

THE IMPACT OF GRID GEOMETRY ON DISPLACEMENT CALCULATIONS

A Thesis

by

EDUARDO ANTONIO JIMENEZ ARISMENDI

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

MASTER OF SCIENCE

August 2004

Major Subject: Petroleum Engineering

THE IMPACT OF GRID GEOMETRY ON DISPLACEMENT CALCULATIONS

A Thesis

by

EDUARDO ANTONIO JIMENEZ ARISMENDI

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Akhil Datta-Gupta
(Chair of Committee)

Daulat D. Mamora
(Member)

Binayak P. Mohanty
(Member)

Stephen A. Holditch
(Head of Department)

August 2004

Major Subject: Petroleum Engineering

ABSTRACT

The Impact of Grid Geometry on Displacement Calculations.

(August 2004)

Eduardo Antonio Jimenez Arismendi, B.S., Universidad Industrial de Santander,
Colombia

Chair of Advisory Committee: Dr. Akhil Datta-Gupta

Reservoir simulation models are becoming increasingly sophisticated in tandem with the rapid development of geological modeling methods. Widely used commercial simulators usually model flow through heavily faulted and structurally complex geometries with the flexibility provided by corner-point geometry. However, the nonorthogonality component present within these frameworks may compromise the solution accuracy of the model and the subsequent operational decisions involved.

We propose a systematic methodology to evaluate the impact of complex gridding introducing a new streamline formulation for corner-point geometry. Based on a new time-like variable, the new formulation provides a significantly simpler and more robust development to handle the complexity in structurally demanding and faulted systems. It retains the simplicity and speed of streamline-based flow models and provides an efficient way to visualize nonorthogonal effects.

Applied to various geometries showing challenging features of geology and flow, the displacement fronts obtained from streamline-derived analytic calculation identified the discrepancies characteristic between known solutions and results from two widely used commercial simulators.

DEDICATION

To my father's memory, my beloved mother and my huge and loving family.

ACKNOWLEDGMENTS

First of all, I'd like to thank my advisor and committee chair, Dr. Akhil Datta-Gupta, for his continuous encouragement, financial support, and especially for his academic and creative guidance. I'd also like to thank Dr. Michael King for his thoughtful and challenging ideas and Dr. Eduardo A. Idrobo for his priceless help and friendship.

Last but not least, my deepest gratitude to the tough guys in rooms 701 and 702. It's been a privilege to work hand in hand with this amazing crew.

Thank you very much.

TABLE OF CONTENTS

	Page
ABSTRACT.....	iii
DEDICATION..	iv
ACKNOWLEDGMENTS	v
TABLE OF CONTENTS.....	vi
LIST OF FIGURES	viii
CHAPTER I INTRODUCTION.....	1
1.1 Motivation and Literature Review	2
1.2 Objective of Study	5
CHAPTER II STREAMLINE-BASED SIMULATION.....	6
2.1 Governing IMPES Equations.....	7
2.2 Coordinate Transform.....	9
2.3 Time Stepping.....	10
CHAPTER III STREAMLINE TRACING AND TIME OF FLIGHT	
CALCULATION	13
3.1 Cartesian Grid Tracing.....	14
3.1.1 Pollock's Interpolation.....	14
3.2 Corner Point Geometry Grid Tracing	19
3.2.1 Unit Space.....	23
3.2.1.1 Pseudo-Time of Flight.....	23
3.2.1.2 Unit Cube Streamline Tracing.....	25
3.2.2 Real Space.....	26
3.2.2.1 Isoparametric Transform	26
3.2.2.2 Time of Flight Calculation.....	28
CHAPTER IV RESULTS AND APPLICATIONS	32
4.1 Orthogonal Grids	34
4.1.1 Homogeneous Quarter of a Five Spot.....	34
4.1.2 Homogeneous X-Z Cross-Section	36
4.2 Non-Orthogonal Grids	37
4.2.1 Homogeneous Quarter of a Five Spot.....	38
4.2.2 Homogeneous X-Z Cross-Section	39
4.3 Gridding Effects.....	41
4.3.1 Pinch-Outs.....	43
4.3.1.1 Gradual Pinching.....	43
4.3.1.2 Layer Discontinuity.....	46
4.3.2 Onlap and Zone Boundaries.....	49
4.3.2.1 Closed Bottom Boundary	50

	Page
4.3.2.2 Open Bottom Boundary.....	54
4.4 Grid Refinement Effects	57
4.4.1 Vertical Coordinate Lines	58
4.4.2 Slanted Coordinate Lines.....	61
CHAPTER V CONCLUDING REMARKS AND RECOMMENDATIONS	64
5.1 Conclusions.....	64
5.2 Recommendations.....	65
REFERENCES.....	66
VITA.....	69

LIST OF FIGURES

FIGURE	Page
1.1 Complex Geologic Features Grid Examples.....	5
3.1 Finite Difference Cell Showing xyz Definitions.	14
3.2 Schematic Showing the Computation of Exit Point and Travel Time in 2D....	18
3.3 Isoparametric Transformation.....	21
3.4 Schematic Showing the Pseudo-Time of Flight in a Unit Cube	24
3.5 Schematic Showing the Exit Coordinates Computation in a Unit Cube	26
3.6 The Eight Corners of a Corner Point Cell.....	27
3.7 Isoparametric Transformation Unit to Real Space.....	28
4.1 Streamline Methodology for CPG Grids	32
4.2 Quarter of a Five-Spot Orthogonal Grid.....	35
4.3 Time of Flight - Quarter of a Five-Spot Orthogonal Grid	35
4.4 Orthogonal X-Z Cross-Section Grid.....	36
4.5 Time of Flight - Orthogonal X-Z Cross-Section Grid	37
4.6 Non-Orthogonal Quarter of a Five-Spot Grid.....	38
4.7 Time of Flight - Quarter of a Five-Spot Non-Orthogonal Grid.....	39
4.8 Numerical Solutions - Non-Orthogonal X-Z Cross-Section Grid	40
4.9 Numerical Time of Flight - Non-Orthogonal X-Z Cross-Section Grid	41
4.10 Analytical Solutions - Non-Orthogonal X-Z Cross-Section Grid.....	42
4.11 Analytical Time of Flight - Non-Orthogonal X-Z Cross-Section Grid	42
4.12 Numerical Solution - Gradual Pinch-out Grid.....	44
4.13 Numerical Time of Flight - Gradual Pinch-out Grid	44
4.14 Analytical Solution - Gradual Pinch-out Grid	45
4.15 Analytical Time of Flight - Gradual Pinch-out Grid	45
4.16 Difference between Numerical and Analytical TOF - Pinch-out Grid.....	46
4.17 Numerical Solution - Pinch-out Grid.....	46
4.18 Numerical Time of Flight - Pinch-out Grid	47
4.19 Analytical Solution - Pinch-out Grid.....	48

FIGURE	Page
4.20 Analytical Time of Flight - Pinch-out Grid	48
4.21 Difference between Numerical and Analytical TOF -Pinch-out Grid	49
4.22 Numerical Solution - Onlap Boundary Grid – Closed Bottom.....	50
4.23 Numerical Time of Flight - Onlap Boundary Grid – Closed Bottom.....	51
4.24 Stream-Function Concept for On-lap Sequence	51
4.25 Analytical Solution - Onlap Boundary Grid – Closed Bottom.....	52
4.26 Analytical Time of Flight - Onlap Boundary Grid – Closed Bottom	53
4.27 Difference between Numerical and Analytical TOF - Onlap Boundary Grid – Closed Bottom	53
4.28 Numerical Solution - Onlap Boundary Grid – Open Bottom	54
4.29 Numerical Time of Flight - Onlap Boundary Grid – Open Bottom	55
4.30 Analytical Solution - Onlap Boundary Grid – Open Bottom	56
4.31 Analytical Time of Flight - Onlap Boundary Grid – Open Bottom.....	57
4.32 Difference between Numerical and Analytical TOF - Onlap Boundary Grid – Closed Bottom	57
4.33 Refined X-Direction Flow i-Face - Vertical Coordinate Lines	59
4.34 Refined X-Direction Flow k-Face - Vertical Coordinate Lines.....	59
4.35 Refined Time of Flight - Vertical Coordinate Lines.....	60
4.36 Refined Streamline Paths - Vertical Coordinate Lines.....	60
4.37 Refined X-Direction Flow in i-Face - Slanted Coordinate Lines	61
4.38 Refined X-Direction Flow in k-Face - Slanted Coordinate Lines	62
4.39 Refined Time of Flight - Slanted Coordinate Lines.....	62
4.40 Refined Streamline Paths - Slanted Coordinate Lines.....	63

CHAPTER I

INTRODUCTION

Streamline-based flow simulation (SL) has received significant attention over the past 5 years, and is now accepted as an effective and complementary technology to more traditional flow modeling approaches such as finite differences (FD). Streamline-based flow simulation is particularly effective in solving large, geologically complex and heterogeneous systems, where fluid flow is dictated by well positions and rates, rock properties (permeability, porosity, and fault distributions), fluid mobility (phase relative permeabilities and viscosities), and gravity. Capillary pressure effects, surface group constraints and expansion-dominated systems, on the other hand, are not modeled efficiently by streamlines.

Modern SL simulation rests on 6 key principles: (1) tracing three-dimensional (3D) streamlines in terms of time-of-flight (TOF); (2) recasting the mass conservation equations along streamlines in terms of TOF; (3) periodic updating of streamlines; (4) numerical 1D transport solutions along streamlines; (5) accounting for gravity effects using operator splitting; and (6) extension to compressible flow. This research will address the increasing challenge present in streamline tracing within complex geometries.

The usefulness and uniqueness of SL simulation rests in the context of what are generally considered important issues in reservoir simulation: (1) upscaling; (2) quantifying displacement efficiency; (3) computational speed; (4) history matching; and (5) field optimization. In addition, novel, streamline-specific data is discussed in the context of injector/producer efficiencies and as a unique aid in upscaling by allowing engineers to go beyond the usual approach of only matching reference solutions.

The nature of the method permits a semi-analytical compositional one dimensional treatment along streamlines and is able to minimize dispersivity effects even when using numerically computed saturation profiles. Furthermore, the method's popularity relies in part on the decoupling of the pressure and component mass conservation equations, leading to fairly large time stepping in the pressure equation and at least an order of magnitude speed up factor. Consequently, streamline simulators can routinely run problem of sizes that go from a few thousands blocks for gas injection and compositional models up to several millions blocks for simpler flow (e.g immiscible displacements).

1.1. Motivation and Literature Review

Today's streamline simulation was preceded by at least four other methods for modeling convection-dominated flow in the reservoir. Line-source/sink methods have been widely used by the petroleum industry.^{1,2} These methods use analytic solutions to the pressure and velocity distribution in the reservoir. The primary limitation of these methods is the requirement for homogeneous properties and constant reservoir thickness.

Streamtube methods are more general and have been applied successfully for field-scale modeling of waterflooding and miscible flooding.³⁻⁵ In these methods, the flow domain is divided into a number of streamtubes and fluid-saturation calculations are performed along these streamtubes. However, the need to keep track of the streamtube geometries can become quite cumbersome in three dimensions. Thus, most applications of streamtube methods have been limited to two dimensions or some form of hybrid approaches to account for 3D effects.

Particle-tracking methods have been used by the oil industry to model tracer transport in hydrocarbon reservoirs and also for groundwater applications.⁶ These methods track the movement of a statistically significant collection of particles along appropriate pathlines; while they generally work well near steep fronts, they do not work

as well for smooth profiles. Another drawback is the loss of resolution of the front with the progression of time and the statistical variance in the concentration response. Finally, front-tracking methods introduce fluid fronts as a degree of freedom in computation.⁷⁻⁹ The primary limitations of these methods are the computational burden associated with complications that arise from the close approach or intersection of frontal contours.

Muskat¹ gave an early description to the governing analytical equations that define the stream function and potential function in simple two dimensional domains for incompressible flow. A notable work with these definitions was by Fay and Pratts¹⁰, who developed a numerical model to predict tracer and two-phase flow on a two-well homogenous 2D system.

Datta-Gupta & King⁶ introduced the concept of “time of flight” along a streamline. This idea shall be used in this research quite extensively. They also presented a streamline model for 2D heterogeneous areal displacements of two well-tracer and waterflooding problems. Most of the current streamline based flow simulators use this concept of time of flight, because of its ease and its decoupling effect, which splits a 3D problem into a 1D problem. This has been the most significant contribution in streamline simulation. The present research work also builds on this concept of ‘time of flight’.

Pollock’s¹¹ proposed a piece-wise linear interpolation of the velocity field within a grid block which significantly improved the original Runge-Kutta streamline tracing technique used by Shafer¹². Pollock tracing was successfully used in a number of streamline simulators where appropriate flow modeling along the streamlines allowed for simulation of first contact miscible displacements (King et al.¹³) and evaluation of the effects of reservoir heterogeneity. A main limitation of the method was the inherent assumption that fluid is linear along the streamline forbidding lateral flow across streamlines. Martin et al¹⁴. showed streamtube models failed predicting waterflood performance for an isolated five-spot pattern under favorable mobility ratio which

highlighted the need to update the streamlines to accurately account for non-linear viscous effects.

Blunt et al¹⁵. extended the streamline method to three dimensional systems, accounting for longitudinal and transverse diffusion as well as gravity effects. Batycky¹⁶ introduced an operator splitting technique similar to that of front tracking methods, allowing him to account for multiphase gravity effects.

As we already mentioned, tracing of streamlines currently rests on Pollock's¹¹ bilinear interpolation, which in turn makes the fundamental assumption that there is a single velocity per cell face. Structurally complex reservoirs¹⁷, on the other hand, often require multiple connections across a single face, as might be the case in the presence of faults. This adds a layer of complexity to the tracing algorithm, since cells might now have multiple velocities across a single face, theoretically even in opposite directions¹⁸. The streamline paths in a cell must now be traced through sub-cells (the geometries of which are dictated by the velocity vectors on each face) and to which Pollock's algorithm can be applied.

In tandem with the complexity associated with structurally demanding models¹⁹⁻²¹, our main motivation is give SL simulation the ability to model flow through heavily faulted and structurally complex systems while retaining much of its simplicity and speed. This is an important extension to the technology and will likely allow its use for systems where traditionally more sophisticated meshing algorithm are used (such as finite-element or PEBI-grids).

As a whole, the industry is still exploring the most optimal use of this technology and how it might be efficiently integrated into the current work flows used by individual companies²²⁻²⁵. The next few years will bring a further maturing and extended application of the technology. It is not unreasonable to expect that most all companies using conventional simulation technology today will in one form or another use SL simulation in their work.

1.2. Objective of Study

The main objective of this research is to evaluate grid geometry effects on displacements calculations using the streamline approach. Followings are the basic objectives:

- Provide a new formulation to trace streamlines in Corner Point Geometry Grids and Unstructured Grids.
- Extend the tracing algorithm to complex geologic features such as faults, onlap boundaries and distorted grids. A few examples of such grids are shown in Fig. 1.1.
- Provide a basis for time of flight calculations and use it as an independent indicator of grid geometry effects on displacements calculations.
- Implementation of this procedure in a FORTRAN program for tracing streamlines in the grid domain and the corresponding time of flight calculation.
- Compare the results with existing commercial simulators to examine the accuracy of all the involved calculations.

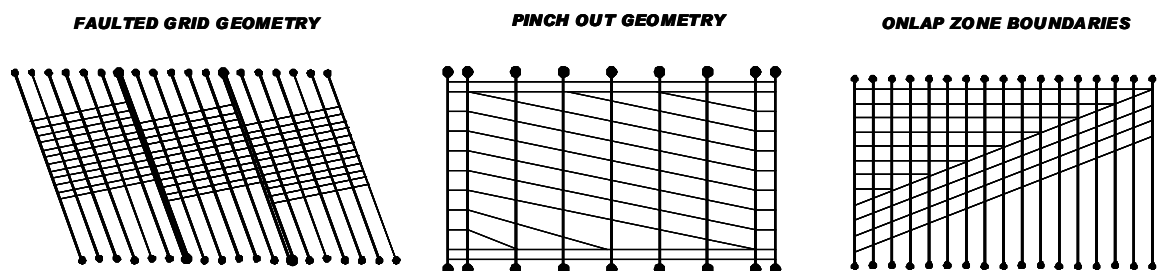


Fig. 1.1 Complex Geologic Features Grid Examples

CHAPTER II

STREAMLINE-BASED SIMULATION

Streamline simulators approximate 3D fluid-flow calculations by a sum of 1D solutions along streamlines. The choice of streamline directions for the 1D calculations makes the approach extremely effective for modeling convection-dominated flows in the reservoir. This is typically the case when heterogeneity is the predominant factor governing flow behavior.

A key underlying concept in streamline simulation is the isolation of geologic heterogeneity from the physics of flow calculations. Mathematically, this is accomplished by use of the streamline time of flight as a coordinate variable. We move to a coordinate system where all streamlines are straight lines and distance is replaced by the time of flight. The impact of reservoir heterogeneity is embedded in the time of flight and trajectory of the streamlines. The physical process calculations are reduced to 1D solutions along streamlines. The streamlines generally are distributed in space with higher resolution than the underlying spatial grid, thus providing excellent transverse resolution. Saturation calculations along streamlines are decoupled from the underlying grid and can be carried out with little or no intrinsic timestep limitations.

Streamline simulation involves the following basic steps:

1. Trace the streamlines on the basis of a velocity field, typically derived numerically with finite-difference or finite-element methods.
2. Compute particle travel time or time of flight along the streamlines. The time-of-flight coordinate provides a quantitative form of flow visualization that can have a variety of applications in reservoir characterization/management.

3. Solve the transport equations (saturation and concentration) along streamlines. The transport calculations are performed in the time-of-flight coordinate, effectively decoupling heterogeneity effects and significantly simplifying calculations.
4. Periodically update the streamlines to account for mobility effects or changing field conditions. Once the streamlines are regenerated, recompute the time of flight along the new streamlines. Finally, saturation calculations are resumed with the updated time of flight. A critical step here is the mapping of information from the old streamlines to the new streamlines. This can be a potential source of error during streamline simulation.

The computational advantage of the streamline methods can be attributed to four principal reasons: (1) streamlines may need to be updated only infrequently; (2) the transport equations along streamlines often can be solved analytically; (3) the 1D numerical solutions along streamlines are not constrained by the underlying geologic grid-stability criterion, thus allowing for larger timesteps; and (4) for displacements dominated by heterogeneity, the computation time often scales nearly linearly with the number of gridblocks, making it the preferred method for fine-scale geologic simulations.

Furthermore, the self-similarity of the solution along streamlines may allow us to compute the solution only once and map it to the time of interest. Other advantages are subgrid resolution and reduced numerical artifacts, such as artificial diffusion and grid orientation effects, because the streamline grid used to solve the transport equations is effectively decoupled from the underlying static grid.

2.1. Governing IMPES Equations

The streamline method is an IMPES method. Ignoring capillary and dispersion effects, the governing equation in terms of pressure P for incompressible multiphase flow in porous media is given by

$$\nabla \cdot \overline{\overline{K}} \cdot (\lambda_f \nabla P + \lambda_g \nabla D) = 0 \quad (2.1)$$

Where the total mobility (λ_f) and the total gravity mobility (λ_g) are defined as

$$\lambda_f = \sum_{j=1}^{n_p} \frac{k_{rj}}{\mu_j} \quad \lambda_g = \sum_{j=1}^{n_p} \frac{k_{rj} \rho_j \mathbf{g}}{\mu_j} \quad (2.2)$$

D represents a depth below datum. To determine the flow of the individual phases we also require a material balance equation for each phase j

$$\phi \frac{\partial S_j}{\partial t} + \bar{u}_t \cdot \nabla f_j + \nabla \cdot \bar{G}_j \quad (2.3)$$

The total velocity \bar{u}_t is derived from the 3D solution to the pressure field and the application of Darcy's Law. The phase fractional flow term is given by

$$f_j = \frac{k_{ri} / \mu_j}{\sum_{i=1}^{n_p} k_{ri} / \mu_i} \quad (2.4)$$

And the gravity fractional flow is given by

$$\bar{G}_j = \overline{\overline{K}} \cdot \mathbf{g} \nabla D f_j \sum_{i=1}^{n_p} k_{ri} / \mu_i (\rho_i - \rho_j) \quad (2.5)$$

In a conventional IMPES finite-difference simulator Eq. 2.3 is solved in its full three-dimensional form. With the streamline method, we decouple the 3D equation into multiple 1D equations that are solved along streamlines. For large problems, solving multiple 1D equations is much faster and more accurate than solving the full 3D problem.

