAT. EqAT has the best recovery factor performance at the end of simulation run.

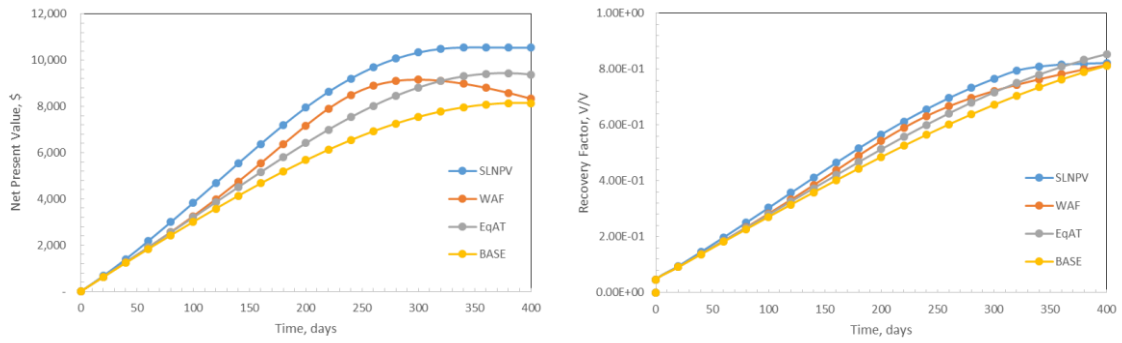


Fig. 30 - The comparisons of net present value (left) and recovery factor (right) for 4 different strategies (2D 5 Spot).

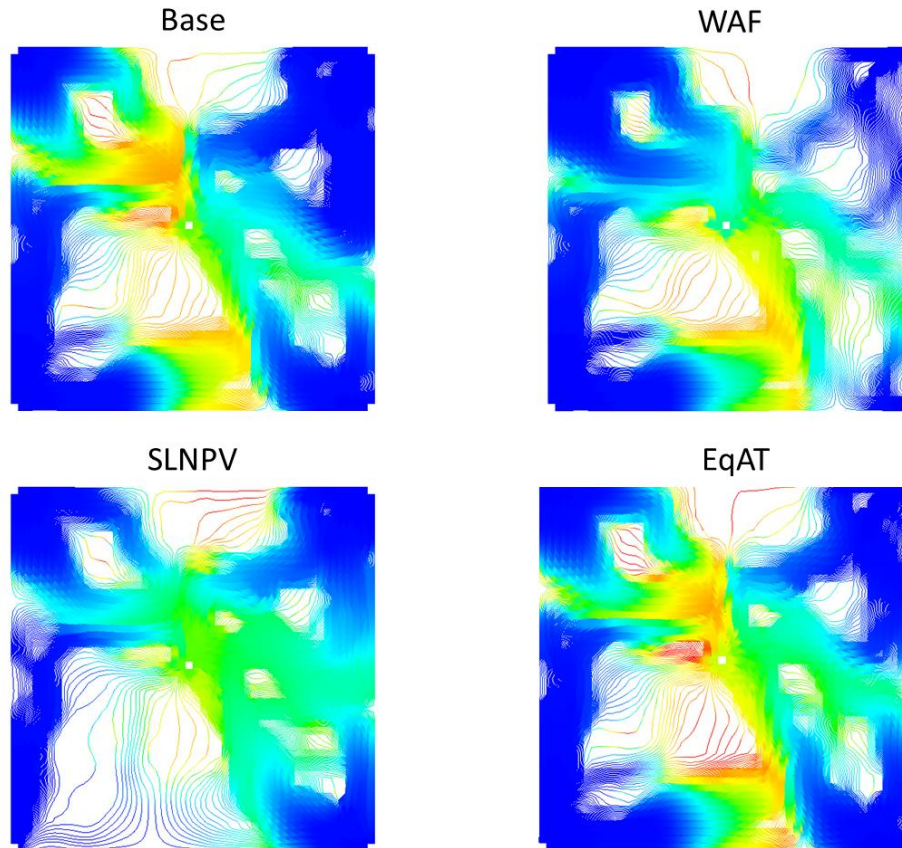


Fig. 31 - Streamline and water saturation distribution for 4 different scenarios at 300 days (5SP).

Fig. 31 shows the streamline and water saturation distribution at 300 days. SLNPV shows the lowest oil saturation among different scenarios, and it is reallocating more rate and streamlines to the right side of the producer, in I3 and I4.

4.2.2 2D Areal Multi-well Case

Waterflooding optimization is also tested in a multi-well 2D case. This case has 3 mobile phases. The total available water is 1,010 rb/day, and maximum production capacity is 1,000 rb/day. There are a total of 8 injectors and 7 producers placed by a 5 spot pattern as shown in Fig. 32. The objective is to maximize NPV or recovery factor by reallocating injection and production for each well under the constraints of total available injection rate and production capacity.

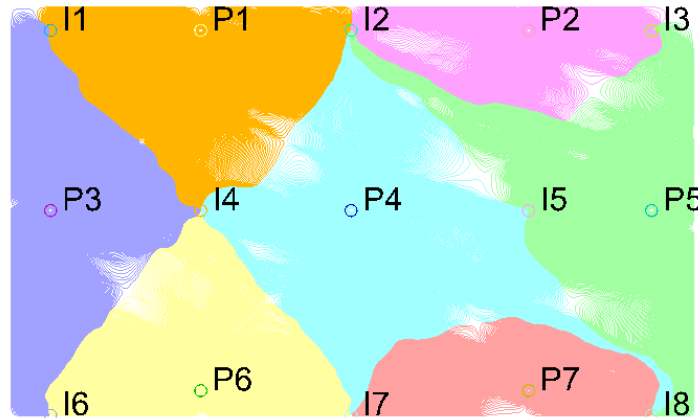


Fig. 32 – Multi-well case well configuration and streamline producer partition.

Permeability, porosity, and initial water saturation fields are shown in Fig. 33. This is a highly heterogeneous reservoir, and the permeability follows the same trend as

the porosity. The bottom part of the reservoir has been swept, resulting in a higher initial water saturation.