2.2. Coordinate Transform

Streamlines are launched from gridblock faces containing injectors. As the streamlines are traced from injectors to producers, we determine the time-of-flight along the streamline, which is defined as

$$\tau = \int_0^s \frac{\phi}{u_t(\zeta)} d\zeta \quad (2.6)$$

This equation gives the time required to reach a point s on the streamline based on the total velocity $u_t(\zeta)$ along the streamline. For orthogonal geometries it is possible to determine the coordinate transform rewriting equation 2.6 as

$$\frac{\partial \tau}{\partial s} = \frac{\phi}{|u_t|} \quad (2.7)$$

This can be rewritten as,

$$|u_t| \frac{\partial}{\partial s} \equiv \vec{u}_t \cdot \nabla = \phi \frac{\partial}{\partial \tau} \quad (2.8)$$

Substituting equation 2.8 into equation 2.3 gives

$$\frac{\partial S_j}{\partial t} + \frac{\partial f_j}{\partial \tau} + \frac{1}{\phi} \nabla \cdot \vec{G}_j = 0 \quad (2.9)$$

This equation is the governing pseudo-1D material balance equation for phase j transformed along a streamline coordinate. It is pseudo-1D since the gravity term is typically not aligned along the direction of a streamline.

To solve equation 2.9 we simply split the equation into two parts. First a convective step along the streamlines governed by

$$\frac{\partial S_j^c}{\partial t} + \frac{\partial f_j}{\partial \tau} = 0 \quad (2.10)$$

This includes boundary conditions at the wells, is taken to construct an intermediate saturation distribution S_j^c . Then, a gravity step is taken along the gravity lines and saturations are moved using

$$\frac{\partial S_j}{\partial t} + \frac{g}{\phi} \frac{\partial G_j}{\partial z} = 0 \quad (2.11)$$

For simplicity it is assumed that the gravity lines are aligned in the z coordinate direction. Equation 2.10 is solved numerically using single point upstream weighting scheme explicit in time. Equation 2.11 is solving using an explicit upstream weighting method. An additional advantage of decoupling equation 2.9 in this way is that equation 2.11 is only solved in flow regions where gravity effects are important. For example, in locations where fluids are completely segregated, equation 2.11 will not be solved, since $\partial G/\partial z = 0$

2.3. Time Stepping

To model field scale displacements our underlying assumption is that the streamline paths change with time due to the changing mobility field and/or changing boundary conditions. Thus the pressure field is updated periodically in accordance with these changes. By using numerical solutions along the recalculated streamline paths the method accounts for the non-uniform initial conditions now present along the recalculated paths.

To move the 3D solution forward in time from t^n to $t^{n+1} = t^n + \Delta t^{n+1}$ the following algorithm is used:

1. At the start of a new time step, t^{n+1} , solve for the pressure field P using equation 2.1 in the IMPES formulation. This equation may be solved using a standard seven-point finite difference scheme, with no-flow boundary conditions over the surface of the domain and specified pressure or rate at the wells.
2. Apply Darcy's law to determine the total velocity at gridblock faces.
3. Trace streamlines from injectors to producers. For each streamline the following is performed: (a) While tracing a streamline, the current saturation information from each gridblock that the streamline passes through is remembered. In this manner, a profile of saturation versus τ is generated for the new streamline; (b) Move the saturations forward by Δt^{n+1} by solving equation 2.10 numerically in 1D. Map the new saturation profile back to the original streamline path.
4. Average all the streamline properties within each gridblock of the 3D domain to determine the saturation distribution at t^{n+1}
5. If $G_j \neq 0$ include a gravity step that traces gravity lines from the top of the domain to the bottom of the domain along \vec{g} . For each gravity line the following is done (a) While tracing a gravity line, the saturation distribution calculated in the convective step as a function of z is remembered; (b) The saturations are moved forward by Δt^{n+1} using equation 2.11. The new saturation profile is mapped back to the original gravity line.
6. If $G_j \neq 0$ average all gravity line properties within each gridblock of the 3D domain to determine the final saturation distribution at t^{n+1} .
7. Return to step 1.

A key reason for large speedup factors in the streamline method is the fact that Δt , the time step size for a convective step that includes a pressure solve, can be orders of magnitude larger than the time step size in conventional simulators. This is a result of

eliminating the global CFL condition by decoupling fluid movement from the underlying grid.

An important consideration in field simulations however, is that the time step size in the streamline method can be limited by the need to honor changing well conditions. Thus we expect speedup factors to be smaller for simulation that must honor historical production information since the pressure field is recomputed every time the well conditions change, as opposed to using the method in a forecast mode.

CHAPTER III

STREAMLINE TRACING AND TIME OF FLIGHT CALCULATION

The tracing of the streamlines is of major importance since their paths determine the total flow pattern and the time of flight along them serves as the primary variable to determine the one-dimensional saturation profile. Finally, the flow rate associated with every streamline determines the weights for the average block saturation used in the pressure equation.

The pressure field from which the streamlines are traced is assumed to be obtained from a control volume discretization of conservation equation, such as the one used in finite difference methods. This type of formulation is based on the discretization of the total mass conservation in a control volume that surrounds a pressure node (center of a finite difference cell). Flux on each control volume face in a cell is approximated by a linear combination of the cell pressure nodes. In the case of a Cartesian grid, the control volume is a rectangle. For a Corner Point Geometry grid, the control volume is a hexahedron.

Tracing of streamlines currently rests on Pollock's bilinear interpolation, which in turn makes the fundamental assumption that there is a single velocity per cell face. However, his formulation works only for Cartesian grids, but very few real reservoirs models use this framework anymore. Corner-point geometry gridding is unique and allows extremely complex geometries to be constructed to give a faithful representation of the reservoir geology, it is especially useful for highly faulted reservoirs, where the grid may be distorted areally to fit along fault lines and displaced vertically to model complex scissor faults. In other to ease the transition to corner point cells, Pollock's equations can be rewritten in dimensionless terms allowing the streamlines to be traced within a unit-transformed space. Mapping the streamlines to real space can be achieved

using an isoparametric back-transformation and finally the time of flight calculation can be accomplished introducing a time-like variable in the unit space.

3.1. Cartesian Grid Tracing

The breakthrough work for tracing streamlines efficiently in 3D was that of Pollock¹¹ (1988). Pollock's method is simple, analytical, and is formulated in terms of time-of-flight (TOF). To apply Pollock's method to any cell, the total flux in and out of each boundary is calculated using Darcy's Law. With the flux known, the algorithm centers on determining the exit point of a streamline and the time to exit, given any entry point assuming a piece-wise linear approximation of the velocity field in each coordinate direction.

3.1.1. Pollock's Interpolation

The average linear velocity component across each face in a particular cell (Fig. 3.1) is obtained by dividing the volume flow rate across the face by the cross sectional area of the face and the porosity of the material in the cell (Eq. 3.1).

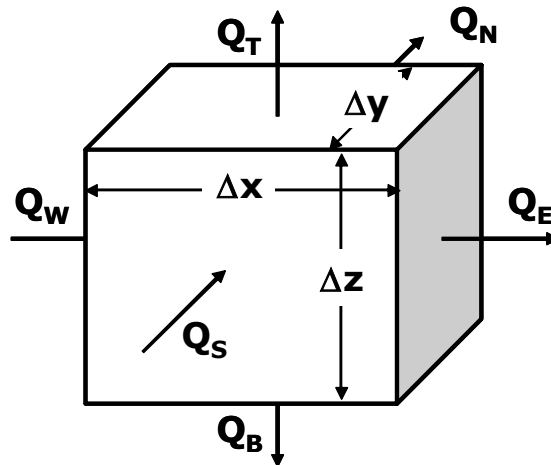


Fig. 3.1 Finite Difference Cell Showing xyz Definitions

$$\begin{aligned}
V_E &= \frac{Q_E}{\phi \cdot \Delta y \cdot \Delta z}, V_W = \frac{Q_W}{\phi \cdot \Delta y \cdot \Delta z} \\
V_N &= \frac{Q_N}{\phi \cdot \Delta x \cdot \Delta z}, V_S = \frac{Q_S}{\phi \cdot \Delta x \cdot \Delta z} \\
V_T &= \frac{Q_T}{\phi \cdot \Delta x \cdot \Delta y}, V_B = \frac{Q_B}{\phi \cdot \Delta x \cdot \Delta y}
\end{aligned} \tag{3.1}$$

Where Q is a volume flow rate across a cell face, and Δx , Δy , and Δz are the dimensions of the cell in the respective coordinate directions. If flow to internal sources or sinks within the cell is specified as Q_s , the following mass balance equation can be written for the cell,

$$\frac{(\phi V_E - \phi V_W)}{\Delta x} + \frac{(\phi V_N - \phi V_S)}{\Delta y} + \frac{(\phi V_T - \phi V_B)}{\Delta z} = \frac{Q_s}{\Delta x \cdot \Delta y \cdot \Delta z} \tag{3.2}$$

The left hand side of this equation represents the net volume rate of outflow per unit volume of the cell, and the right hand side represents the net volume rate of production per unit volume due to internal sources and sinks.

In order to compute path lines, it is required to compute values of the principal components of the velocity vector at every point in the flow field based on the inter-cell flow rates from the finite difference model. Pollock's method uses a simple linear interpolation to compute the principal velocity components at points within a cell, the principal velocity components can be expressed in the form,

$$\begin{aligned}
V_x &= A_x(x - x_1) + V_W \\
V_y &= A_y(y - y_1) + V_N \\
V_z &= A_z(z - z_1) + V_T
\end{aligned} \tag{3.3}$$

Where A_x , A_y , and A_z are constants that correspond to the components of the velocity gradient within the cell and are given by,

$$\begin{aligned}
A_x &= \frac{(V_W - V_E)}{\Delta x} \\
A_y &= \frac{(V_N - V_S)}{\Delta y} \\
A_z &= \frac{(V_T - V_B)}{\Delta z}
\end{aligned}
\tag{3.4}$$

Linear interpolation of the six cell face velocity components results in a velocity vector field within the cell that automatically satisfies the differential conservation of mass equation at every point inside the cell. This is correct only if it is assumed that the internal sources or sinks are considered to be uniformly distributed within the cell.

The fact that the velocity vector field within each cell satisfies the differential mass balance equation assures that path lines will distribute liquid throughout the flow field in a way that is consistent with the overall movement of liquid in the system as indicated by the solution of the finite-difference flow equations.

In order to find the position of the particle, its movement through a three-dimensional finite-difference cell must be considered. Let's start with the rate of change in the particle's *x*-component of velocity as it moves through the cell, this is given by,

$$\left(\frac{dV_x}{dt} \right)_p = \left(\frac{dV_x}{dx} \right) \left(\frac{dx}{dt} \right)_p
\tag{3.5}$$

The subscript, *p*, is used to indicate that a term is evaluated at the location of the particle denoted by the *x*-*y*-*z* coordinates (*x_p*, *y_p*, *z_p*). The term $(dx/dt)_p$ is the time rate of change of the *x*-location of the particle. By definition,

$$V_{xp} = \left(\frac{dx}{dt} \right)_p
\tag{3.6}$$

Where V_{xp} is the particle's x -velocity-component. Differentiating the principal velocity components (Eq. 3.3) with respect to x yields the additional relation,

$$\left(\frac{dV_x}{dx}\right) = A_x \quad (3.7)$$

Substituting equations (3.6) and (3.7) into equation (3.5.) gives,

$$\left(\frac{dV_x}{dt}\right)_p = A_x V_{xp} \quad (3.8)$$

This equation can be rearranged to the form,

$$\frac{1}{V_{xp}}(dV_x)_p = A_x dt \quad (3.9)$$

This equation can be integrated and evaluated for times t_1 and t_2 leading to,

$$\ln\left(\frac{V_{xp}(t_2)}{V_{xp}(t_1)}\right) = A_x \Delta t \quad (3.10)$$

By taking the exponential of each side of the equation and substituting the velocity components in equation (3.3) the x -position of the particle can be evaluated using the next expression,

$$x_p(t_2) = x_1 + \frac{1}{A_x} [V_{xp}(t_1)e^{A_x \Delta t} - V_{x1}] \quad (3.11)$$

The velocity components of the particle at time t_1 are known functions of the particle's coordinates; consequently, the coordinates of the particle at any future time t_2 can be computed directly from equation (3.11).

For steady-state flow, the direct integration method described above can be imbedded in a simple algorithm that allows a particle's exit point from a cell to be determined directly given any known starting location within the cell. To illustrate the method, consider the two-dimensional example shown in Fig. 3.2 cell (i,j) is in the x - y plane and contains a particle, p , located at (x_p, y_p) at time t_p .

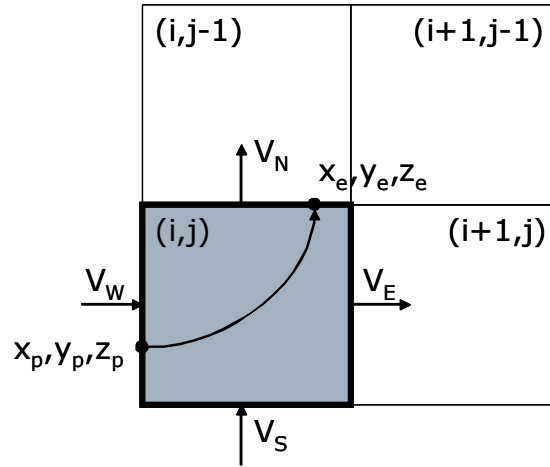


Fig. 3.2 Schematic Showing the Computation of Exit Point and Travel Time in 2D

The first step is to determine the face across which the particle leaves cell (i,j) . For the present example, this is accomplished by noting that the velocity components at the four faces require that the particle leave the cell through either the north or the east face. Consider the x -direction first. From equation (3.3) V_{xp} can be calculated at the point (x_p, y_p) , since we also know V_x equals V_E at the east face, equation (3.10) can be used to determine the time that would be required for the particle to reach the east face. An analogous calculation can be made to determine the time required for the particle to reach the north face. If Δt_x is less than Δt_y , the particle will leave the cell across the east face and enter cell $(i+1,j)$. Conversely, if Δt_y is less than Δt_x , the particle will leave the cell across the north face and enter cell $(i,j-1)$.

The length of time required for the particle to travel from point (x_p, y_p) to a boundary face of cell (i, j) is taken to be the smaller of Δt_x and Δt_y , and is denoted as Δt_e . The value Δt_e is then used in equation (3.11) to determine the exit coordinates (x_e, y_e) for the particle as it leaves cell (i, j) ,

$$\begin{aligned} x_e &= x_1 + \frac{1}{A_x} \left[V_{xp}(t_p) e^{A_x \Delta t_e} - V_{x1} \right] \\ y_e &= y_1 + \frac{1}{A_y} \left[V_{yp}(t_p) e^{A_y \Delta t_e} - V_{y1} \right] \end{aligned} \quad (3.12)$$

The time at which the particle leaves the cell is given by: $t_e = t_p + \Delta t_e$. This sequence of calculations is repeated, cell by cell, until the particle reaches a discharge point. The approach can be generalized to three dimensions in a straight forward way by performing all of the calculations for the *z-direction* in addition to the *x-* and *y-directions*.

3.2. Corner Point Geometry Grid Tracing

Pollock's equations are derived assuming orthogonal grid blocks, but very few real reservoirs models use such a strict Cartesian framework anymore. Using an isoparametric transformation, it is possible to transform Corner Point Geometry grids (CPG) into unit cubes, apply Pollock's method, and then transform the exit coordinate back to physical space. The solution we describe follows the development of Cordes and Kinzelbach²² (CK), which is the most commonly used extension to Pollock's algorithm for rectangular cells.

Pollock's algorithm can be rephrased in a way to ease the transition from Cartesian grids to CPG grids. The first step is to re-write Pollock's equation in dimensionless variables using the fractional distances through all three coordinate directions.

$$\begin{aligned}
\alpha &= x/DX \\
\beta &= y/DY \\
\gamma &= z/DZ
\end{aligned}
\tag{3.13}$$

It is also necessary to convert the directional interstitial velocities into volumetric fluxes. These fluxes each vary linearly across the cell such that a simple linear interpolation can be applied to compute the principal velocity components at points within a cell.

$$\begin{aligned}
Q_x &= u_x \cdot DY \cdot DZ \\
Q_y &= u_y \cdot DX \cdot DZ \\
Q_z &= u_z \cdot DX \cdot DY
\end{aligned}
\tag{3.14}$$

These set of equations can be re-written using the rate of change in the particle's velocity components as it moves through the cell,

$$\begin{aligned}
\phi \cdot DX \cdot \frac{d\alpha}{d\tau} &= \frac{Q_x(\alpha)}{DY \cdot DZ} \\
\phi \cdot DY \cdot \frac{d\beta}{d\tau} &= \frac{Q_y(\beta)}{DX \cdot DZ} \\
\phi \cdot DZ \cdot \frac{d\gamma}{d\tau} &= \frac{Q_z(\gamma)}{DX \cdot DY}
\end{aligned}
\tag{3.15}$$

Combining these equations the following relationships are obtained

$$\frac{d\tau}{\phi \cdot DX \cdot DY \cdot DZ} = \frac{d\alpha}{Q_x(\alpha)} = \frac{d\beta}{Q_y(\beta)} = \frac{d\gamma}{Q_z(\gamma)}
\tag{3.16}$$

At this stage no new results have been provided, Pollock's model has been rescaled in terms of dimensionless distances and volumetric fluxes. Using tri-linear interpolants in the form $x(\alpha, \beta, \gamma)$, $y(\alpha, \beta, \gamma)$, $z(\alpha, \beta, \gamma)$ Pollock's algorithm can be extended to deal with the streamline tracing in CPG grids. The only difference will be

using flux instead of velocity as a coordinate parameter. We can integrate the last of these equations to determine the trajectories traced out by the streamlines in three dimensions.

Now let's describe the corner point cell geometry, Figure 3.3. The cell is defined as a tri-linear mapping from the unit cube into physical space. Each point in initial physical space (right) is considered as the back-transform of a point in unit space by conserving its barycentric coordinates.

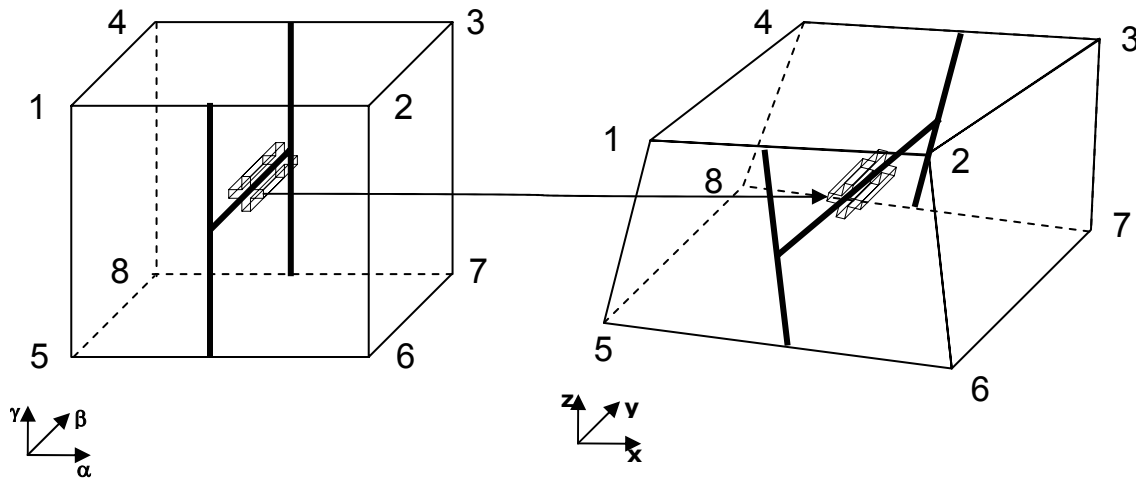


Fig. 3.3 Isoparametric Transformation

The Jacobian may be used to determine the cell volume, since it is the ratio of physical volume to unit volume.

$$J(\alpha, \beta, \gamma) = \left\| \frac{\partial(x, y, z)}{\partial(\alpha, \beta, \gamma)} \right\| \quad (3.17)$$

$$Volume = \int_{(0,0,0)}^{(1,1,1)} J(\alpha, \beta, \gamma) d\alpha d\beta d\gamma \quad (3.18)$$

In three dimensions the Jacobian is less than quadratic in $\alpha\beta\gamma$. In two dimensions it is less than linear in $\alpha\beta$. For a rectangular cell, the Jacobian is simply the cell volume.