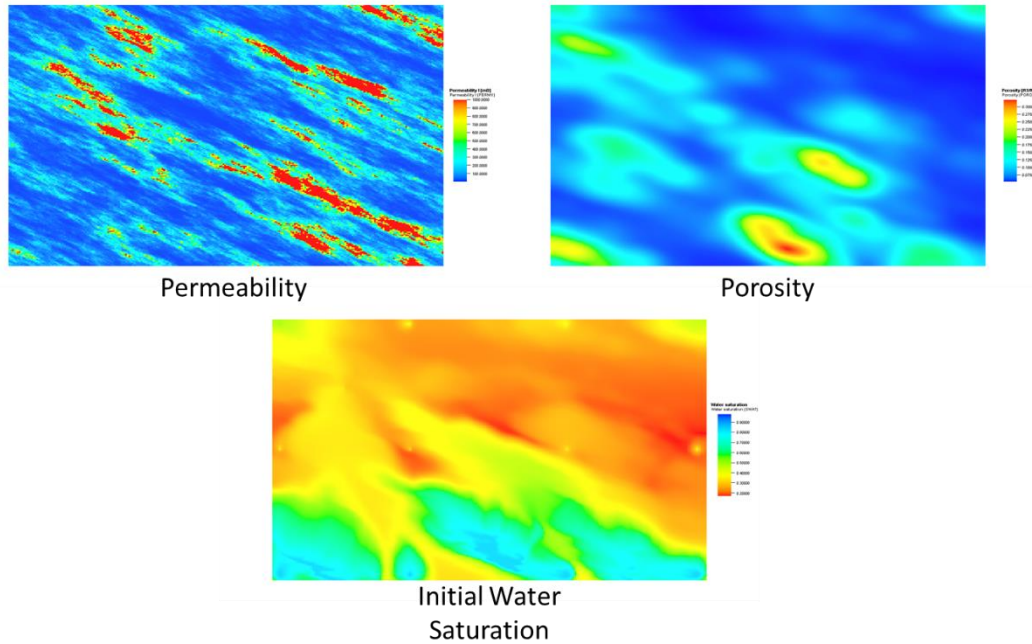


Fig. 33 – Multi-well case permeability, porosity, and initial water saturation field.

The relative price of oil, gas, and water is 1.0, 0.0, and -0.2 respectively. There is a tradeoff between creating a large dropdown to support higher total rate and maintaining high pressure to stop gas coming out of the solution since the appearance of free gas will reduce NPV.

Again, four different optimization algorithms are tested: base case (uniform injection), well allocation factor (WAF), equalizing arrival time (EqAT), and streamline NPV (SLNPV). Fig. 34 shows the NPV efficiency plot for base case and SLNPV case for the first step of the simulation.

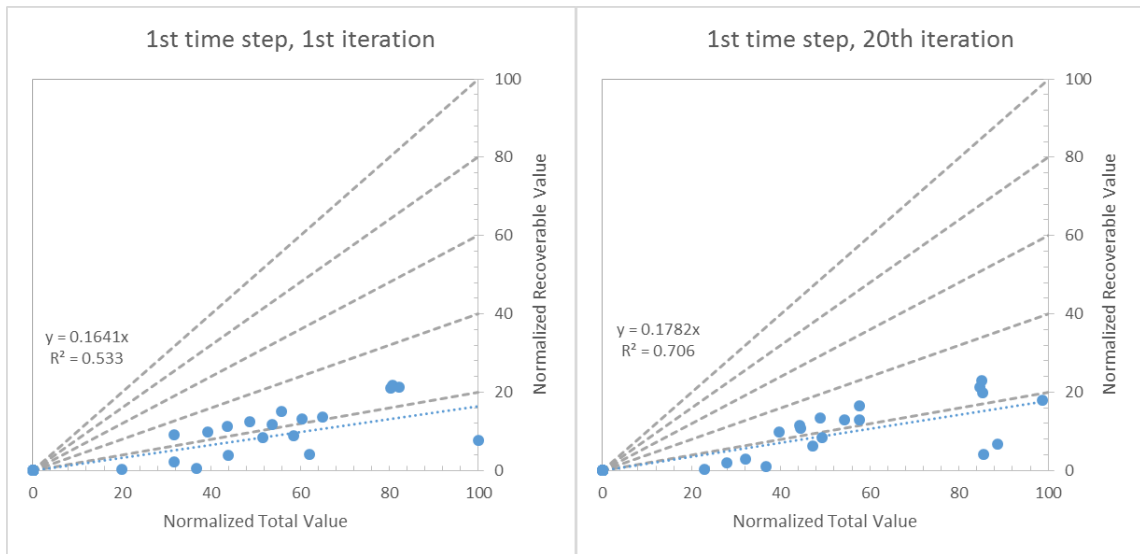


Fig. 34 – Multi-well case NPV efficiency plot of base case (left) and SLNPV case (right).

As shown in the trend lines, the average NPV efficiency improves from 16.41% to 17.82%. Meanwhile, the standard deviation of NPV efficiency per well pair reduces. By equalizing the efficiency for each well pair, we obtain the results of NPV and recovery factor in Fig. 35. All optimization methods show improvement compared to the base case. SLNPV shows the best NPV as well as recovery performance before 900 days. After 900 days, EqAT and WAF pass SLNPV on recovery performance due to the fact that SLNPV reduces the total injection and production.

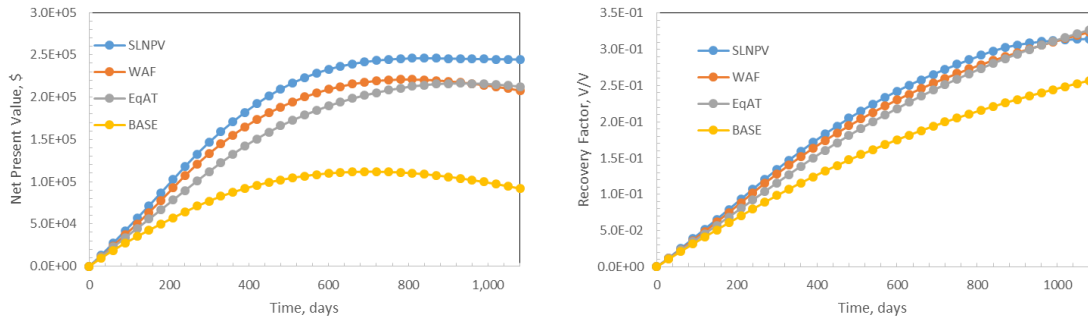


Fig. 35 - The comparisons of net present value (left) and recovery factor (right) for 4 different strategies (2D Multi-well).