To determine the time of flight a velocity model is required within the cell. For a corner point cell in three dimensions, the following velocity model is imposed.

$$\begin{aligned}\phi \frac{d\alpha}{d\tau} &= \frac{Q_x(\alpha)}{J(\alpha, \beta, \gamma)} \\ \phi \frac{d\beta}{d\tau} &= \frac{Q_y(\beta)}{J(\alpha, \beta, \gamma)} \\ \phi \frac{d\gamma}{d\tau} &= \frac{Q_z(\gamma)}{J(\alpha, \beta, \gamma)}\end{aligned}\tag{3.19}$$

In these set of equations each volumetric flux is linearly interpolated in the appropriate direction, similar to Pollock's velocity interpolation

$$Q_j(\alpha_j) = A_j + C_j \cdot \alpha_j, \quad j = 1, 2, 3\tag{3.20}$$

The Jacobian is essentially a cross-sectional area times a physical distance, and so the right hand side in each equation is effectively a Darcy velocity in one direction, scaled by the cell length in that direction. The physical velocity is obtained from these dimensionless velocities by application of the chain rule. In component form:

$$u_i = \frac{1}{J(\alpha, \beta, \gamma)} \cdot \sum_{j=1}^3 \frac{\partial x_i}{\partial \alpha_j} \cdot Q_j(\alpha_j), \quad i = 1, 2, 3\tag{3.21}$$

In the form of Equation (3.19), the (α, β, γ) trajectories are much more difficult to integrate than for rectangular cells, as all three parameters are coupled through the Jacobian. The CK solution to Equation (3.19) selects one of the three integrals and then substitutes for $\beta(\alpha)$ and $\gamma(\alpha)$.

However, a significantly simpler development is possible with the introduction of a parameter T that increases along a trajectory, and which acts as a time-like variable,

$$dT = \frac{1}{\phi} \frac{d\tau}{J(\alpha, \beta, \gamma)} = \frac{d\alpha}{Q_1(\alpha)} = \frac{d\beta}{Q_2(\beta)} = \frac{d\gamma}{Q_3(\gamma)} \quad (3.22)$$

For constant scaling factors, these equations for $\alpha(T)$, $\beta(T)$, $\gamma(T)$ are identical to the Pollock equations in a three dimensional rectangular cell. After obtaining their solution, we can determine τ from the remaining equation:

$$\tau = \phi \int_0^T J(\alpha(T), \beta(T), \gamma(T)) dT \quad (3.23)$$

In this equation α , β , and γ are all known functions of T . Each parameter will depend upon T through constructions of the form $((e^{cT} - 1)/c)$, and the Jacobian is a polynomial in α , β , and γ . The resulting integrand is a sum of exponentials and constants, which can be integrated numerically using a quadrature approach.

The following sections will address how the mathematical formulation was implemented in a FORTRAN code and the overall procedure will be discussed in detail. The isoparametric transformation will be presented for streamline tracing purposes as well as the Jacobian of the dimensionless transformed coordinates for time of flight calculation.

3.2.1. Unit Space

3.2.1.1. Pseudo-Time of Flight

Consider the unit cube shown in Fig. 3.4, this particular example will be used to explain the tracing methodology. The first step is to determine the face across which the particle leaves cell (i, j, k) . For the present example, this is accomplished by noting that the flux components at the six faces require that the particle leave the cell through any face but the west one.

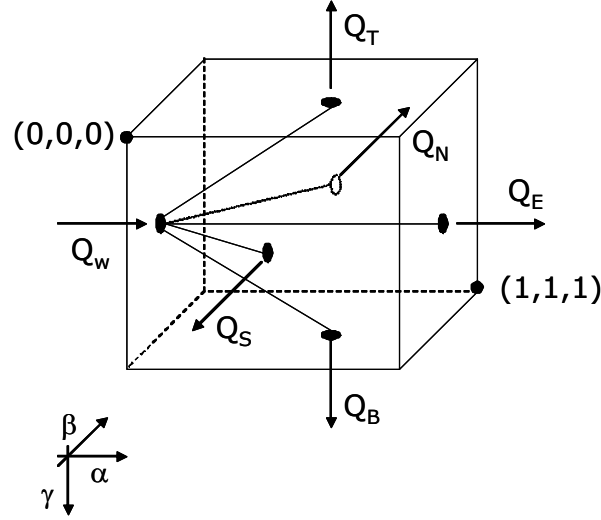


Fig. 3.4 Schematic Showing the Pseudo-Time of Flight in a Unit Cube

To find the face where the particle is leaving the cube, equation (3.22) can be used to determine the time that the particle would require to reach each face. Consider the x -direction,

$$\int_0^{T_E} dT = \int_{\alpha_0}^{\alpha} \frac{d\alpha}{Q_1(\alpha)} \quad (3.24)$$

The volumetric flux can be replaced by its linear interpolate in the x -direction using equation (3.20). The time to reach the east face will be,

$$T_E = \int_{\alpha_0}^{\alpha} \frac{d\alpha}{A_1 + C_1 \cdot \alpha} = \frac{1}{c_1} \ln \left[\frac{a_1 + \alpha \cdot c_1}{a_1 + \alpha_0 \cdot c_1} \right] \quad (3.25)$$

Now, since the particle might leave the cell through the east face, its α coordinate is already known, reducing the expression to,

$$T_E^{\alpha=1} = \frac{1}{c_1} \ln \left[\frac{a_1 + c_1}{a_1 + \alpha_0 \cdot c_1} \right] \quad (3.26)$$

An analogous integration can lead to the time for the particle to reach all the other faces,

$$\begin{aligned}
 T_T^{\gamma=0} &= \frac{1}{c_3} \ln \left[\frac{a_3}{a_3 + \gamma_0 \cdot c_3} \right] \\
 T_B^{\gamma=1} &= \frac{1}{c_3} \ln \left[\frac{a_3 + c_3}{a_3 + \gamma_0 \cdot c_3} \right] \\
 T_N^{\beta=0} &= \frac{1}{c_2} \ln \left[\frac{a_2}{a_2 + \beta_0 \cdot c_2} \right] \\
 T_S^{\beta=1} &= \frac{1}{c_2} \ln \left[\frac{a_2 + c_2}{a_2 + \beta_0 \cdot c_2} \right]
 \end{aligned} \tag{3.27}$$

Therefore the length of time required for the particle to go from any given entry point to a boundary face will be the minimum non-negative pseudo-time of flight denoted by T . For this example let's assume that the particle is leaving the top face, the next step will be to find the exit coordinate of the particle.

3.2.1.2. Unit Cube Streamline Tracing

Following the previous example (Fig. 3.5), once the pseudo-time of flight T is known, the exit coordinate of the particle is easily calculated using the general solution of equation (3.22) in all three directions and solving for each unit coordinate. The set of equations to use are given as,

$$\begin{aligned}
 \alpha_e &= \alpha_0 + (a_1 + \alpha_0 c_1) \left(\frac{e^{c_1 T} - 1}{c_1} \right) \\
 \beta_e &= \beta_0 + (a_2 + \beta_0 c_2) \left(\frac{e^{c_2 T} - 1}{c_2} \right) \\
 \gamma_e &= \gamma_0 + (a_3 + \gamma_0 c_3) \left(\frac{e^{c_3 T} - 1}{c_3} \right)
 \end{aligned} \tag{3.28}$$

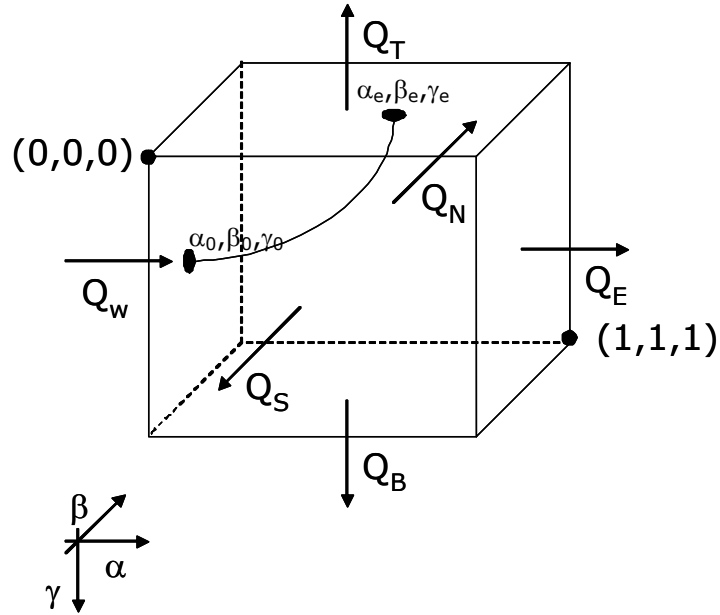


Fig. 3.5 Schematic Showing the Exit Coordinates Computation in a Unit Cube

So far Pollock's technique has been extended to sketch the streamlines in the unit cube using linear flux interpolants instead of velocity. The pseudo-time of flight T , has been introduced as a time-act variable and it is also used as a unit-cube coordinate tracer. However, half the problem remains to be solved: How the streamlines can be mapped back to the real space? How this pseudo-time of flight is transformed to the real space? Both questions will be addressed in the next section.

3.2.2. Real Space

3.2.2.1. Isoparametric Transform

When using CPG grids, each gridblock is a solid defined by the location of its eight corners (Fig. 3.6). Each grid cell can be mapped to a unit cube by an isoparametric set of equations as follows,

$$\begin{aligned}
x &= p_{1,x}\alpha + p_{2,x}\beta + p_{3,x}\gamma + p_{4,x}\alpha\beta + p_{5,x}\beta\gamma + p_{6,x}\alpha\gamma + p_{7,x}\alpha\beta\gamma + p_{8,x} \\
y &= p_{1,y}\alpha + p_{2,y}\beta + p_{3,y}\gamma + p_{4,y}\alpha\beta + p_{5,y}\beta\gamma + p_{6,y}\alpha\gamma + p_{7,y}\alpha\beta\gamma + p_{8,y} \\
z &= p_{1,z}\alpha + p_{2,z}\beta + p_{3,z}\gamma + p_{4,z}\alpha\beta + p_{5,z}\beta\gamma + p_{6,z}\alpha\gamma + p_{7,z}\alpha\beta\gamma + p_{8,z}
\end{aligned} \tag{3.29}$$

Where

$$\begin{aligned}
p_{1,x} &= x_2 - x_1 & p_{2,x} &= x_4 - x_1 & p_{3,x} &= x_5 - x_1 \\
p_{4,x} &= x_1 + x_3 - x_2 - x_4 & p_{5,x} &= x_1 + x_8 - x_4 - x_5 & p_{6,x} &= x_1 + x_6 - x_2 - x_5 \\
p_{7,x} &= x_2 + x_4 + x_5 + x_7 - x_1 - x_3 - x_6 - x_8 & p_{8,x} &\approx x_1
\end{aligned} \tag{3.30}$$

This works similarly for $p_{1,y}, \dots, p_{8,y}$ in terms of the y 's, and $p_{1,z}, \dots, p_{8,z}$ in terms of the z 's. It can be seen that $\alpha=0$ corresponds to face 1-4-8-5, $\alpha=1$ to face 2-3-7-6, $\beta=0$ to face 1-2-6-5, $\beta=1$ to face 4-3-7-8, etc.

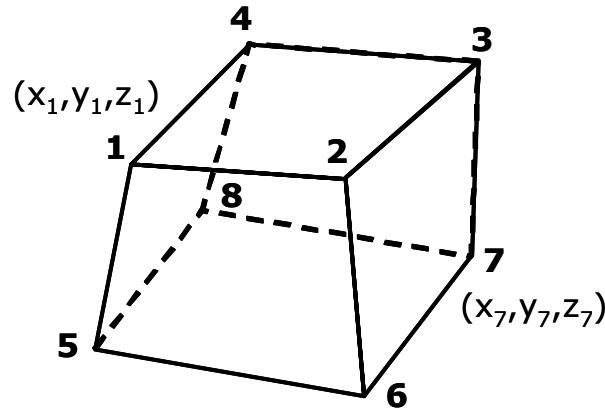


Fig. 3.6 The Eight Corners of a Corner Point Cell

These sets of equations (3.29 and 3.30) are used to map the unit cube to real space. However, when going from real to unit space, a non-linear system must be solved using any iterative procedure such as Newton's method. Fig 3.7 shows the example from the previous sections and illustrates the concept of isoparametric transformation.

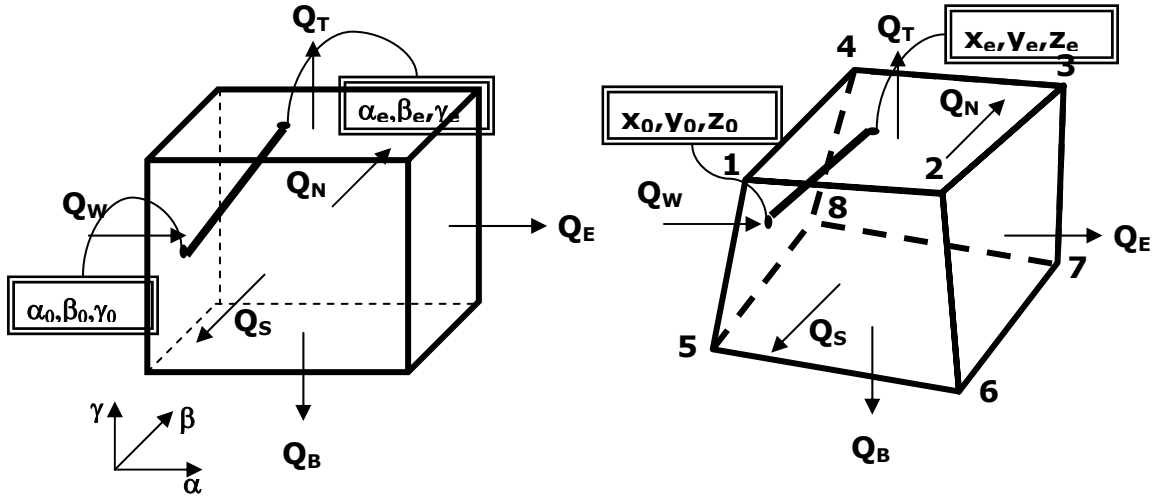


Fig. 3.7 Isoparametric Transformation Unit to Real Space

3.2.2.2. Time of Flight Calculation

It has already been mentioned how the time of flight in real space can be obtained introducing the pseudo-time of flight in the velocity model. To carry out this integral the Jacobian must be expressed in terms of the pseudo-time of Flight. Going back to equation (3.23) the Jacobian of the real coordinates is defined as,

$$J(x, y, z) = \begin{vmatrix} \frac{\partial x}{\partial \alpha} & \frac{\partial x}{\partial \beta} & \frac{\partial x}{\partial \gamma} \\ \frac{\partial y}{\partial \alpha} & \frac{\partial y}{\partial \beta} & \frac{\partial y}{\partial \gamma} \\ \frac{\partial z}{\partial \alpha} & \frac{\partial z}{\partial \beta} & \frac{\partial z}{\partial \gamma} \end{vmatrix} \quad (3.31)$$

The partial derivatives are obtained differentiating equations (3.29). For the x -coordinate the derivatives are given by,

$$\begin{aligned}
\frac{\partial x}{\partial \alpha} &= p_{1,x} + p_{4,x}\beta + p_{6,x}\gamma + p_{7,x}\beta\gamma \\
\frac{\partial x}{\partial \beta} &= p_{2,x} + p_{4,x}\alpha + p_{5,x}\gamma + p_{7,x}\alpha\gamma \\
\frac{\partial x}{\partial \gamma} &= p_{3,x} + p_{5,x}\beta + p_{6,x}\alpha + p_{7,x}\alpha\beta
\end{aligned} \tag{3.32}$$

In line with the previous definition $\tau = \phi \int_0^T J(\alpha(T), \beta(T), \gamma(T)) dT$, the Jacobian must be expressed in terms of the pseudo-time of flight T . Replacing equations (3.28)

$$\begin{aligned}
\frac{\partial x}{\partial \alpha} &= p_{1,x} + p_{4,x} \left[\beta_0 + (a_2 + \beta_0 c_2) \left(\frac{e^{c_2 T} - 1}{c_2} \right) \right] + p_{6,x} \left[\gamma_0 + (a_3 + \gamma_0 c_3) \left(\frac{e^{c_3 T} - 1}{c_3} \right) \right] + \\
& p_{7,x} \left[\beta_0 + (a_2 + \beta_0 c_2) \left(\frac{e^{c_2 T} - 1}{c_2} \right) \right] \left[\gamma_0 + (a_3 + \gamma_0 c_3) \left(\frac{e^{c_3 T} - 1}{c_3} \right) \right] \\
\frac{\partial x}{\partial \beta} &= p_{2,x} + p_{4,x} \left[\alpha_0 + (a_1 + \alpha_0 c_1) \left(\frac{e^{c_1 T} - 1}{c_1} \right) \right] + p_{5,x} \left[\gamma_0 + (a_3 + \gamma_0 c_3) \left(\frac{e^{c_3 T} - 1}{c_3} \right) \right] + \\
& p_{7,x} \left[\alpha_0 + (a_1 + \alpha_0 c_1) \left(\frac{e^{c_1 T} - 1}{c_1} \right) \right] \left[\gamma_0 + (a_3 + \gamma_0 c_3) \left(\frac{e^{c_3 T} - 1}{c_3} \right) \right] \\
\frac{\partial x}{\partial \gamma} &= p_{3,x} + p_{5,x} \left[\beta_0 + (a_2 + \beta_0 c_2) \left(\frac{e^{c_2 T} - 1}{c_2} \right) \right] + p_{6,x} \left[\alpha_0 + (a_1 + \alpha_0 c_1) \left(\frac{e^{c_1 T} - 1}{c_1} \right) \right] + \\
& p_{7,x} \left[\alpha_0 + (a_1 + \alpha_0 c_1) \left(\frac{e^{c_1 T} - 1}{c_1} \right) \right] \left[\beta_0 + (a_2 + \beta_0 c_2) \left(\frac{e^{c_2 T} - 1}{c_2} \right) \right]
\end{aligned} \tag{3.33}$$

Similarly for $\partial y/\partial \alpha, \partial y/\partial \beta, \partial y/\partial \gamma$, and so forth. The Jacobian will now be defined as,

$$J(x, y, z) = \begin{vmatrix} p_{1,x}\alpha + p_{4,x}\beta + p_{6,x}\gamma + p_{7,x}\beta\gamma & p_{2,x}\beta + p_{4,x}\beta + p_{5,x}\beta + p_{7,x}\alpha\gamma & p_{3,x}\gamma + p_{5,x}\beta + p_{6,x}\alpha + p_{7,x}\alpha\beta \\ p_{1,y}\alpha + p_{4,y}\beta + p_{6,y}\gamma + p_{7,y}\beta\gamma & p_{2,y}\beta + p_{4,y}\beta + p_{5,y}\beta + p_{7,y}\alpha\gamma & p_{3,y}\gamma + p_{5,y}\beta + p_{6,y}\alpha + p_{7,y}\alpha\beta \\ p_{1,z}\alpha + p_{4,z}\beta + p_{6,z}\gamma + p_{7,z}\beta\gamma & p_{2,z}\beta + p_{4,z}\beta + p_{5,z}\beta + p_{7,z}\alpha\gamma & p_{3,z}\gamma + p_{5,z}\beta + p_{6,z}\alpha + p_{7,z}\alpha\beta \end{vmatrix}$$

Where

$$\begin{aligned}
\alpha &= \alpha_0 + (a_1 + \alpha_0 c_1) \left(\frac{e^{c_1 T} - 1}{c_1} \right) \\
\beta &= \beta_0 + (a_2 + \beta_0 c_2) \left(\frac{e^{c_2 T} - 1}{c_2} \right) \\
\gamma &= \gamma_0 + (a_3 + \gamma_0 c_3) \left(\frac{e^{c_3 T} - 1}{c_3} \right)
\end{aligned} \tag{3.34}$$

As mentioned before, in the final equation α , β , and γ are all known functions of T . Each parameter depends upon T through constructions of the form $((e^{cT} - 1)/c)$, and the Jacobian is a polynomial in α , β , and γ . The resulting integrand is a sum of exponentials and constants. This integral may be evaluated quite accurately by the N-point summation in the form of:

$$\int f(x) dx = \sum_{k=1}^N w_k f(x_k) \tag{3.35}$$

The w_k are weights and the x_k are Gauss points, or quadrature points, at which the function $f(x)$ is evaluated. For $N=1, 2$ or 3 , the weights and Gauss points are given by:

For $N = 1$

$$w_1 = 1$$

$$x_1 = \frac{1}{2}$$

For $N = 2$

$$w_1 = w_2 = \frac{1}{2}$$

$$x_1 = \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), x_2 = \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right)$$

For $N = 3$

$$w_1 = w_3 = \frac{5}{18}, w_2 = \frac{4}{9}$$

$$x_1 = \frac{1}{2} \left(1 - \sqrt{\frac{3}{5}} \right), x_2 = \frac{1}{2}, x_3 = \frac{1}{2} \left(1 + \sqrt{\frac{3}{5}} \right)$$

If $f(x)$ is a polynomial of degree d , then Gaussian quadrature is exact if $d \leq 2N-1$. Thus 2-point quadrature is exact for a cubic, and 3-point quadrature is exact for a fifth-degree polynomial. To define the number of points, let's check the triple integral to find the volume of the cell, this integral may be approximated by triple summations,

$$Volume = \int_{(0,0,0)}^{(1,1,1)} J(\alpha, \beta, \gamma) d\alpha d\beta d\gamma = \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \sum_{k=1}^{N_k} w_i w_j w_k f(w_i w_j w_k) \quad (3.36)$$

Where N_i , N_j , and N_k are, respectively, the number of quadrature points in the x , y and z directions. It turns out that the integrands within all the triple integrals for volumetric calculations are composite polynomials of low enough degree providing the exact integration. This is true no matter how arbitrarily the corner points are located.

CHAPTER IV

RESULTS AND APPLICATIONS

The overall procedure of the proposed methodology is summarized in Fig. 4.1.

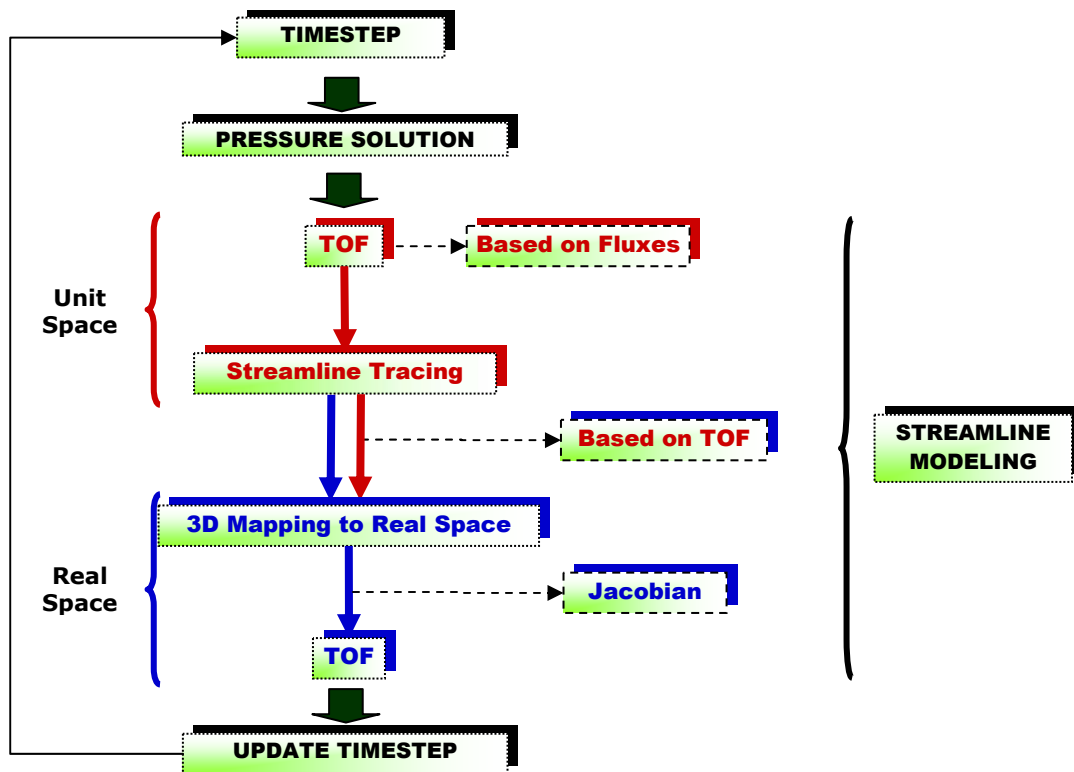


Fig. 4.1 Streamline Methodology for CPG Grids

For each timestep any finite difference tool can be used to solve for pressure and find the fluxes within each gridblock. Streamlines are traced from an arbitrary number of injection blocks to production blocks using the equations defined in Section 3.2. The streamline paths do not start at the center of an injection or production block since the velocity field cannot be approximated as piecewise linear within a gridblock containing a

point source. Rather, streamlines are launched from each gridblock face containing an injector or producer.

Streamlines are launched in proportion to the flux out of a face, such that more streamlines are launched from high flow rate injectors/producers, while fewer streamlines are launched from low flow rate injectors/producers. Thus more streamlines are traced through high flow velocity regions and fewer streamlines are traced through low flow velocity regions.

The flux across each injection block face is uniform, consistent with the underlying velocity field. Since the flux is uniform, streamlines are distributed on each face in a uniform manner. Not every gridblock in the domain will contain a streamline for a fixed total number of streamlines launched. A missed gridblock is assigned a streamline which is then traced backwards in the velocity field towards an injector and then the time of flight can be calculated and mapped to the gridblock.

In order to evaluate the tracing algorithm and TOF calculation, a FORTRAN code following the flowchart in Fig. 4.1 was developed to handle CPG grids. The forward tool used to solve for pressure and find the fluxes is the commercial simulator ECLIPSE. For each timestep a report is obtained with the gridblock fluxes, the streamlines are traced and the time of flight is calculated.

The following sections will discuss the results obtained and how the grid geometry affects the displacement calculation obtained with numerical simulation. First, we'll discuss how the formulation was validated, followed by several cases showing subsequent geologic complexities. For each case the following items are presented:

- Reservoir grid.
- Numerical/Analytical fluxes along flow directions.
- Numerical/Analytical streamlines paths.
- Numerical/Analytical time of flight.

4.1. Orthogonal Grids

The procedure validation was addressed using known solutions to a homogeneous quarter of a five spot and a homogeneous X-Z cross-section using Cartesian grids. Both models were constructed following Corner Point Geometry conventions. In turn to get the best approximation using numerical simulation, the simplicity of the models was fulfilled following the next assumptions:

- Incompressible Fluid.
- Homogeneous petro-physical properties (porosity, permeability, net to gross, water saturation).
- Producer constrained by pressure, injector constrained by rate.
- Same initial pressure for all gridblocks
- Zero vertical transmissibility among injector and producer cells to avoid cross-flow.
- High horizontal transmissibility among injector and producer cells to make sure the fluid is uniformly distributed towards the flow direction.

4.1.1. Homogeneous Quarter of a Five Spot

A 21x21x1 Cartesian grid was built defining the eight corners of each gridblock. Fig 4.2 shows the orthogonal grid, the streamline paths using the proposed CPG tracing method and the numerical X and Y flow for each cell. Fig 4.3 shows the time of flight.

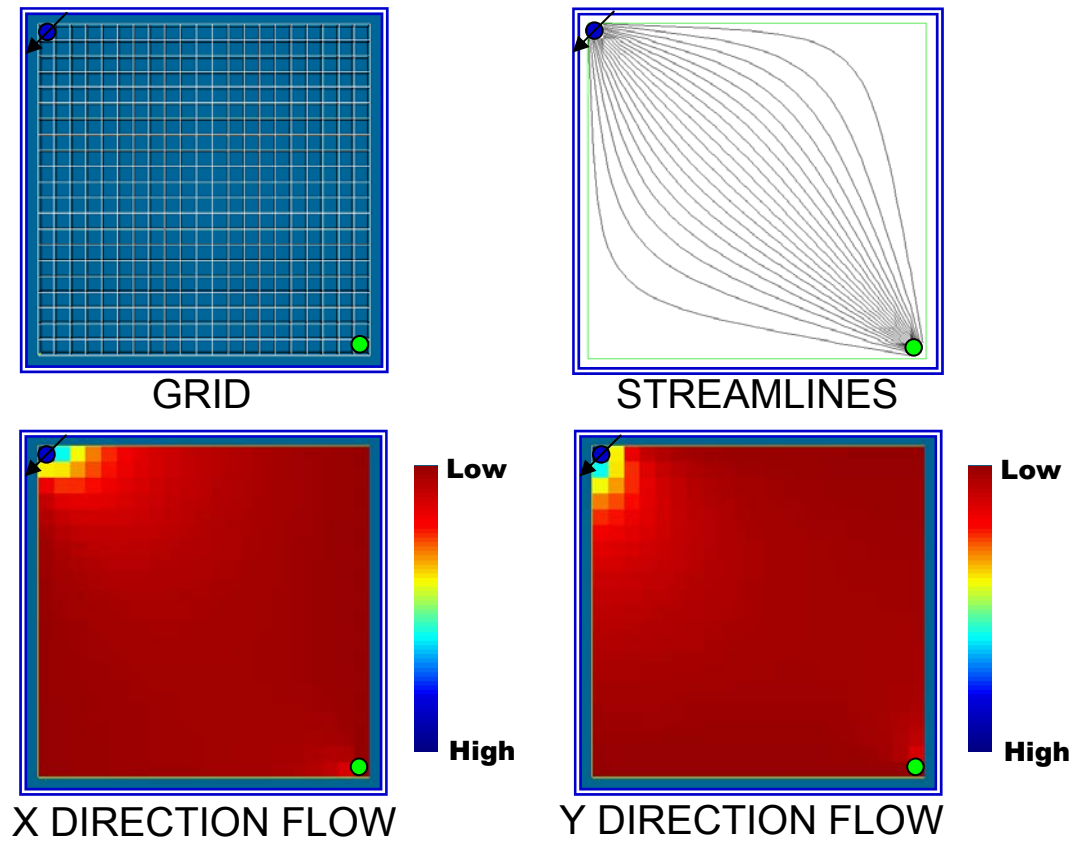


Fig. 4.2 Quarter of a Five-Spot Orthogonal Grid

The streamline paths are in good agreement with the known solutions using Pollock's tracing algorithm, so is the time of flight, it shows the expected symmetry and is congruent with the anticipated front position obtained with any forward modeling tool.

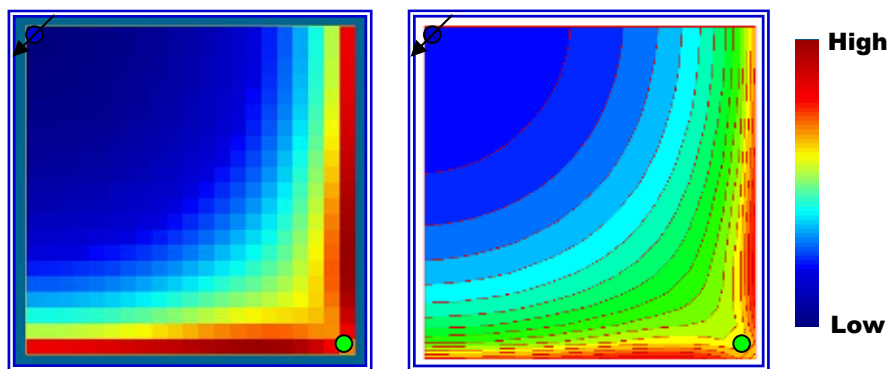


Fig. 4.3 Time of Flight - Quarter of a Five-Spot Orthogonal Grid

4.1.2. Homogeneous X-Z Cross-Section

The second model used to validate the methodology is a 21x1x21 X-Z cross-section. Fluid is injected from right to left. Similar to the quarter of a five spot, Fig 4.4 shows the grid, the streamlines paths and the numerical fluxes.

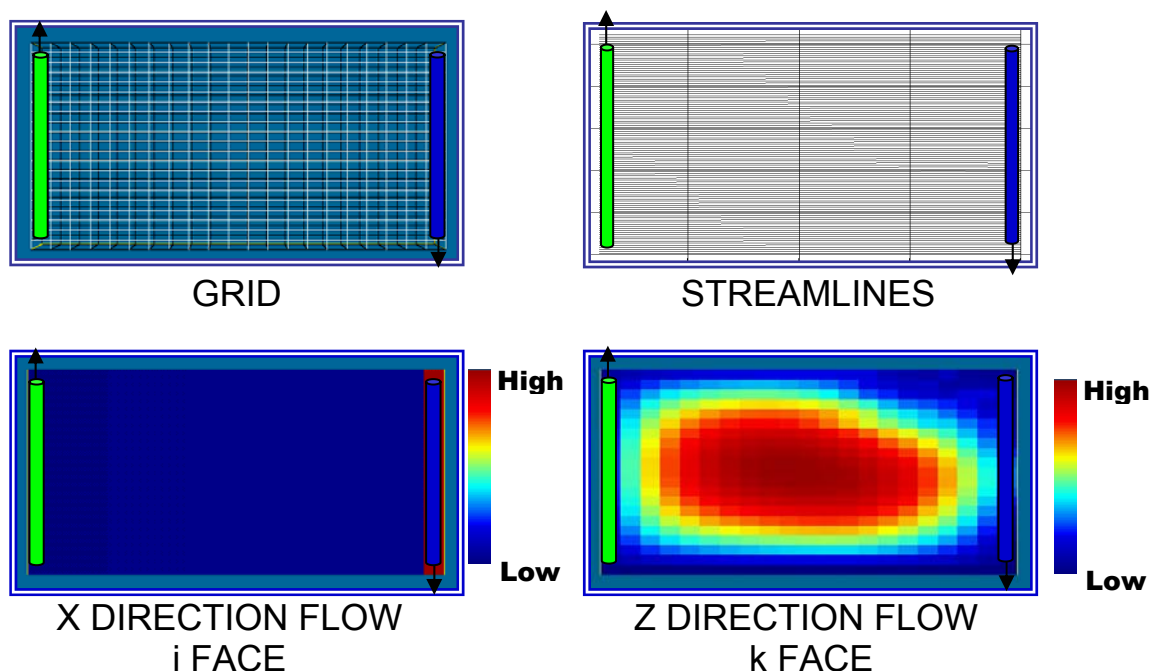


Fig. 4.4 Orthogonal X-Z Cross-Section Grid

Once more the outputs are in good concurrence with the known solutions. The streamlines are totally horizontal regardless of the vertical flux obtained. This vertical flux is very low and is negligible; its occurrence is probably due to round off errors in the numerical solution.

In the TOF picture the front propagation lacks of any distortion and is totally uniform; the time required to a particle to reach the producer is unvarying regardless of its starting position.

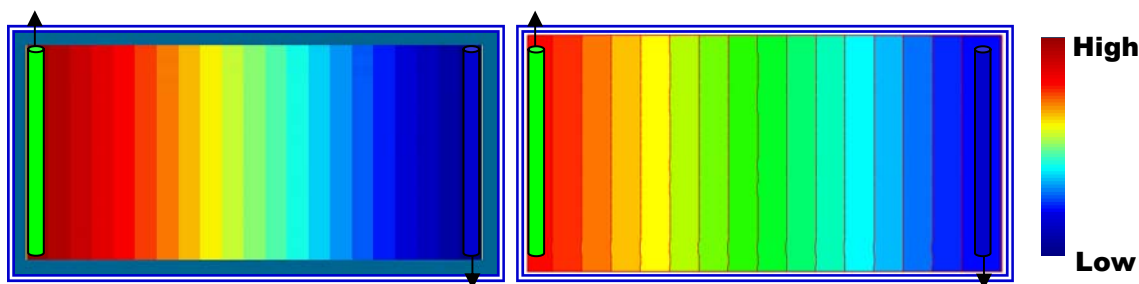


Fig. 4.5 Time of Flight - Orthogonal X-Z Cross-Section Grid

Up to this point it has been proved that the formulation is in conformity with simple-known orthogonal cases. However, what happens when the overall geometry is distorted? , how are the streamlines and TOF affected by these distortions? In the next section the same models will be presented adding non-orthogonal components to the grid.

4.2. Non-Orthogonal Grids

In the past, most reservoir simulation models were defined using rectangular Cartesian coordinate systems. Non-rectangular grid systems were introduced to reduce spatial truncation error, using orthogonal or near-orthogonal curvilinear coordinate systems. However, it is almost impossible to use orthogonal systems to model complicated flow boundaries frameworks caused by faults, irregular reservoir boundaries or channels. CPG grids were then introduced to provide more flexibility in grid-cell sizing and modeling of flow boundaries.

In CPG, the individual gridblocks may have different shapes and the direction of any coordinate axis may vary over the reservoir. Because of this freedom, ridiculous shapes and extreme non-orthogonality might be present; it is obvious that CPG should be used with extreme care.

The default grid discretization in ECLIPSE employs a five-point discretization in two dimensions or a seven point discretization in three dimensions. However when the

grid is non-orthogonal, or the flow is not aligned with the principal directions of the grid, then the solution accuracy may be compromised by this discretization. ECLIPSE uses a nine-point scheme to address this (the discretization stencil uses 9 cells in a 3*3 patch in two dimensions, or 27 cells in a 3*3*3 patch in three dimensions). This nine-point scheme was used for both the quarter of a five spot and the x-z cross-section, to account for non-orthogonality and will be presented in the two following sections.

4.2.1. Homogeneous Quarter of a Five Spot

Fig 4.6 shows the 21x21x1 near-orthogonal grid used to model the quarter of a five spot system. A slightly non-orthogonal component was added to the outer gridblocks, but the inner ones were kept orthogonal to each other. The streamlines are quite similar to the rectangular case, but there's a faintly asymmetry within the outer streamlines. This is mainly related to the difference through all the outer-cell fluxes.

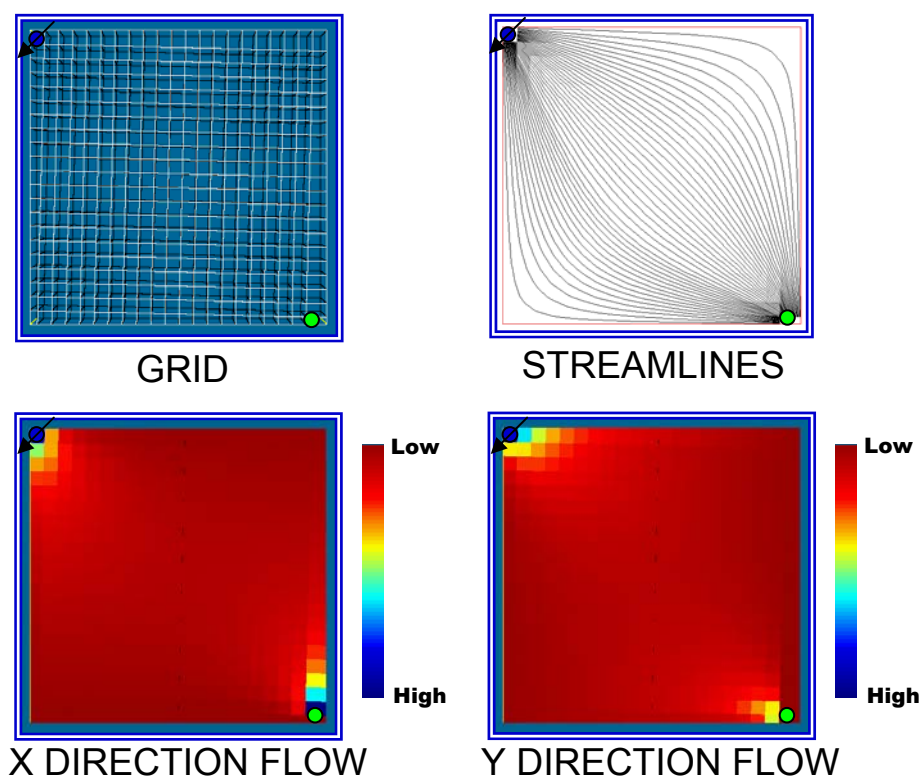


Fig. 4.6 Non-Orthogonal Quarter of a Five-Spot Grid

The TOF for this case is presented in Fig. 4.7, the results obtained are reliable enough compared to the known solution. However a contour map shows a discrepancy all along the outer non-orthogonal cells.

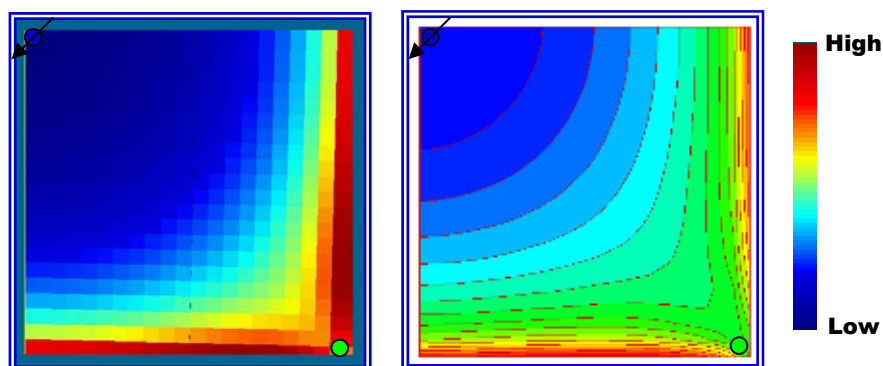


Fig. 4.7 Time of Flight - Quarter of a Five-Spot Non-Orthogonal Grid

4.2.2. Homogeneous X-Z Cross-Section

Fig 4.8 shows the numerical solution for the non-orthogonal 21x1x21 grid used to model the homogeneous X-Z cross-section. Similar to the orthogonal case there's a round off error in the x-direction flow along the k face. This flux component is adding a slightly vertical distortion in the streamlines paths.