4.2.3 Brugge Benchmark Case

In this section, we extend the application to the Brugge benchmark case. The details of the case have been explained in Chapter III. It has 20 vertical producers and 10 peripheral water injectors. It is important to note in the previous run by Tanaka (2014), the non-neighbor connections for Brugge were ignored. In this research we use Eclipse as the forward simulator and Destiny 6 as the streamline tracing engine, and we are able to take non-neighbor connections into consideration. Thus, the results are more flexible and realistic.

The reservoir permeability, porosity, and initial water saturation are shown in Fig. 13. The simulation period is 20 years, and the frequency of the injector/producer rate reallocation is every 4 months. Again, four different algorithms are run to compare NPV and recovery factor performance. The maximum available water in a field is 20,100 RB/day, the maximum injection/production rate per well is 6,000 RB/day, and the minimum producer bottom hole pressure is 500 psi.

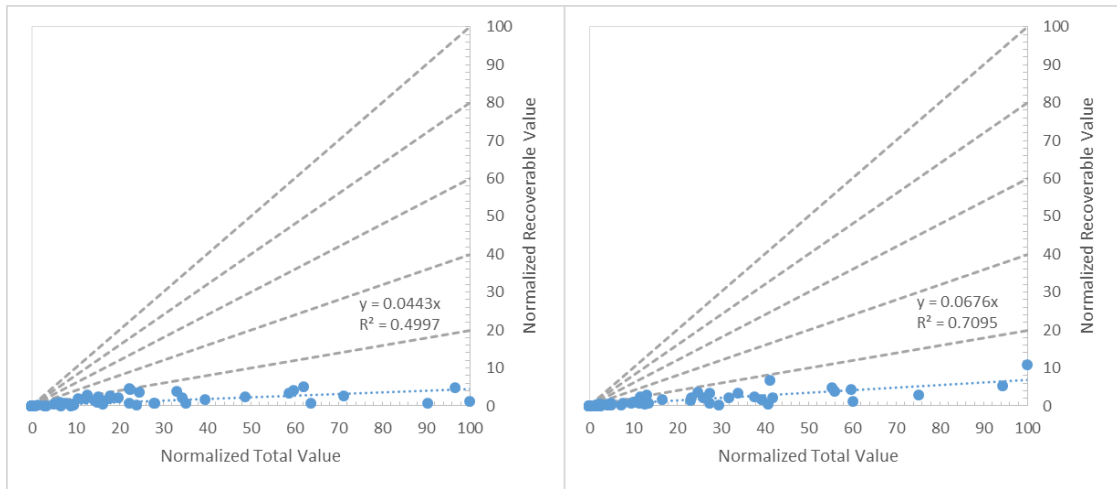


Fig. 36 – Brugge NPV efficiency plot of base case (left) and SLNPV case (right).

Fig. 36 shows the comparison of NPV efficiency between base case and SLNPV updated case for the first time step. The trend line shows around 4.43% average NPV efficiency for uniform rate operation, and the trend line shows that average NPV efficiency improves to 6.76% after using SLNPV to update the rates.

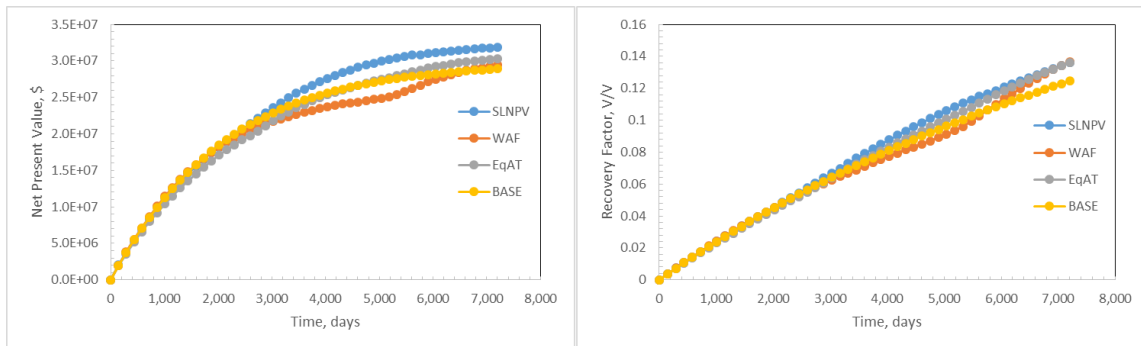


Fig. 37 - The comparisons of net present value (left) and recovery factor (right) for 4 different strategies (Brugge).

Fig. 37 shows the NPV and recovery performance for 4 different scenarios. SLNPV shows the best NPV performance throughout the simulation period. WAF shows the best reservoir recovery at the end of the simulation, and SLNPV shows the best recovery performance in the first 6,000 days.