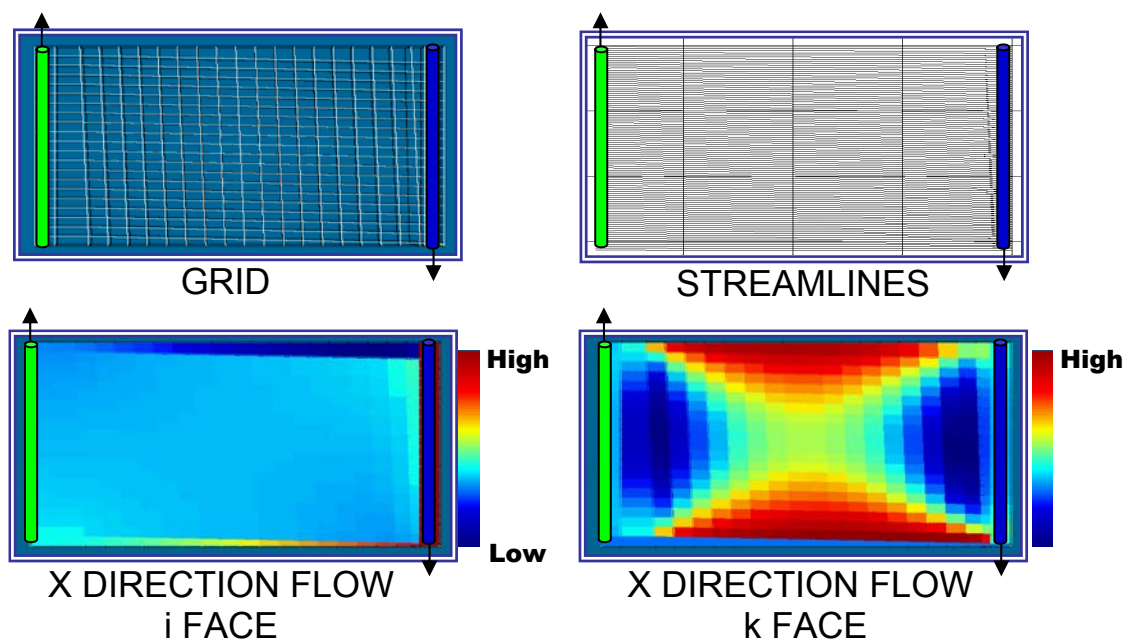


Fig. 4.8 Numerical Solutions - Non-Orthogonal X-Z Cross-Section Grid

It is important to mention that in order to reduce numerical non-orthogonal errors, the transmissibility between gridblocks is defined using nine-point schemes. However, the streamline non-horizontality is still present even using this expanded discretization stencil.

The front distortion due to the grid non-orthogonality can be appreciated both in the TOF field and contour map. The distortion severity is obviously related to the slanting degree of the coordinate lines between the injector and the producer. For the simple cases presented the numerical solution is fair enough, but dealing with complicated geologic geometries might result in unreal displacements and eventually misleading operational decisions.

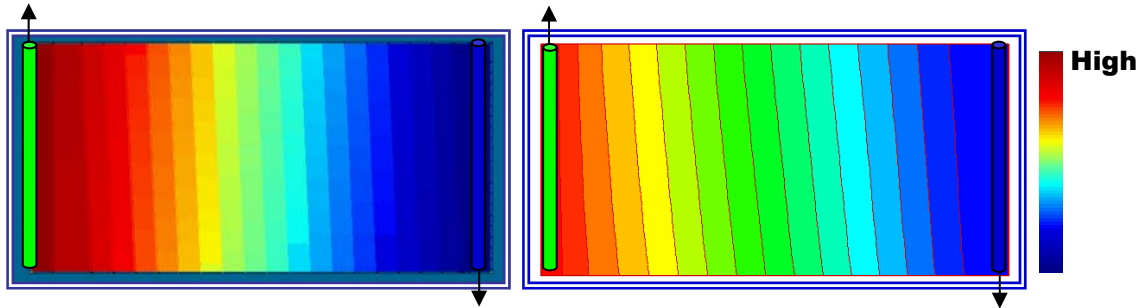


Fig. 4.9 Numerical Time of Flight - Non-Orthogonal X-Z Cross-Section Grid

4.3. Gridding Effects

So far, our new formulation has been applied to two well-known cases for orthogonal and non-orthogonal geometries. For the orthogonal cases we found both streamline paths and time of flight to be in excellent agreement with the known solutions. However for the non-orthogonal cases we found some non-symmetry and distortion effects on the solutions. Now that we have the ability to handle flexible grids and trace streamlines, we'll step into more complicated models showing common geologic features present in reservoir modeling.

In order to evaluate how accurate is the numerical solution, we used the potential and stream functions in all presented geometries to get the real solution. When the potential function is differentiated with respect to the flowing directions, the analytical form of all velocity components is obtained. Knowing the velocity components it is possible to find the total velocity and eventually the fluxes.

The first case used to compare the numerical and the stream-function derived solution is the non-orthogonal X-Z cross-section. Fig. 4.10 shows the results, as expected the streamlines are totally horizontal, even with the flux through the k face, which actually represents the x-direction flow.

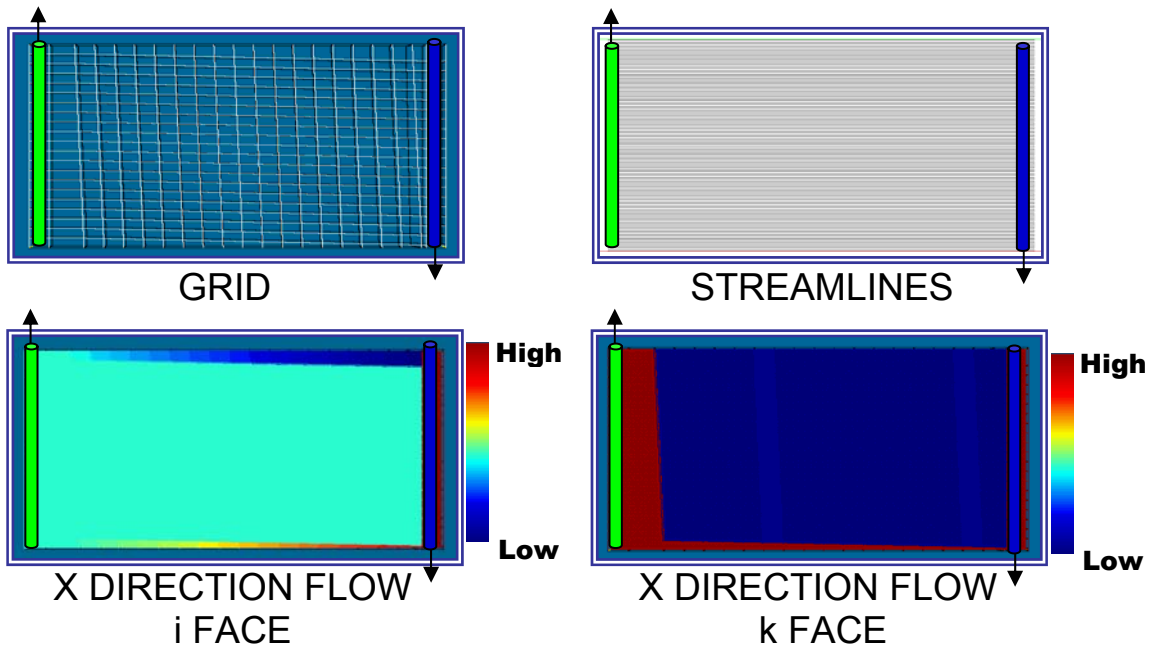


Fig. 4.10 Analytical Solutions - Non-Orthogonal X-Z Cross-Section Grid

The analytical time of flight is presented in Fig. 4.11; the front propagation is totally vertical, similar to the orthogonal case.

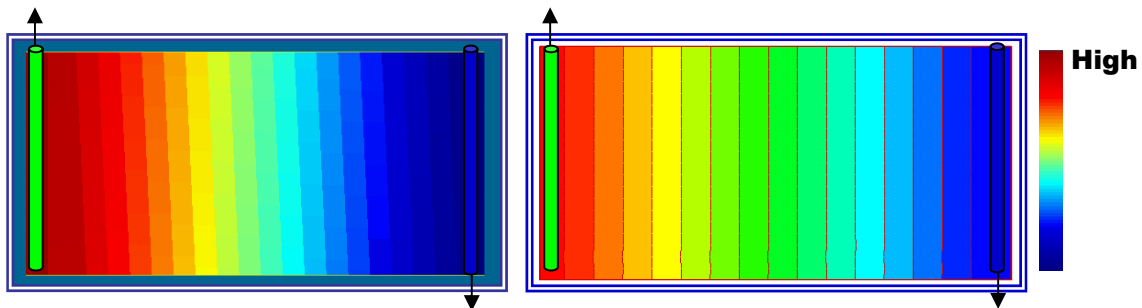


Fig. 4.11 Analytical Time of Flight - Non-Orthogonal X-Z Cross-Section Grid

After testing the stream-function over the cross-section, several cases were selected to compare the numerical approach to the analytical solution. The cross-section was selected as the base case and successive geological complexities were added to the grid.

4.3.1. Pinch-Outs

A pinch-out is a type of stratigraphic trap that occurs when a bed thins progressively in one direction. The overall framework may create a favorable geometry to trap hydrocarbons, particularly if the adjacent sealing rock is a source rock such as a shale. This type of model is of great interest in reservoir modeling due to its frequent appearance. We'll review two common pinch-outs models; the first one shows both an increase and decrease along the layer thickness and the second one shows missing layers related to erosive and diagenetic non-conformities.

4.3.1.1. Gradual Pinching

A pinch-out may or may not be accompanied by the increase or decrease in thickness of an adjacent unit. In some cases, the entire sedimentary section thins in a certain direction. This type of pinching-out was modeled using the grid in Fig. 4.12. The top and bottom units have a constant thickness acting as sealing members. These units are modeled as non-active cells in the numerical model following the close-boundary condition imposed by the pinch-out. The entire sedimentary section has two pinching-out directions, one is toward the upper-left sector and the other to the bottom-right.

The numerical solution shows a higher flux along the pinch-out compared to the non-deformed layer between both pinch-outs. This squeezing effect can be observed in the streamline paths. They're essentially horizontal but there's a small distortion aligned with the pinch-out direction.

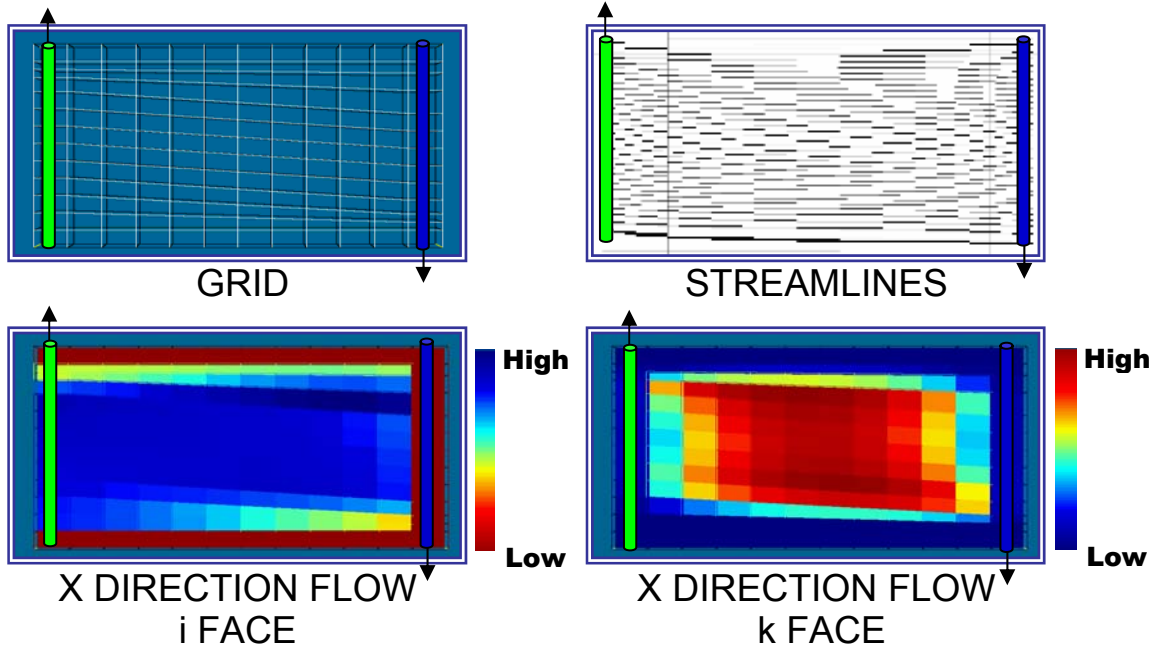


Fig. 4.12 Numerical Solution - Gradual Pinch-out Grid

The time-of-flight obtained with the numerical fluxes is presented in Fig. 4.13. The output is a fair enough representation of the straight evolving front from right to left.

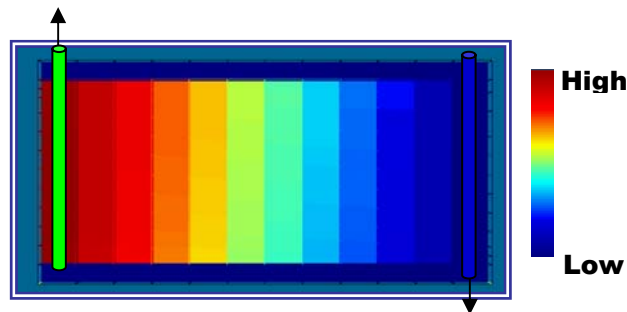


Fig. 4.13 Numerical Time of Flight - Gradual Pinch-out Grid

Regardless of the good numerical approximation the analytical solution is still presented. Once again the streamlines are perfectly horizontal, the numerical x-direction flow through the i-face is highly similar to the analytical one and there's a notable difference in the x-direction flow through the k-face.

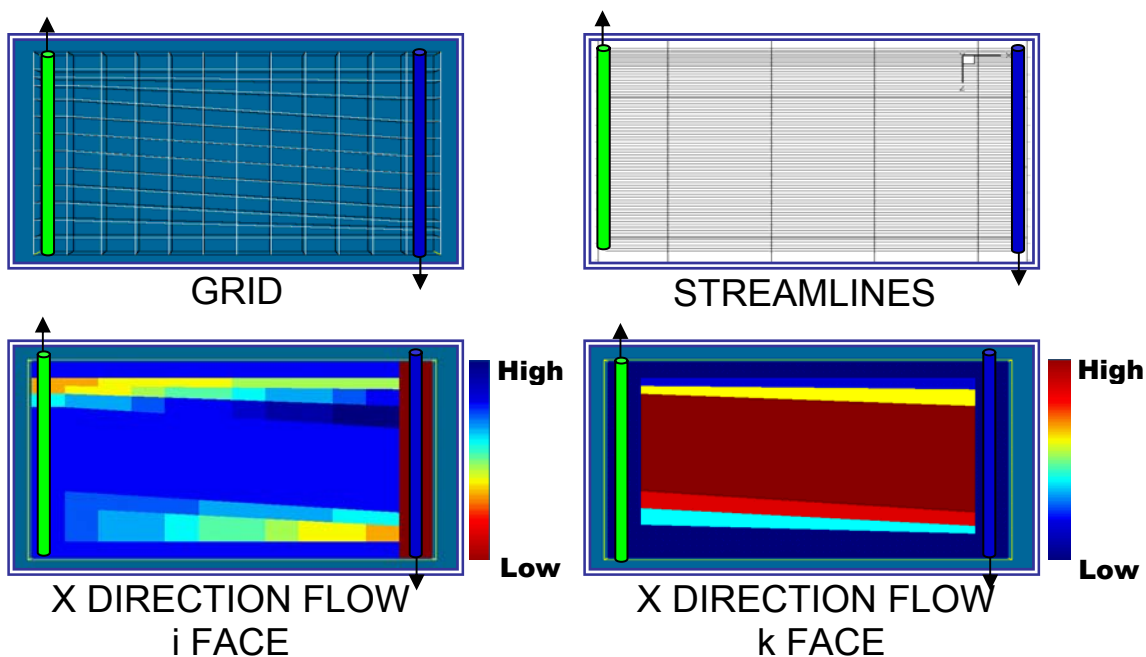


Fig. 4.14 Analytical Solution - Gradual Pinch-out Grid

There's little to say about the time of flight (Fig. 4.15) other than the straight and uniform evolution of the displacement front.

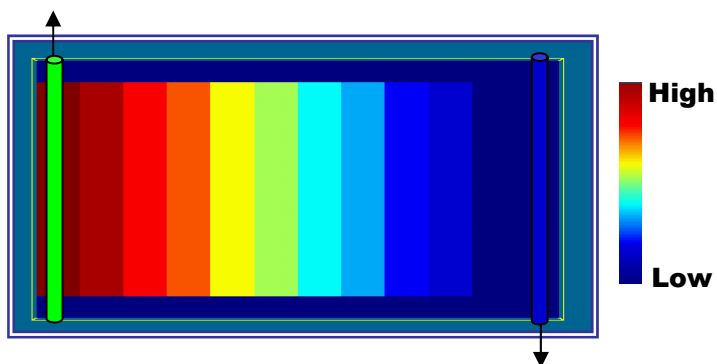


Fig. 4.15 Analytical Time of Flight - Gradual Pinch-out Grid

The difference between the numerical and the analytical solution is presented in Fig. 4.16, the difference is higher closer to the producer well and it decreases towards the injector. There's also an offset in the front displacement differences, at the same x-location the difference is higher at the bottom.

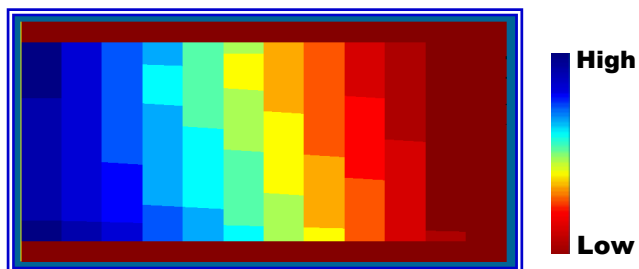


Fig. 4.16 Difference between Numerical and Analytical TOF - Pinch-out Grid

4.3.1.2. Layer Discontinuity

The second pinch-out is a little bit more complicated, the top and bottom layers are still acting as sealing units. Instead of having a gradual thinning of the sedimentary body, several layers are missed pinching-out near the sealing units. This geologic feature adds an extra complexity component since now we're dealing with triangular cells.

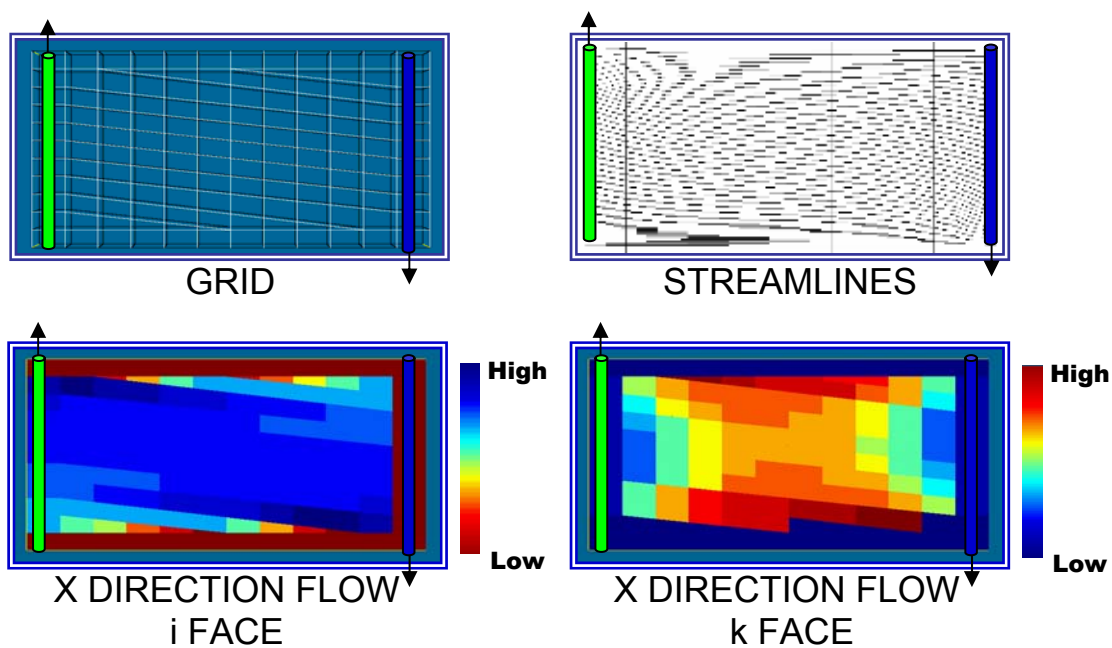


Fig. 4.17 Numerical Solution - Pinch-out Grid

Fig. 4.17 shows the numerical solutions to the pinch-out. It can be seen from the streamline paths that the code and therefore the formulation is doing a suitable job through triangular cells. The streamlines are nearly-horizontal, especially at the bottom where no vertical flux is present.