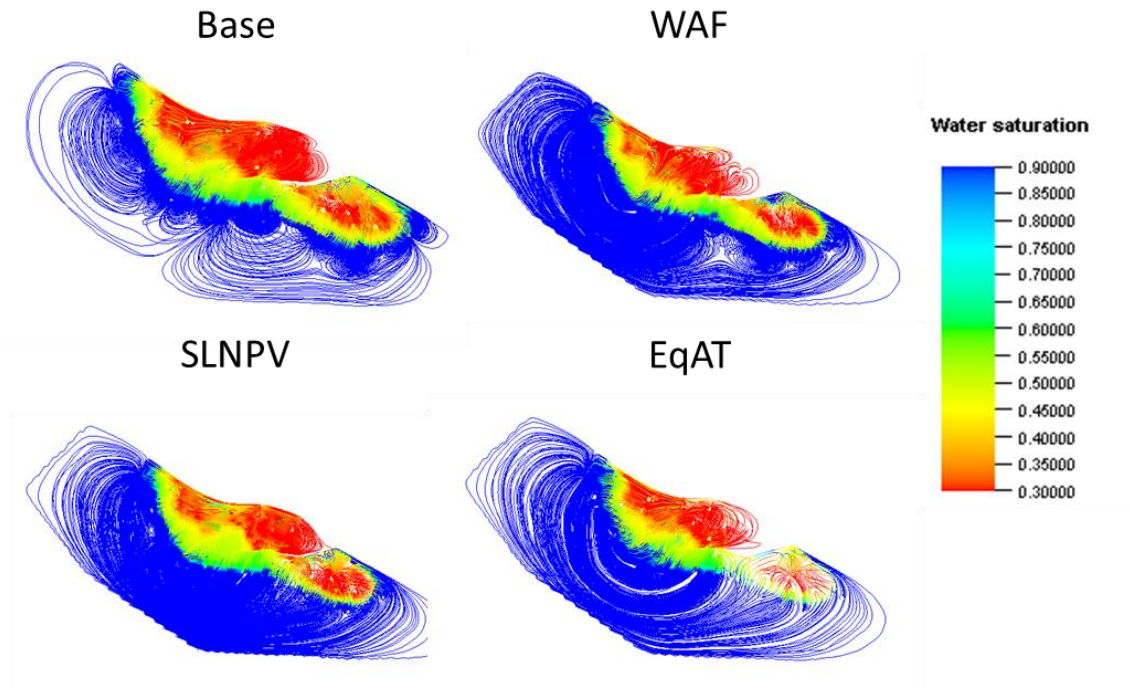


Fig. 38 - Streamline and water saturation distribution for 4 different scenarios at 4,500 days (Brugge).

Fig. 38 shows streamline and water saturation distribution at 4,500 days. All waterflooding optimization methods show improvement from the base uniform injection/production case. However, the focusing production regions for these optimization methods are very different. It is clear that SLNPV has the least oil saturation at the cone region, and EqAT has the least oil saturation at the lower right zone.

CHAPTER V

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusions

This study summarized the development and application of a post-simulation streamline tracing and reservoir management tool. Compared to the previous version, it shows big improvements in computational efficiency for large scale models, making it a good tool for field applications. The improved tool is demonstrated in flow diagnostics and rate allocation optimization problems.

For the new streamline tracing tool, a flux-based algorithm has been included to trace streamlines for non-neighbor connections. It is more reliable and case studies show streamlines can be successfully traced using this algorithm.

The tracing tool is applied to waterflooding optimization problems. Three optimization methods (WAF, EqAT, SLNPV) are tested to compare their performance and their advantages and disadvantages are explored in details.

The summary of all the work and findings are listed below:

- The new tool is applied to streamline and time of flight flow diagnostics. There is a huge improvement in computational time especially for cases which have many non-neighbor connections. Meanwhile, more streamlines can be successfully traced using the new algorithm for non-neighbor connections.

- The tool's robustness and efficiency make flow diagnostic plots available in both line view and grid view, especially for large scale field cases. It provides a more comprehensive image of reservoir and waterflooding performance.
- The new tool is used as the tracing engine to test different waterflooding optimization algorithms (WAF, EqAT, SLNPV). WAF can maximize oil production, EqAT can optimize sweep efficiency, and SLNPV can optimize field NPV.
- WAF is best in optimizing field recovery especially after water breakthrough occurs. EqAT has the best performance at the early stage of the operation, before the water breakthrough occurs. EqAT can also be used to reduce fluctuation in reallocating rates. SLNPV is best in optimizing field NPV, and it is flexible at any stage of the operation.
- SLNPV includes NPV efficiency as well as efficiencies from WAF and EqAT, making it a more integrated method.
- SLNPV is the only method that can detect whether the field is in economic operation and able to adjust total injection and production rates correspondingly.
- Incorporating all three optimization methods, we can optimize NPV or recovery factor depending on our need.

5.2 Recommendations

There are several recommendations from this study.

- Streamline properties calculated on grid may be used for streamline-based optimization and history matching. This can help take every grid cell into consideration, not just the grid cells where streamlines pass through.
- Destiny 6 NNC tracing algorithm is not robust in reversal flow tracing. Local boundary layering methods may be implemented to make the tracing tool more robust.
- The waterflooding optimization method needs to be applied for real field cases to test its advantages and limitations.
- Streamline-based waterflooding optimization may not be able to reach the global optimum because it makes rate adjustments based on the current operation and streamline distribution. It is possible for these optimization methods to get trapped at a local optimum scenario.
- The tested waterflooding optimization algorithms cannot shut in a producer or an injector because once a well is shut, it cannot be reopened. Thus, the minimum well rate constraints need to be investigated.

REFERENCES

- Agarwal, B. and Blunt, M. J. 2001. Full-Physics, Streamline-Based Method for History Matching Performance Data of a North Sea Field: Paper SPE-66388-MS Presented at the SPE Reservoir Simulation Symposium, 11-14 February, Houston, Texas, U.S.A., doi: 10.2118/66388-MS.
- Alhuthali, A., Oyerinde, A. and Datta-Gupta, A. 2007. Optimal Waterflood Management Using Rate Control: SPE Journal 10 (05): 539-551, doi: 10.2118/102478-PA.
- Datta-Gupta, A. 2000. Streamline Simulation: A Technology Update (includes associated papers 71204 and 71764): SPE Journal 52 (12): 68-84, doi: 10.2118/65604-JPT.
- Al-Khalifa, M.A. 2004. Advances in Generating and Ranking Integrated Geological Models for Fluvial Reservoir, Paper SPE 86999 presented at the 2004 Asia Pacific Conference on Integrated Modeling for Asset Management, Kuala Lumpur, Malaysia, 29-30 March.
- Bratvedt, F., Bratvedt, K., Buchholz, C. F., Holden, L., Holden, H. and Risebro, N. H. 1992. A New Front-Tracking Method for Reservoir Simulation: SPE Journal 7 (01): 107-116, doi: 10.2118/19805-PA.
- Bratvedt, F., Gimse, T., and Tegnander, C. 1996. Streamline Computations for Porous Media Flow Including Gravity: Transport Porous Media 25 (1): 63-78, doi: 10.1007/BF00141262.
- Cheng, H., Osako, I., Datta-Gupta, A. and King, M. J. 2006. A Rigorous Compressible Streamline Formulation for Two and Three-Phase Black-Oil Simulation: SPE Journal 11 (04): 407-417, doi: 10.2118/96866-PA.
- Coats, K. H. 2003. IMPES Stability: The CFL Limit: SPE Journal 8 (03): 291-297, doi: 10.2118/85956-PA.
- Cordes, C. and Kinzelbach, W. 1992. Continuous Groundwater Velocity Fields and Path Lines in Linear, Bilinear, and Rilinear Finite Elements: Water Resources Research 28 (11): 2903-2911, doi: 10.1029/92WR01686.
- Datta-Gupta, A. and King, M. J. 2007. *Streamline Simulation: Theory and Practice*, SPE Textbook series Vol. 11., Society of Petroleum Engineers. Richardson, Texas, U.S.A.
- Gilman, J. R., Meng, H., Uland, M. J., Dzurman, P. J., and Cosic, S. 2002. Statistical Ranking of Stochastic Geomodels Using Streamline Simulation: A Field

Application, Paper SPE 77374 presented at 2002 SPE Annual Technical Conference and Exhibition, San Antonio, 29 September – 2 October.

Giordano, R. M., Jayanti, S., Chopra, A., Yuan, H., Asakawa, K., Suleimani, A., Gheithy, A., and Edwards, C. 2007. A Streamline Based Reservoir Management Workflow to Maximize Oil Recovery, SPE 111143 presented at the 2007 SPE/EAGE Reservoir Characterization and Simulation Conference, Abu Dhabi, UAE, 28-31 October.

Grinestaff, G. H. 1999. Waterflood Pattern Allocations: Quantifying the Injector to Producer Relationship with Streamline Simulation, Paper SPE 54616 presented at 1999 SPE Western Regional Meeting, Anchorage, Alaska, 26-27 May.109

Harding, T. J., Radcliffe, N. J. and King, P. R. 1996. Optimisation of Production Strategies using Stochastic Search Methods: Paper SPE-35518-MS Presented at the European 3-D Reservoir Modelling Conference, 16-17 April, Stavanger, Norway. doi: 10.2118/35518-MS.

Jimenez, E., Datta-Gupta, A. and King, M. J. 2010. Full-Field Streamline Tracing in Complex Faulted Systems with Non-Neighbor Connections: SPE Journal 15 (01): 7-17, doi: 10.2118/113425-PA.

King, M.J. and Datta-Gupta, A.: Streamline Simulation: A Current Perspective, *In Situ* (1998), **22**, No. 1, 91.

Lax, P. and Wendroff, B. 1960. Systems of Conservation Laws: Communications on Pure and Applied Mathematics 13 (2): 217-237, doi: 10.1002/cpa.3160130205.

Milliken, W. J., Emanuel, A. S. and Chakravarty, A. 2001. Applications of 3D Streamline Simulation to Assist History Matching: SPE Journal 4 (06): 502-508, doi: 10.2118/74712-PA.

Moyner, O., Krostad, S., and Lie. K. A. 2014. Flow Diagnostics for use in Reservoir Management. SINTEF. Oslo, Norway.

Park, H. and Datta-Gupta, A. 2011. Reservoir Management Using Streamline-based Flood Efficiency Maps and Application to Rate Optimization: Paper SPE-144580-MS Presented at the SPE Western North American Region Meeting, 7-11 May, Anchorage, Alaska, U.S.A., doi: 10.2118/144580-MS.

Pollock, D. W. 1988. Semianalytical Computation of Path Lines for Finite-Difference Models: Ground Water 26 (6): 743-750, doi: 10.1111/j.1745-6584.1988.tb00425.x.

- Prevost, M., Edwards, M. G. and Blunt, M. J. 2002. Streamline Tracing on Curvilinear Structured and Unstructured Grids: SPE Journal 7 (02): 139-148, doi: 10.2118/78663-PA.
- Spall, J. C. 2005. Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. Wiley Series in Discrete Mathematics and Optimization, John Wiley & Sons. New York, U.S.A.
- Stenerud, V. R. and Lie, K. A. 2004. Streamline-Based History-Matching: A Review. SINTEF. Oslo, Norway.
- Stenerud, V. R., Kippe, V., Lie, K.A. and Datta-Gupta, A. 2008. Adaptive Multiscale Streamline Simulation and Inversion for High-Resolution Geomodels: SPE Journal 13 (01): 99-111, doi: 10.2118/106228-PA.
- Suwartadi, E. 2012. Gradient-based Methods for Production Optimization of Oil Reservoirs. Ph.D Thesis, Norwegian University of Science and Technology, Trondheim, Norway.
- Tanaka, S. 2014. Effective Reservoir Management Using Streamline-Based Reservoir Simulation, History Matching And Rate Allocation. PhD. Dissertation, Texas A&M University, College Station, Texas, U.S.A.
- Thiele, M. R. and Batycky, R. P. 2003. Water Injection Optimization Using a Streamline-Based Workflow: Paper SPE-84080-MS Presented at the SPE Annual Technical Conference and Exhibition, 5-8 October, Denver, Colorado, U.S.A., doi: 10.2118/84080-MS.
- Thiele, M. R. and Batycky, R. P. 2006. Using Streamline-Derived Injection Efficiencies for Improved Waterflood Management, SPE Reservoir Evaluation and Engineering 187.
- Wang, Y. and Kovscek, A. R. 2000. Streamline Approach for History Matching Production Data: Paper SPE-58350-PA Presented at the SPE/DOE Improved Oil Recovery Symposium, 3-5 April, Tulsa, Oklahoma, U.S.A., doi: 10.2118/58350-PA.
- Wu, Z. and Datta-Gupta, A. 2002. Rapid History Matching Using a Generalized Travel-Time Inversion Method: SPE Journal 07 (2): 113-122, doi: 10.2118/78359-PA.
- Yin, J., Park, H., Datta-Gupta, A. and Choudhary, M. K. 2010. A Hierarchical Streamline-Assisted History Matching Approach with Global and Local Parameter Updates: Paper SPE-132642-MS Presented at the SPE Western Regional Meeting, 27-29 May, Anaheim, California, U.S.A., doi: 10.2118/132642-MS.