The highest rates through the k-face are occurring within the cells along the pinch-out direction, having the highest ones at the triangular cells. A major discrepancy is that the streamlines are not aligned with the direction of flow; they're aligned with the layering direction. According to this result the nine-point scheme proposed by ECLIPSE is not totally removing the non-orthogonal effects of the grid.

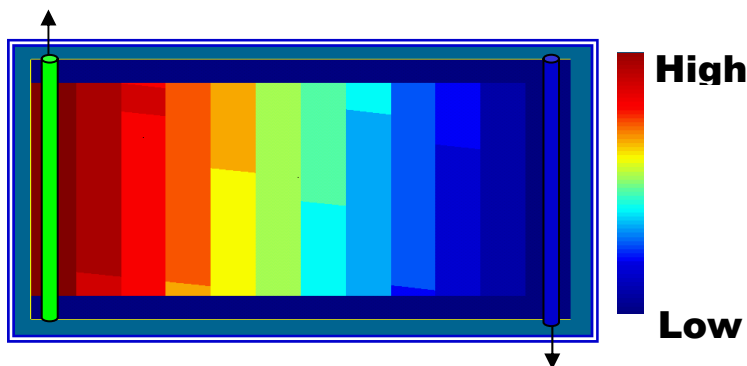


Fig. 4.18 Numerical Time of Flight - Pinch-out Grid

The time of flight is close to a straight evolving front from left to right, however this case shows a greater front distortion compared to all the previous ones; remember that in this case we're not only dealing with the pinch-out itself, we're also working with triangular cells and a higher non-orthogonality.

Fig. 4.19 shows the analytical solution for this type of pinch-out. The stream-function derived flux is just the difference between each cell corner depth multiplied by the injection rate per length. The numerical solution preserves the main flow pattern of the x-direction flow through i-face. On the other hand the flow through k-face has a severe difference, especially through the intermediate layers.

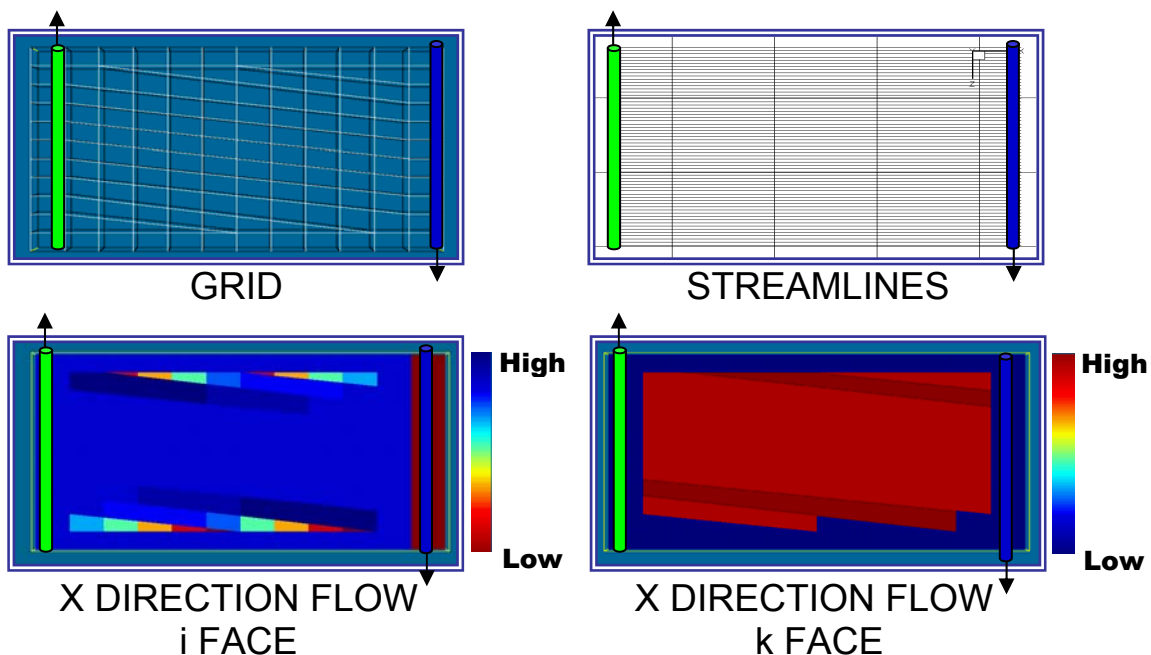


Fig. 4.19 Analytical Solution - -Pinch-out Grid

The exact solution of the displacement front is a straight line evolving from left to right (Fig. 4.20). While the simulation with a nine-point scheme results in a distorted front in the presence of non-orthogonal grid lines (Fig. 4.18). The difference between the solutions is higher towards the pinch-outs at the bottom and near to the producer.

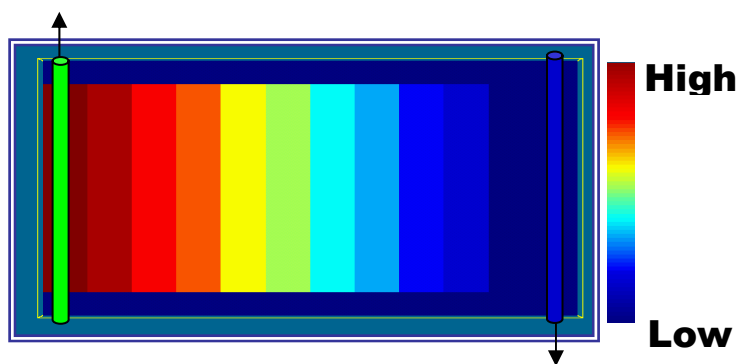


Fig. 4.20 Analytical Time of Flight - Pinch-out Grid

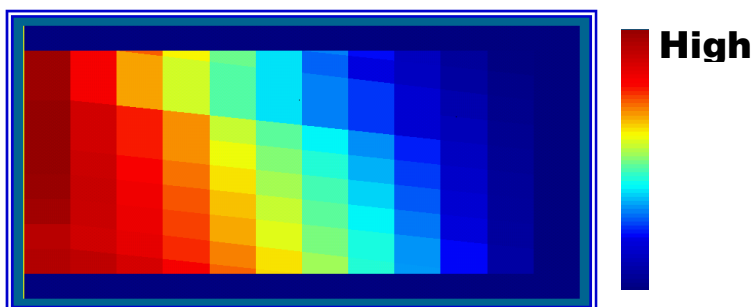


Fig. 4.21 Difference between Numerical and Analytical TOF -Pinch-out Grid

4.3.2. On-lap and Zone Boundaries

Another frequent and more challenging geologic feature involved in reservoir modeling, occurs when the reduction in the bed thickness result in on-lapping stratigraphic sequences. The size and shape of beds reflects the depositional process and environment. A common example occurs when a period of deposition dominated by some transport mechanism ceases and is followed by a different one. The infilling by the latter usually produces an on-lapping sequence featuring several sequence boundaries.

The simplest on-lap geometry is when an initially sub-horizontal stratum laps out against an initially inclined surface. Both on-laps and zone boundaries are indicators of non-depositional breaks (hiatuses). Usually the on-lap is accompanied by layer truncation which is the lateral termination of a stratum by erosion, and occurs at the upper boundary of a depositional sequence.

To model the on-lapping sequence, a grid showing several stratigraphic unconformities was built (Fig. 4.22). The grid shows two main sedimentary bodies with parallel strata separated by an angular unconformity where tilting and erosion of strata were followed by deposition. Fluid will be flowing from left to right and two boundary conditions will be modeled.

4.3.2.1. Closed Bottom Boundary

To model the boundary conditions an extra layer was added to the bottom of the grid, it will act as an open or closed boundary. The numerical solution for the bottom close boundary is presented in Fig. 4.22. In this case the streamlines paths follow a fanning shape. Both fluxes in i and k faces trend to squeeze near the right end of the on-lapping sequence. To model the flow through the on-lapping zone ECLIPSE automatically generates x -direction horizontal non-neighbor connections between the active cells on either side of the pinched-out columns, allowing fluid to flow through it in the x -direction.

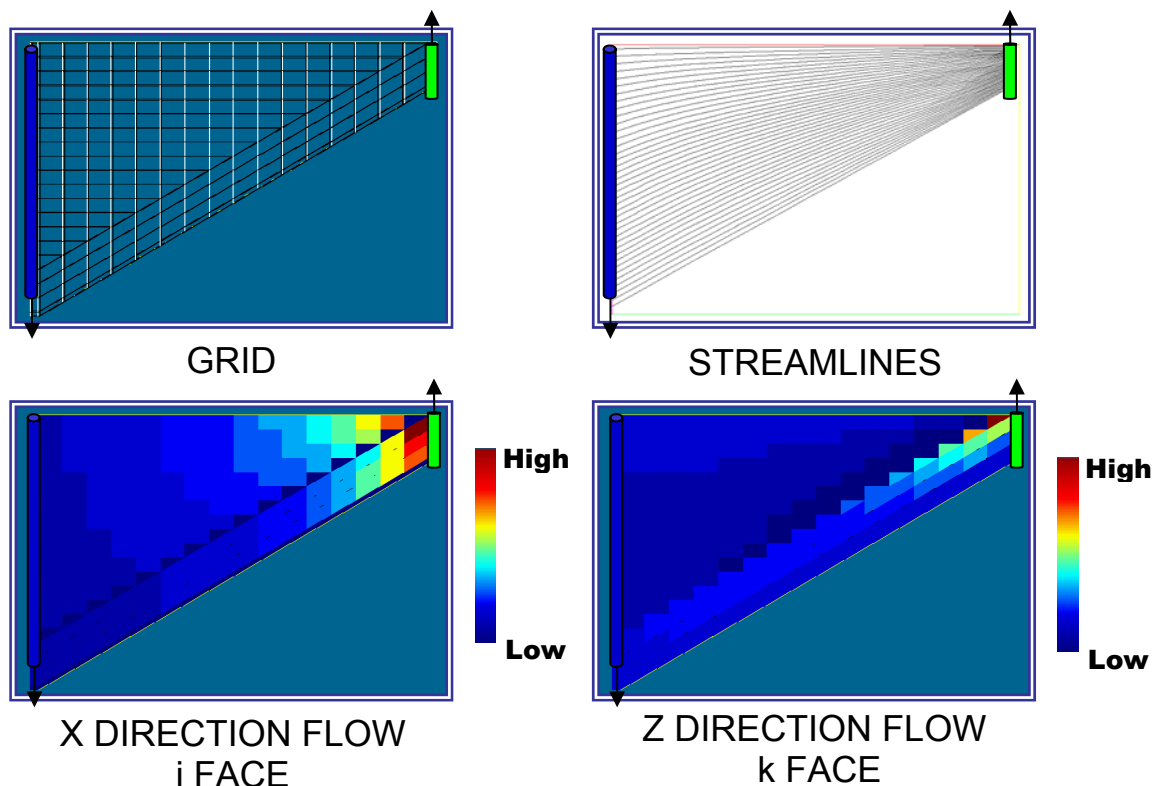


Fig. 4.22 Numerical Solution - Onlap Boundary Grid – Closed Bottom

Reviewing the numerical TOF on Fig. 4.23 several things can be appreciated. Both sedimentary bodies are having a considerable front distortion as the geometry is pinching-out. The distortion in the upper body is not as severe as the bottom one,

however there's a remarkable difference between the times required to reach the same x-location. The bottom body shows an uniform front in the first half of the pinch-out, but the second half is characterized for the sudden severe increase on the time of flight.

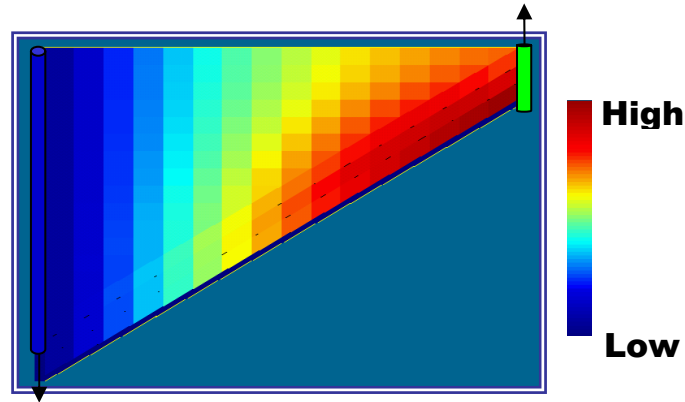


Fig. 4.23 Numerical Time of Flight - Onlap Boundary Grid – Closed Bottom

It has already been mentioned, how the analytical solution is found differentiating the stream-function and multiplying it by the injection rate per unit length. For the on-lapping sequences a diagram showing the overall concept is presented in Fig. 4.24. Instead of finding the difference between the node depths of each cell, each cell corner is mapped to the unit-transformed space. To perform this mapping, it is necessary to solve the non-linear system of equation arising when using the equations in section 3.2.2. The algorithm used is described by Powell²⁷ and is implemented by Press²⁶.

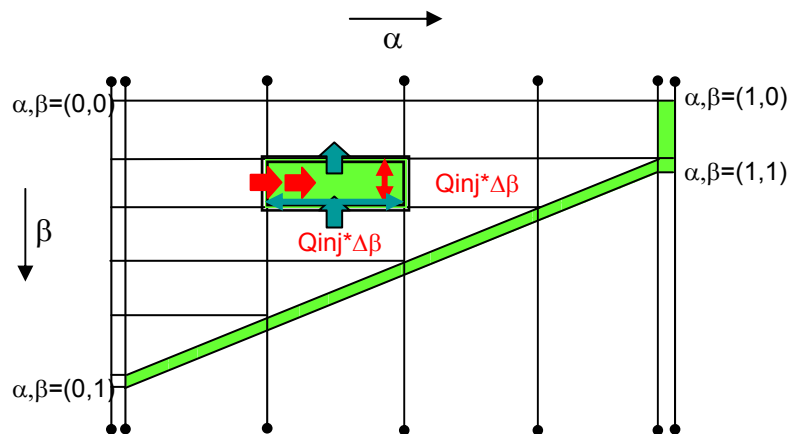


Fig. 4.24 Stream-Function Concept for On-lap Sequence.

Once the transformed coordinates are found, the real flux will be the difference between the transformed depths multiplied by the injection rate. The real solution can be appreciated as fluxes iso-lines going from 0 to 1 and directly proportional to the injection rate.

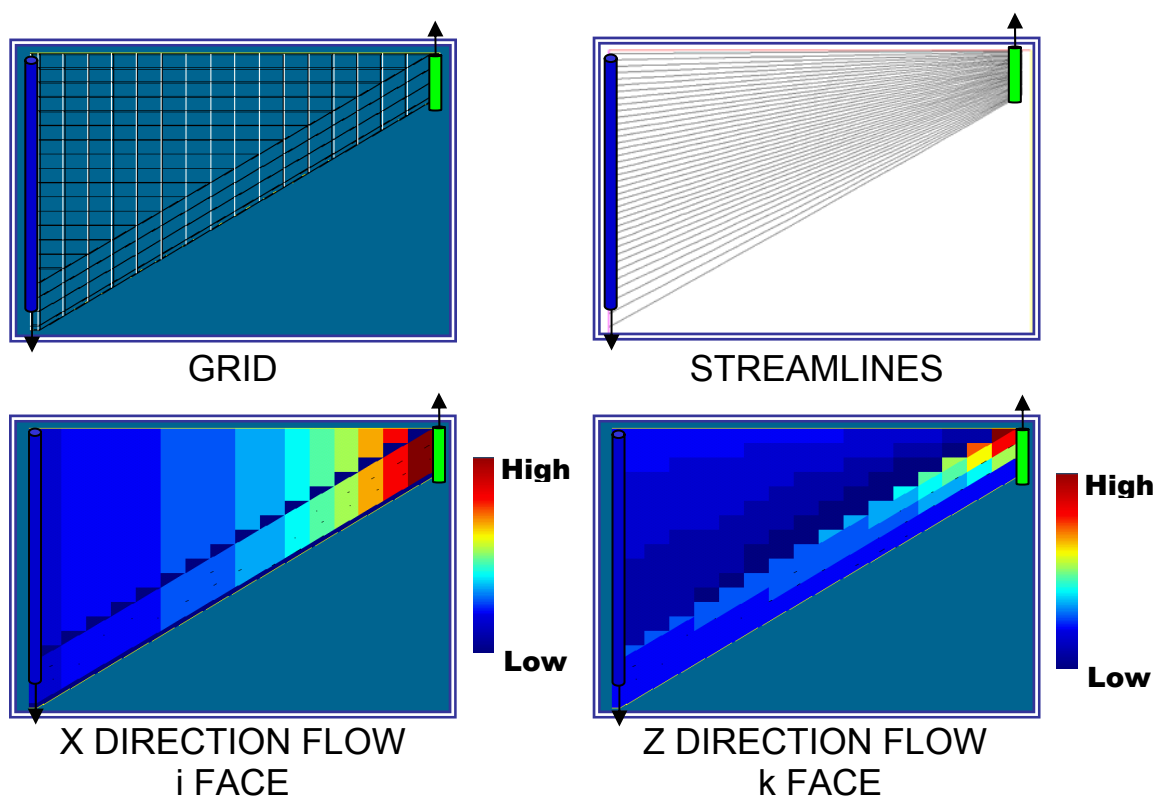


Fig. 4.25 Analytical Solution - Onlap Boundary Grid – Closed Bottom

The corresponding analytical solutions are presented in Fig 4.25 and 4.26. Just like the numerical solution, the streamline paths show a fanning trend uniformly distributed. The x-direction flow shows an uniform increasing velocity front, in contrast with the severely distorted numerical one. Curiously, the simulator is doing a better job approximating the z-direction flow compared to the previous pinch-out cases. The overall pattern in this direction is preserved and even some rate allocation details are conserved.

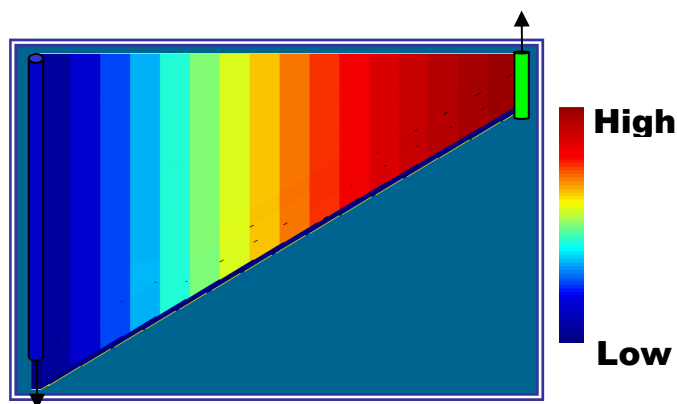
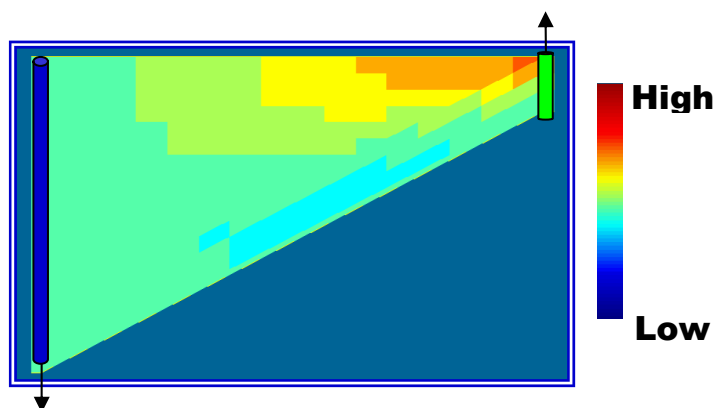


Fig. 4.26 Analytical Time of Flight - Onlap Boundary Grid – Closed Bottom

The time of flight is a straight-line front which is developing from left to right (Fig 4.26). This result is not surprising since the x-direction flow is a uniform evolving front too, and the vertical flux component is compensated with the streamline fan-shape nature.

Fig. 4.27 shows the difference between both numerical and analytical TOF. Both upper and bottom sedimentary bodies have the highest differences growing towards de truncation zone where the upper strata are terminated.



*Fig. 4.27 Difference between Numerical and Analytical TOF - Onlap Boundary Grid –
Closed Bottom*

4.3.2.2. Open Bottom Boundary

Fig. 4.28 shows the numerical solution for the open bottom boundary case. To model the open boundary, producing wells were added to the bottom layer and were constrained by rate. This can be appreciated in the z-direction flow where the last layer shows a constant flow rate.

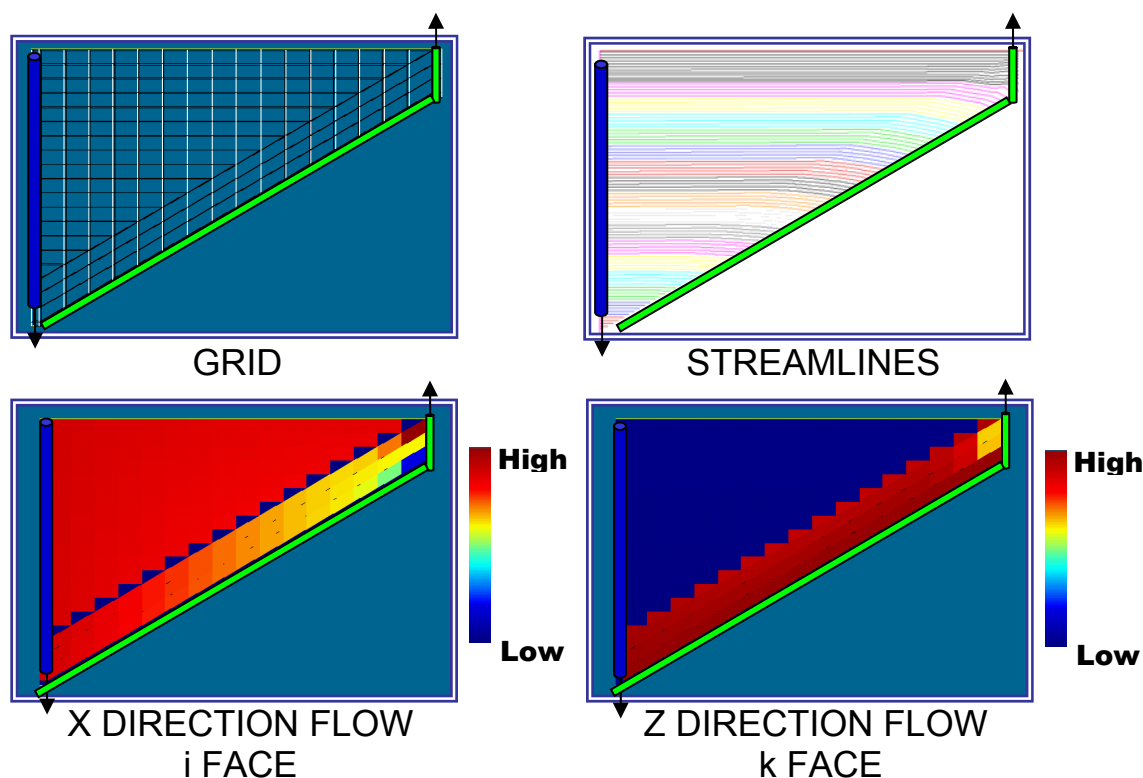


Fig. 4.28 Numerical Solution - Onlap Boundary Grid – Open Bottom

The streamline paths across the upper sedimentary body are nearly horizontal, but they have a wicker twist when they cross the on-lapping sequence. This effect is greater as the upper strata are terminated and is mainly related to the velocity fronts obtained at the bottom sedimentary body. This similar situation occurred in the closed-bottom boundary case, where the first half had a uniform front and the second one a sudden and abrupt distortion on it.

The time of flight (Fig. 4.29) is again a faithful reproduction of what is going on with the fluxes. The upper layers of the structure show a very uniform displacement front just like the x-direction flow and the horizontal streamline paths. The bottom layers follow the same trend just in the first half of the sequence; the second half is characterized by the abrupt distortion towards the termination of the upper layers and the time of flight increase.

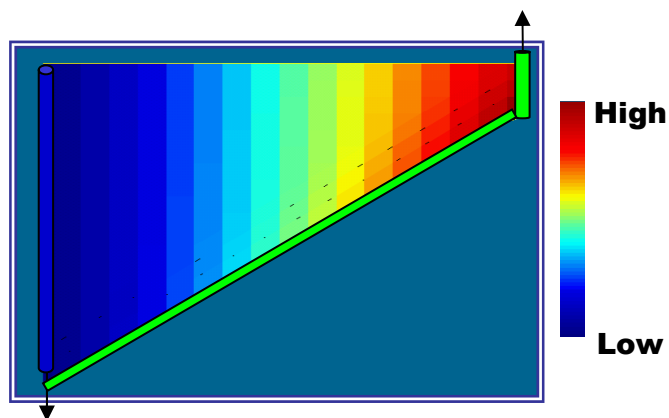


Fig. 4.29 Numerical Time of Flight - Onlap Boundary Grid – Open Bottom

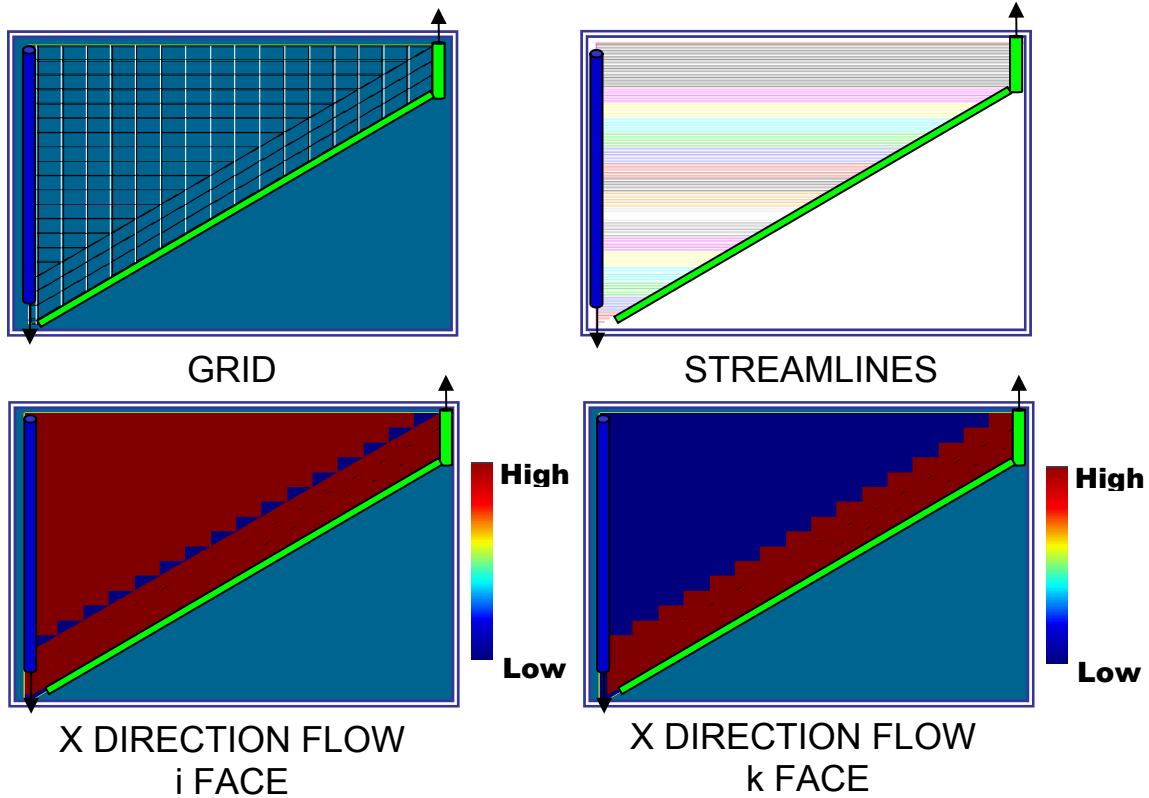


Fig. 4.30 Analytical Solution - Onlap Boundary Grid – Open Bottom

The analytical solution is presented in Fig. 4.30 and Fig. 4.31. The streamline horizontality is product of the upper strata zero vertical flux and the uniform horizontal flux distribution over the bottom one. The time of flight is again distinguished by its linear evolving front.

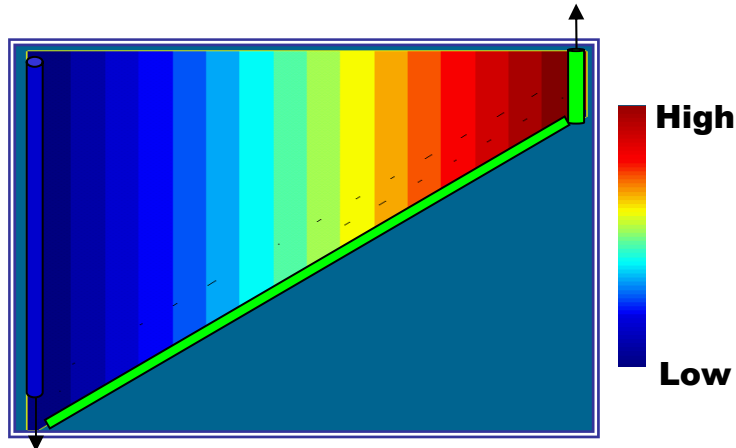


Fig. 4.31 Analytical Time of Flight - Onlap Boundary Grid – Open Bottom

Finally, Fig. 4.32 presents the differences between both numerical and analytical time of flight. As mentioned before, the upper section is well approximated by the numerical solution, while the bottom one has an unexpected growing non-accuracy headed for the end of the on-lapping sequence.

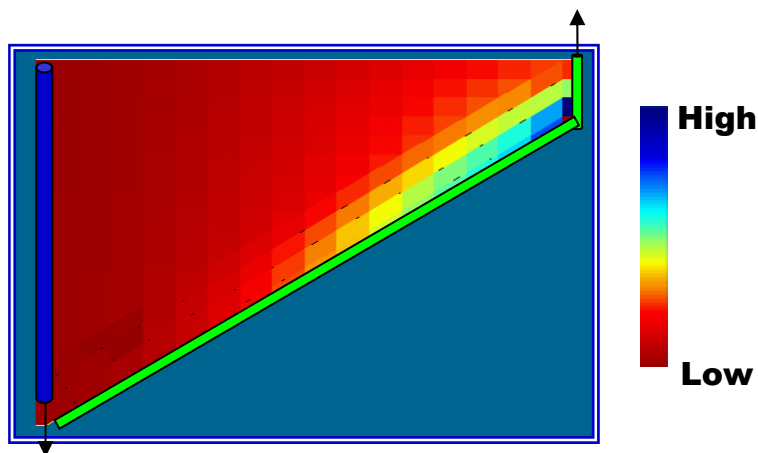


Fig. 4.32 Difference between Numerical and Analytical TOF - Onlap Boundary Grid – Closed Bottom

4.4. Grid Refinement Effects

In any type of computer simulation work, there might be accuracy problems associated with the cell size and eventually the number of cells used. The Buckley-Leverett model used for immiscible displacement calculations, such as the one used in

streamline simulation, assumes an incompressible system and ignores capillary pressure. A characteristic of these systems is the “shock” or clearly defined saturation front.

When using finite differences, the equations solved are parabolic, implying a smooth function with no discontinuities. Therefore, simulation does not give a clear front like Buckley-Leverett predicts. For large gridblocks the saturation definition is lost and numerical dispersion appears on the solutions. To remove and minimize this effect, smaller (or more) gridblocks are added to the model.

Within all previously presented cases, the pinch-out grid with missing layers was refined in both x and z direction to evaluate the numerical solution accuracy and reduce dispersion effects. In addition to the refinement, the grid non-orthogonality was increased using slanted coordinate lines instead of vertical lines. For all refined grids both numerical and analytical solutions are presented.

4.4.1. Vertical Coordinate Lines

Figs. 4.33 through 4.36 show both the numerical and analytical solution for the refined grids. In all three models, the upper right and bottom left gridblocks of the numerical x-direction flow through i-face (Fig.4.33) lose the main flow pattern observed in the analytical solution. On the other hand, the x-direction flow through k-face (Fig. 4.34) is far from being a good representation of the real solution, there's a severe dissimilarity present within all gridblocks. Finally, the time of flight and the streamline paths (Figs. 4.35 and 4.36) are clearly showing that the grid non-orthogonality is causing the displacement front to be aligned with the principal directions of the grid, when it should be a straight line evolving front.

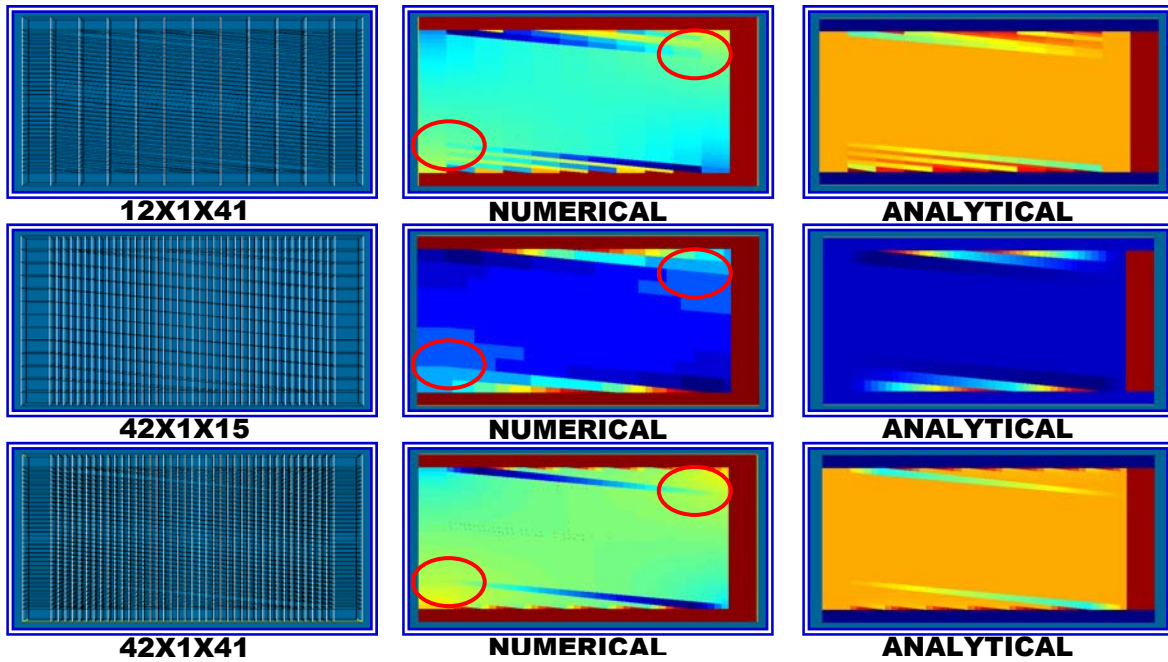


Fig. 4.33 Refined X-Direction Flow i-Face - Vertical Coordinate Lines

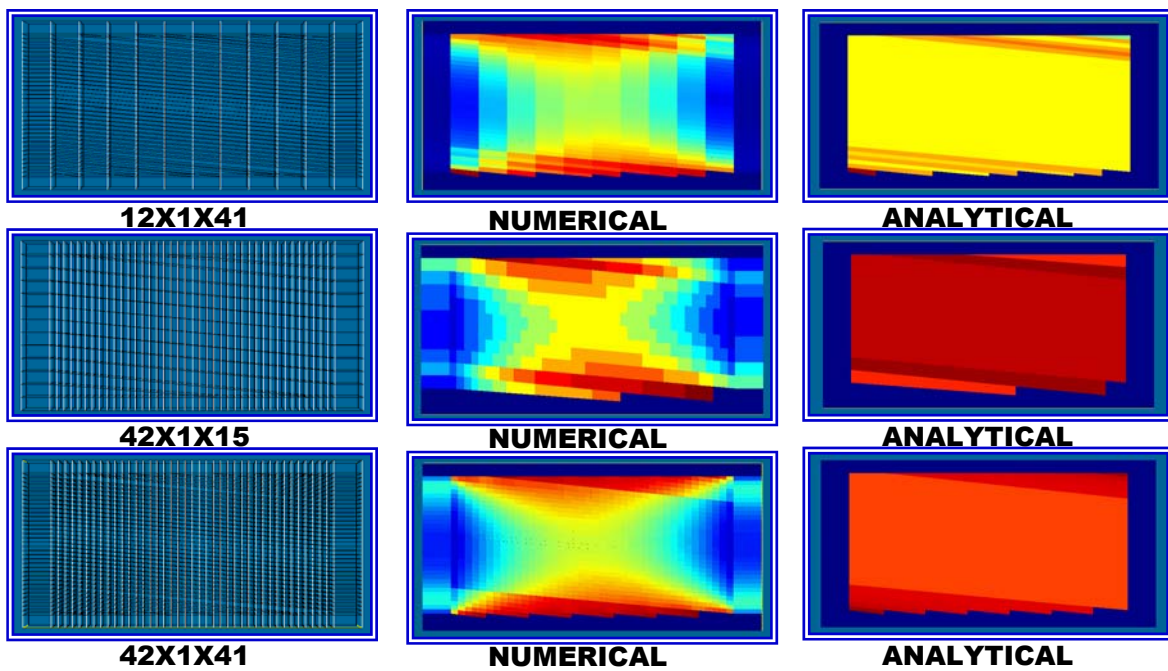


Fig. 4.34 Refined X-Direction Flow k-Face - Vertical Coordinate Lines

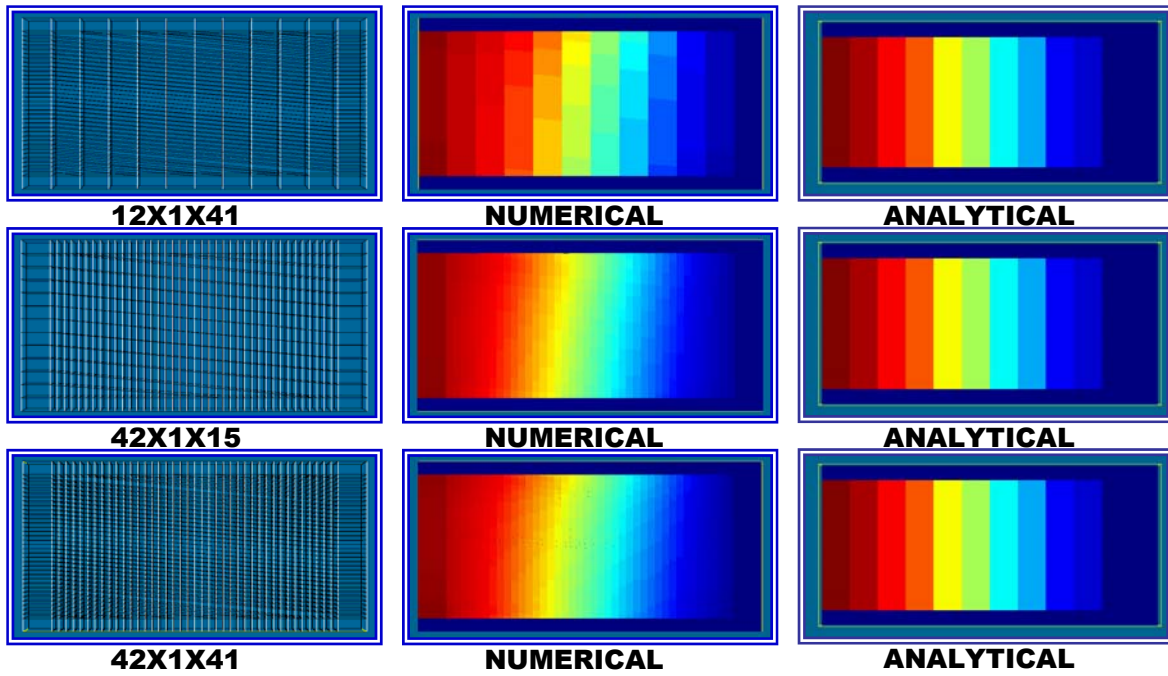


Fig. 4.35 Refined Time of Flight - Vertical Coordinate Lines

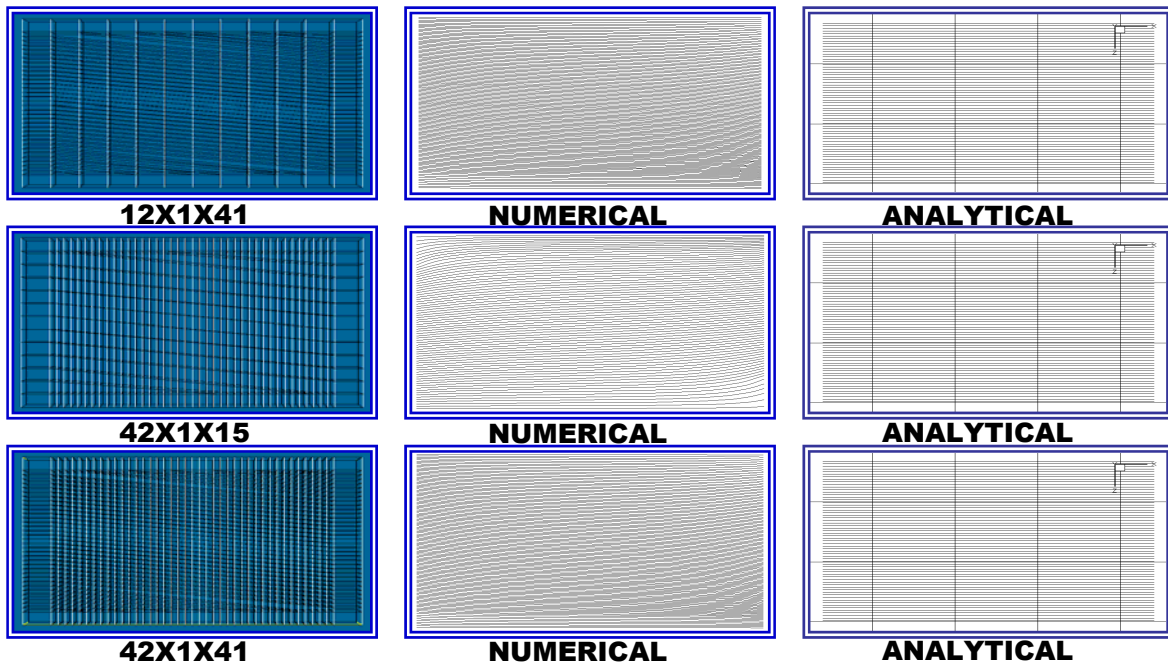


Fig. 4.36 Refined Streamline Paths - Vertical Coordinate Lines

4.4.2. Slanted Coordinate Lines

Within all presented cases, it has been shown how the grid geometry can seriously compromise the accuracy and reliability of the numerical solution. The last cases to be presented will not only have the non-orthogonality associated with the layering geometry; they will also have a non-orthogonal component in the coordinate lines used to build the CPG grid.