APPENDIX

ANALYSIS OF DESTINY6 NNC CALCULATION

The discussion of NNC streamline tracing is divided into two scenarios: uni-directional flow scenario and reversal flow scenario. While the current implementation in Destiny 6 can handle uni-directional flow conditions efficiently, it has limitations on reversal NNC flow problems. Local boundary layering is by far the most general and robust approach for handling both situations. The analyses for both methods are discussed in this chapter, and stream function solutions will be used to compare with these two methods.

A.1 Uni-directional NNC Flow

The uni-directional NNC flow conditions mean that all the NNC fluxes are pointing to the same direction. In this situation, Destiny 6 algorithm, local boundary layering method, and stream function solutions all achieve the identical result for streamline NNC tracing.

A.1.1 Destiny 6 Algorithm

Destiny 6 uses a flux-based assumption to determine the exit or entry cell as well as its corresponding coordinate. If a streamline leaving a cell with multiple NNCs shown in Fig. 39, n is the number of all the fluxes in the NNC face, with F_1 amount of flux going to connection 1 and F_n amount of flux going to connection n .

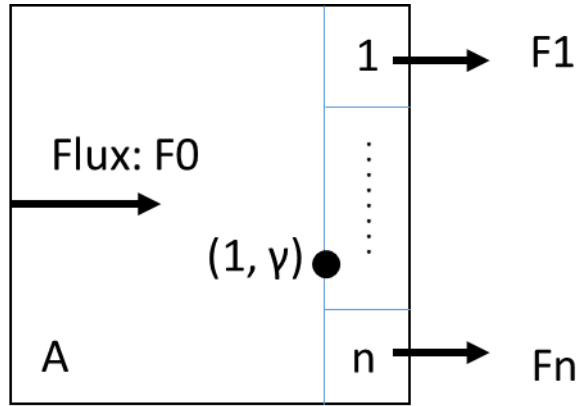


Fig. 39 – A streamline leaving a cell with multiple NNCs

First, Destiny 6 calculates all the flux cutoffs to determine which connection the streamline will enter. For example, if a streamline leaves at connection k , then the leaving

coordinate γ in cell A must be in the range of $[\sum_{i=k+1}^n F_i / \sum_{i=1}^n F_i, \sum_{i=k}^n F_i / \sum_{i=1}^n F_i]$. Second, it

rescales and normalizes γ in connection k as

$$\gamma' = (\gamma - \sum_{i=k+1}^n F_i / \sum_{i=1}^n F_i) / (F_k / \sum_{i=1}^n F_i) \text{ when } k \neq n \dots\dots\dots(A.1)$$

$$\gamma' = (\gamma) / (F_k / \sum_{i=1}^n F_i) \text{ when } k = n$$

Then, we can discuss the situation when a streamline entering a cell with multiple NNCs as shown in Fig. 40.

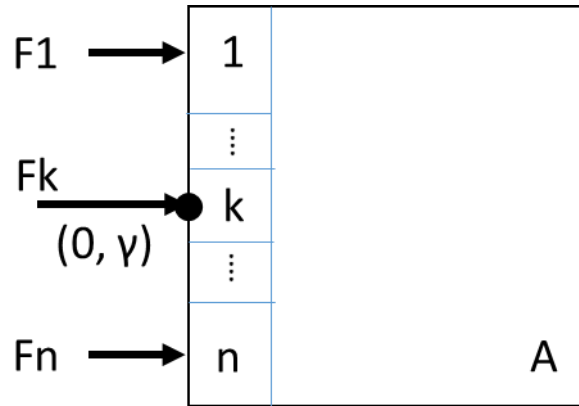


Fig. 40 - A streamline entering a cell with multiple NNCs

A streamline enters cell A from connection k at $(0, \gamma)$, the corresponding coordinate in cell A can be calculated as

$$\gamma_A = (\gamma * F_k + \sum_{i=k+1}^n F_i) / \sum_{i=1}^n F_i \text{ when } k \neq n \dots\dots\dots(A.2)$$

$$\gamma_A = (\gamma * F_k) / \sum_{i=1}^n F_i \text{ when } k = n$$

The underlying assumption for Destiny 6 NNC algorithm is the flux is uniform in the cell but variable on the connections. The other assumption is that the flux is incompressible across the connections. These assumption are flux conservative based on stream function derivation which will be discussed later as long as there is no reversal flux, so the results show good agreement with that of stream function solutions and local boundary layering method for uni-directional NNC flow situations.

A.1.2 Local Boundary Layering

The fundamental of local boundary layering method is to construct local cells near the boundary area and do cell by cell streamline tracing to calculate the entering and

exiting cells and coordinates. For a streamline leaving a cell with multiple NNCs, if we look at Fig. 39, we have n local cells constructed, and the flux entering the left hand side of the local cell is computed as

$$F_{i,left} = (A_k / \sum_{i=1}^n A_i) / \sum_{i=1}^n Fi \dots\dots\dots(A.3)$$

where A_k is the common area between the main cell A and the connection k.

Under the assumption that all the local cells are incompressible, we are able to get all the fluxes for all the local cells. We normalize the exiting coordinate from the main cell to a local cell, and then we perform cell by cell streamline tracing within all the local cells until we get an exit in the NNC face.

For a streamline entering a cell with multiple NNCs, we will follow a similar process. First, we construct local cells and calculate all the fluxes for all the entering local cells shown in Fig. 40. Then, we trace streamline line within these local cells until we find an entry in the main cell. Finally, we normalize the coordinate from the local cell to the main cell.