The numerical and analytical solutions are presented in Figs. 4.37 through 4.40. Again, the numerical solutions show the same non-accuracies in both x-direction flow through i and k faces. The displacement fronts are severely distorted along the upper gridblocks of the pinch-out, where the highest non-orthogonality is present. The distortion is also appreciated in the streamline trajectories.

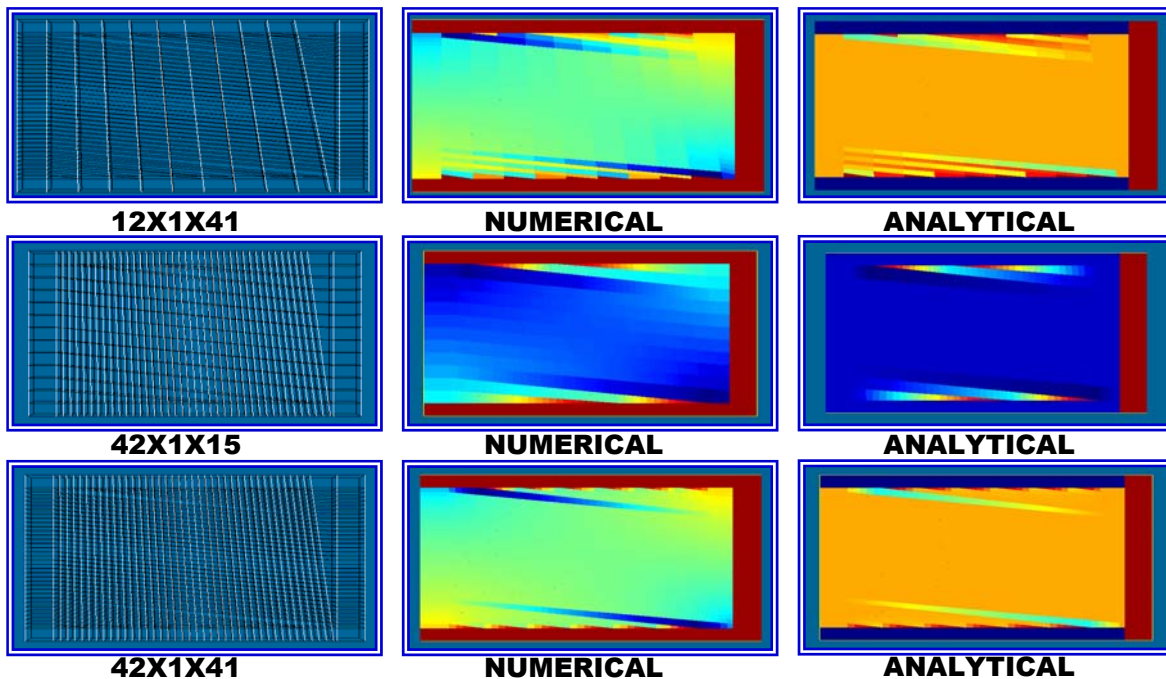


Fig. 4.37 Refined X-Direction Flow in i-Face - Slanted Coordinate Lines

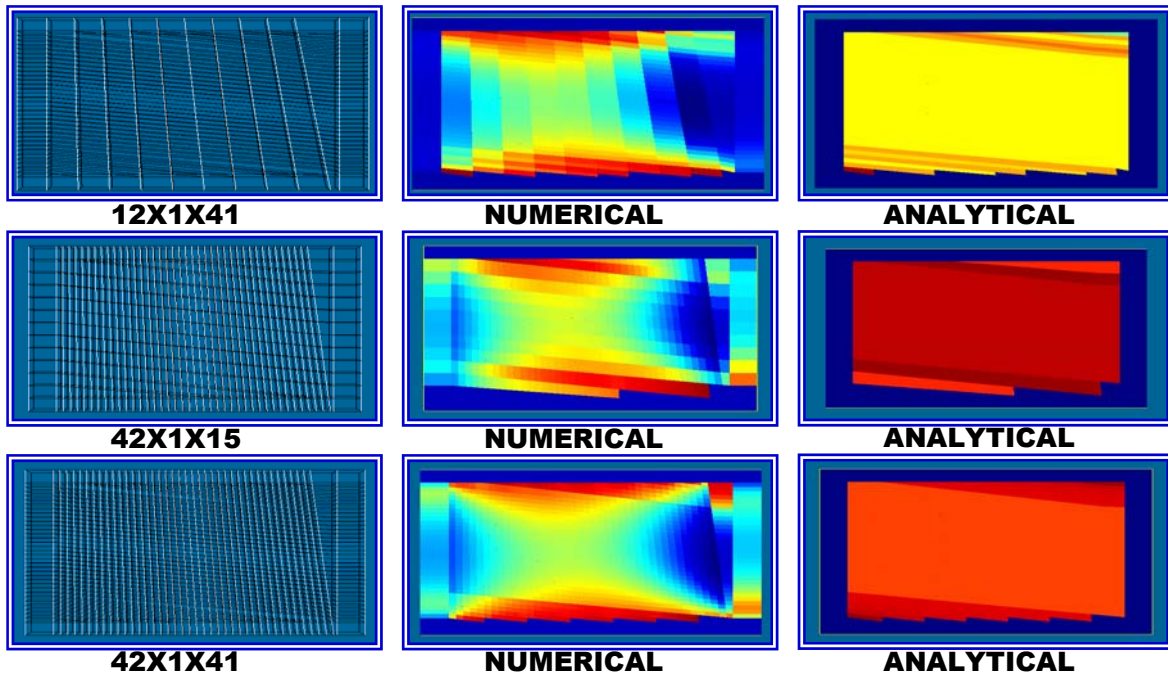


Fig. 4.38 Refined X-Direction Flow in k-Face - Slanted Coordinate Lines

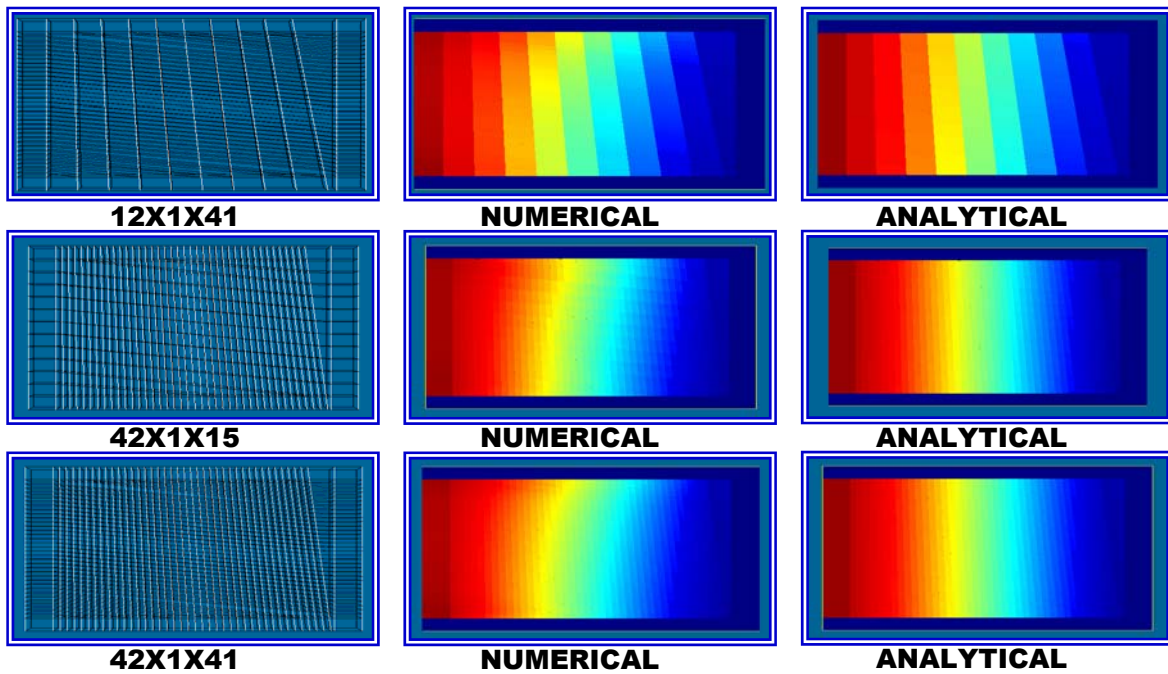


Fig. 4.39 Refined Time of Flight - Slanted Coordinate Lines

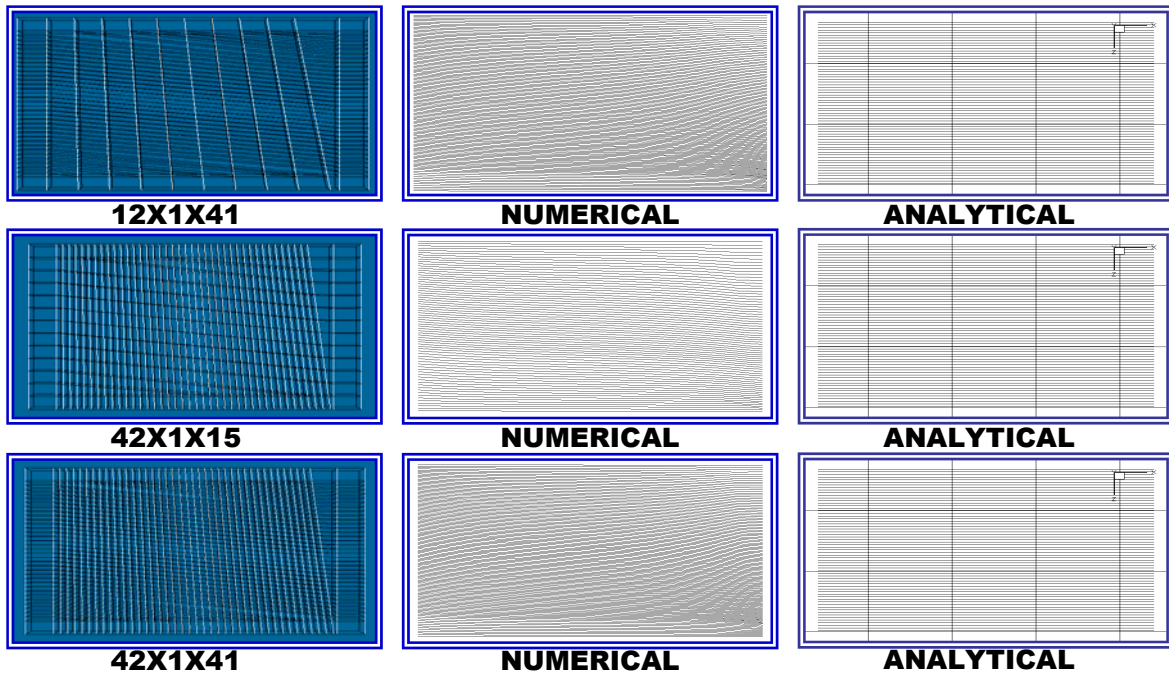


Fig. 4.40 Refined Streamline Paths - Slanted Coordinate Lines

CHAPTER V

CONCLUDING REMARKS AND RECOMMENDATIONS

5.1. Conclusions

- A new streamline method to handle fluid displacements within complex reservoir geometries has been developed. We managed to take full advantage of the uniqueness and usefulness of streamline simulation to approach the difficulties found in modeling structurally demanding geometries.
- We have developed and implemented the proposed procedure in a FORTRAN program for tracing streamlines in the grid domain and the corresponding time of flight calculation.
- The Corner point geometry freedom can lead to extreme nonorthogonality, exerting a significant impact on the flow performance and subsequent operational decisions involved in reservoir management. The grid non-orthogonality may cause the displacement front to be aligned with the principal directions of the grid instead of the flow direction.
- The pseudo-time-of-flight introduction provides a significantly simpler and more robust development to handle the complexity in structurally demanding and faulted systems. We have provided a new basis for time of flight calculations and we have used it as an independent indicator of grid geometry effects on displacements calculations.

- The extension to the tracing algorithm retains the simplicity and speed featured in streamline-based flow. The extension is able to handle complex geologic features such as faults, onlap boundaries and distorted grids.
- We have compared the results with existing commercial simulators, comparing the accuracy of all the involved calculations with streamline-derived analytic solutions. Several discrepancies were found related to grid orientation and non-orthogonality components.

5.2. Recommendations

We used the commercial simulator ECLIPSE as the forward tool to obtain the fluxes required to trace the streamlines in all presented geometries. Another widely used simulator is VIP, we recommend to perform the same numerical exercises using VIP and compare the results between both simulators.

The program has not been tested using real data, it's highly recommended to run some field cases in order to have an operational and confident tool. The main goal will be to couple the code with any finite difference software and perform the streamline procedure.

Dealing with non-neighbor connections is a required geologic feature present in almost every reservoir model. It is imperative to extend the ability of the code to handle heavily faulted systems. To accomplish this, it might be necessary to re-design the FORTRAN code into an object oriented structure such as the one used in C++.

REFERENCES

1. Muskat, M., "Flow of Homogeneous Fluids," Intl. Human Resources Development Corp., 137 Newbury Street, Boston Massachusetts 02116 (1937, 1982).
2. LeBlanc, J.L. and Caudle, B.H., "A Streamline Model for Secondary Recovery," *SPE Journal* (March 1971), pp 7-12.
3. Martin, J.C. and Wegner, R.E., "Numerical Solution of Multiphase Two-Dimensional Incompressible Flow Using Streamtube Relationships," *SPEJ* (October 1979) 313, Trans., AIME, **267**.
4. Emanuel, A.S., Alameda, G.K., Behrens, R.A., and Hewett, T.A.: "Reservoir Performance Prediction Methods Based on Fractal Geostatistics," *SPEJ* (August 1989) 311, Trans., AIME, **287**.
5. Bratvedt, F., Bratvedt, K., Buchholz, C. F., Gimse, T., Holden, H., Holden, L. and Risebro, N. H., "FRONTLINE and FRONTSIM. Two Full Scale, Two-Phase, Black Oil Reservoir Simulators Based on Front Tracking," *Surv. Math. Ind.* (1993), **3**, 185.
6. Datta-Gupta A. and King M.J.: "A Semianalytic Approach to Tracer Flow Modeling in Heterogeneous Permeable Media," *Adv. in Water Resources* (1995), **18**, No. 1, 9.
7. King, M. J. and A. Datta-Gupta, "Streamline Simulation: A Current Perspective," *In Situ* (1998), **22**, No. 1, 91.
8. Bratvedt, F., Gimse, T. and Tegnander, C., "Streamline Computations for Porous Media Flow Including Gravity," *Transport in Porous Media* (1996), **25**, 63.
9. Crane, M. J. and M. J. Blunt, "Streamline-based simulation of solute transport," *Water Resour. Res.* (1999), **35**, No. 10, 3061.
10. Fay, C.H. and Prats. M.: "The Application of Numerical Methods to Cycling and Flooding Problems," *Proc. of the 3rd World Petroleum Congress* (1951)
11. Pollock, D.W.: "Semi-analytical Computation of Path Lines for Finite Difference Models," *Ground Water* (November/December 1988) **26**, No. 6, 743.
12. Shafer, J.M.: "Reverse Pathline Calculation of Time-Related Zones in Nonuniform Flow," *Ground Water* (May/June 1987) **25**, No. 3, 283.

13. King, M. J., Blunt, M., Mansfield, M. and Christie, M. A., "Rapid Evaluation of the Impact of Heterogeneity on Miscible Gas Injection," paper SPE 26079 presented at the SPE Western Regional Meeting (1993).
14. Martin, J.C., Woo, P.T., and Wegner, R.E.: "Failure of Stream Tube Methods to Predict Waterflood Performance of an Isolated Inverted Five-Spot at Favorable Mobility Ratios," *Journal of Petroleum Technology* (February 1973) **25**, 151-153.
15. Blunt, M.J., Lui, K., Thiele, M.R.: "A Generalized Streamline Method to Predict Reservoir Flow," *Petroleum Geoscience* (1996) **2**, 259-269.
16. Batycky, R.P., Blunt, M.J., and Thiele, M.R.: "A 3D Field Scale Streamline Simulator with Gravity and Changing Well Conditions," paper SPE 36726 presented at the 1996 Annual Technical Conference and Exhibition, Denver, Co, October 1996.
17. Behrens, R. A., Jones, R. C. and Emanuel, A. S., "Implementation of a Streamline Method for Flow Simulation of Large Fields," *Journal of Canadian Petroleum Technology, Special Edition* (1999), **38**, 13.
18. Sabir K.: "Velocity Models, Material Balance and Solution Convergence in Streamline-Based Simulation," M.S. Thesis, Texas A&M University, College Station, TX (December 2002)
19. Aziz, K. and Settari, A., *Petroleum Reservoir Simulation*, Applied Science Publishers, Essex, England (1979).
20. Bear, J., *Dynamics of Fluid in Porous Media*, Dover Publications, New York (1973).
21. Schlumberger Geoquest, *ECLIPSE 100 Reference Manual*, Geoquest Simulation Software Launcher, Texas A&M University, (Aug. 2003).
22. Ponting, D. K., "Hybrid Streamline Methods," paper SPE 39756, presented at the 1998 Asia Pacific Conference on Integrated Modeling, Kuala Lumpur, Malaysia, 23-24 March, 1998.
23. Kulkarni, K. N., Datta-Gupta, A. and Vasco, D. W., "A Streamline Approach to Integrating Transient Pressure Data into High Resolution Reservoir Models," *SPE Journal* (September 2001), **6**(3), pp 273-282.
24. Bratvedt, F., Bratvedt, K., Buchholz, C. F., Holden, L., Holden, H. and Risebro, N. H., "A New Front Tracking Method for Reservoir Simulation," *SPE Reser. Eng.* (1992), **7**, 107.

25. Cordes, C. and Kinzelbach, W.: “Continuous Groundwater Velocity Fields and Path Lines in Linear, Bilinear and Trilinear Finite Elements,” *Water Resour. Res.* (1992) **28**, 2903.
26. Press, W.H., Flannery, B.P., Teukolsky, S.A. and Wetterling, W.T.: *Numerical Recipes – The Art of Scientific Computing*, Cambridge University Press, Cambridge (1987).
27. Powell, M.J.D.: “A Fortran Subroutine for Solving System of Nonlinear Algebraic Equations,” *Numerical Methods for Nonlinear Algebraic Equations*, Rabinowitz, P. Editor, Gordon and Breach Science Publishers, London (1970).

VITA

Eduardo Antonio Jimenez Arismendi

Petroleum Eng. Dept.

3116 TAMU

College Station, TX USA, 77843

Ph: (979)847-8797

jimaris@yahoo.com

PROFILE

Petroleum engineer with four years of experience in reservoir engineering. Specific knowledge in geostatistics, reservoir and streamline simulation, integrated reservoir characterization, and software development. Currently, involved in research concerning stochastic reservoir characterization and streamline simulation. Special interest in the development and application of high-resolution numerical schemes for reservoir simulation, gridding effects on fluid displacements, dynamic data inversion, software design and development.

EDUCATION

Master of Science. Petroleum Engineering. Texas A&M University. August 2004

Bachelor of Science. Petroleum Engineering. Universidad Industrial de Santander.
March 2000

EXPERIENCE

Texas A&M University. Research Assistant. 2002-2004

Ecopetrol-Colombia. Reservoir Engineer. 2000-2002