A.1.3 Stream Function Solutions

We use Fig. 39 again for a streamline leaving a cell with multiple NNCs. If we construct local cells the same as local boundary layering method, we know that there will be n possible exits for a streamline. We first calculate the stream function for each local cells by setting $\psi_k(0,0)=\psi_{k+1}(0,1)$. If a streamline enters cell A at $(1, \gamma)$, because the stream

function has a constant value along a streamline, we have $\psi_{\text{enter}}(1,\gamma)=\psi_{\text{exit}}(1,\gamma_k)$, so the exit for each connection γ_k ($k=1$ to n) can be calculated as

$$\begin{aligned} \gamma_k &= \left(\sum_{i=1}^n F_i * \gamma - \sum_{i=k+1}^n F_i \right) / F_k \text{ when } k < n \\ \gamma_k &= \left(\sum_{i=1}^n F_i * \gamma \right) / F_k \text{ when } k = n \end{aligned} \dots\dots\dots(\text{A.4})$$

For uni-directional flow scenarios, stream function is strictly monotonic. The proof can be shown in Fig. 41. This is an arbitrary local boundary cell constructed in NNC tracing, and the NNC face is either the left or the right face of the cell. By knowing all fluxes, we can solve for its stream function as

$$\psi = (F_{x2} - F_{x1}) * xy + F_{x1} * y - F_{y1} * x + \psi_{\text{ref}} \dots\dots\dots(\text{A.5})$$

To prove the stream function is monotonic in uni-directional flow scenarios, we take derivative of the stream function with y, so we have

$$\frac{\partial \psi}{\partial y} = (F_{x2} - F_{x1}) * x + F_{x1} = F_{x2} * y + F_{x1} * (1 - x) \dots\dots\dots(\text{A.6})$$

Since F_{x1} and F_{x2} are in the same direction and x is between 0 and 1, so

$$\frac{\partial \psi}{\partial y} > 0 \text{ for } F_{x1}, F_{x2} > 0 \text{ or } \frac{\partial \psi}{\partial y} < 0 \text{ for } F_{x1}, F_{x2} < 0. \text{ This is true for all the constructed}$$

local cells so stream function is strictly monotonic in uni-directional flow conditions.

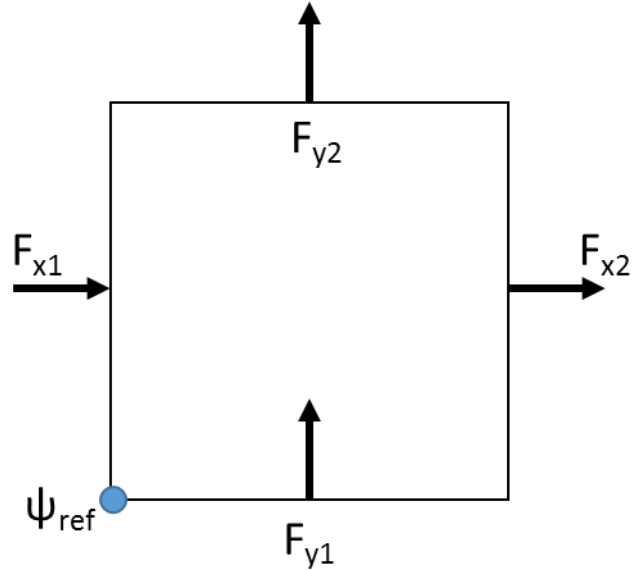


Fig. 41 - A zoom-in view of a local unit.

Thus, only one γ_k is in the range of $[0,1]$ can satisfy $\psi_{\text{enter}}(1,\gamma)=\psi_{\text{exit}}(1,\gamma_k)$, so if $0 \leq \gamma_k \leq 1$, then the cell with flux F_k is the next cell.

We use Fig. 40 for a streamline entering a cell with multiple NNCs. Under the uni-directional flow conditions, the streamline has to enter cell A using the assumption that $\psi_{\text{enter}}(1,\gamma)=\psi_{\text{exit}}(1,\gamma_k)$. The entering coordinate in cell A can be calculated as

$$\begin{aligned} \gamma_A &= (\gamma * F_k + \sum_{i=k+1}^n F_i) / \sum_{i=1}^n F_i \text{ when } k \neq n \\ \gamma_A &= (\gamma * F_k) / \sum_{i=1}^n F_i \text{ when } k = n \end{aligned} \dots\dots\dots(\text{A.7})$$

Stream function solutions are identical to local boundary layering method in uni-directional flow conditions because stream function is strictly monotonic and a constant value is the trajectory for a streamline. In other words, given any entry for a stream function, there will be a unique exit solution whose stream function value is equal to

stream function value of this entry point while satisfying the constraints of γ in the range of $[0, 1]$, and this is the unique trajectory for the streamline. Thus, stream function solutions will achieve the same results as local boundary layering in uni-directional flow conditions.

Stream function solutions are also identical to Destiny 6's implementation in uni-directional flow conditions. For streamline leaving a cell problem, in order for γ_k to fall into the range of $[0, 1]$ in Eq. A.4, γ must be in range of $[\sum_{i=k+1}^n Fi / \sum_{i=1}^n Fi, \sum_{i=k}^n Fi / \sum_{i=1}^n Fi]$, which is the criteria in Destiny 6 for the streamline to enter connection k. Thus, the only difference is that Destiny 6 tests which cell the streamline will enter, while stream function solutions use the equation to solve for all possible solutions and choose the one that physically makes sense. The equations used are identical. Meanwhile, the streamline entering equations are exactly the same because the entering cell is already known.

All three methods give identical solutions in uni-directional flow conditions, but stream function solution and Destiny 6's algorithm may provide more computational efficiency because they do not require construction of local cells and they have analytical expressions for finding the entry or exit cells and coordinates.

A.2 Reversal NNC Flow

For reversal NNC flow problems, local boundary layering method is the only method that is still robust and able to provide a unique streamline trajectory accurately.

Both stream function solutions and Destiny 6's algorithm have limitations in handling these situations, and examples are given below to show when the problems occur.

A.2.1 Destiny 6 Algorithm

Destiny 6 handles the NNC reversal flow problems by assigning zero value to the flux whose direction is opposite to the gross flux direction when setting up the flux cutoff criteria. However, it has limitations in dealing with the situation where the gross flux is zero. A sample problem is used to demonstrate the limitations of Destiny 6 algorithm in Fig. 42.

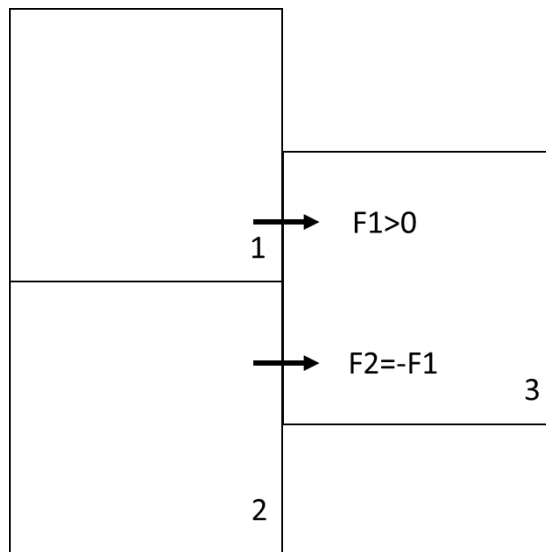


Fig. 42 - NNC reversal problem for limitations of Destiny 6 algorithm.

There is a streamline going from cell 1 to cell 3 with the flux F_1 , and there is F_2 amount of flux leaving cell 3 at the same NNC face, and $F_3 = -F_1$. In this situation, if we

apply Eq. A.2, we encounter a divided by zero error because $\sum_{i=1}^n F_i = F_1 + F_2 = 0$. This

error forces the streamline to terminate at cell 3.

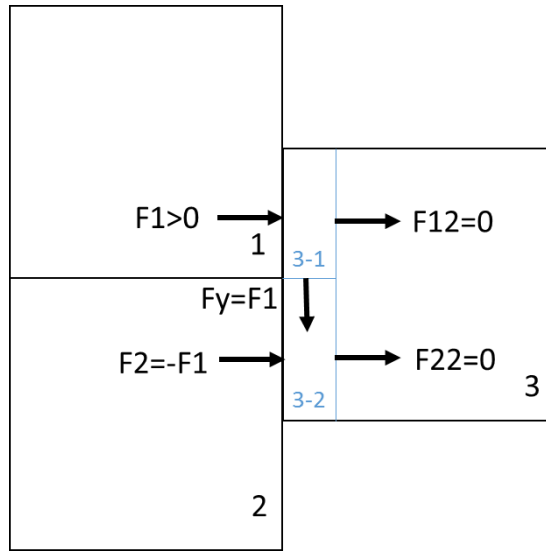


Fig. 43 - NNC reversal problem solved by local boundary layering.

This problem can be solved by local boundary layering method as shown in Fig. 43. First, it constructs two local cells 3-1 and 4-1 and calculates the fluxes for these cells. When a streamline enters cell 3-1 from the left, after performing a single cell tracing, it will leave cell 3-1 at the bottom to enter cell 3-2 because there is no flux at other face. Performing a single cell tracing again, this streamline will leave cell 3-2 at the left face to enter cell 2. Thus, any streamline trying to enter cell 3 from cell 1 will go through local cells 3-1 and 3-2 to enter cell 2.

A.2.2 Stream Function Solutions

Stream function solution cannot handle the flow reversal problems because of the non-uniqueness of stream function solutions. If we revisit Eq. A.6, in uni-directional flow problems, F_{x1} and F_{x2} are either both positive or both negative, resulting in $\frac{\partial \psi}{\partial y}$ is either positive or negative for all the constructed local cell. Thus, given a stream function value for the entry point, there will be a unique exit point whose stream function value matches that of the entry point while remain in the range of $[0,1]$. When reversal flows exist, $\frac{\partial \psi}{\partial y}$ can either be positive or negative, and stream function is no long monotonic. Therefore, multiple exit points whose stream function value is equal to that of the entry point may exist, and there is no good way to distinguish which exit is the correct trajectory for this streamline.

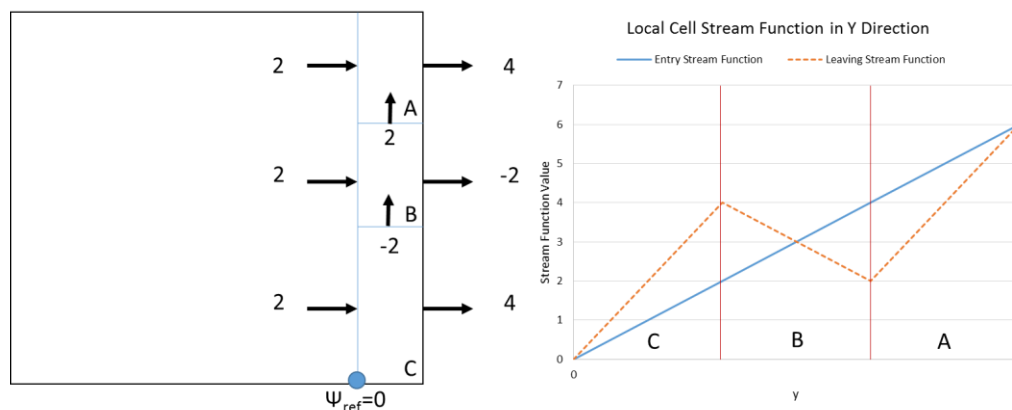


Fig. 44 – NNC Reversal flow problem for non-uniqueness in stream function solutions.

A reversal problem is shown in Fig. 44, with connection B has a flux in the opposite direction to the gross flux. The stream function for each local cells are calculated as

$$\begin{aligned}
 \psi_A &= 2 * xy + 2 * y - 2 * x + 4 \\
 \psi_B &= -4 * xy + 2 * y + 2 * x + 2 \dots\dots\dots(A.8) \\
 \psi_C &= 2 * xy + 2 * y
 \end{aligned}$$

With the stream functions calculated, we can plot the stream function value along γ direction in both entry face and leaving face. As expected, the leaving face stream function is not monotonic, any stream function value of the entry point between 2 and 4 can have physical exiting solutions in both cell A and cell C, but only one of these solutions is correct. This example shows that stream function solutions are not robust in reversal flow situations. Local boundary layering method does not have this issue because a streamline will have a unique trajectory.