FAST MARCHING METHODS: APPLICATION VIA INTEGRATION WITH COMMERCIAL E&P SOFTWARE

A Thesis

by

MUHAMMED HASSAN AL-RUKABI

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

MASTER OF SCIENCE

Chair of Committee,	Akhil Datta-Gupta
Committee Members,	Michael King
	Yalchin R. Efendiev
Head of Department,	Dan Hill

December 2014

Major Subject: Petroleum Engineering

ABSTRACT

Development and production of unconventional reservoirs, especially shale, are on the rise and so is the need to better understand drainage volumes, reliably estimate reservoir properties, and forecast well performance. Numerical simulation and analytical techniques, like decline curve analysis and pressure transient analysis, have been applied to unconventional resources. However, analytical methods rely on several simplifications and while numerical simulation can account for complex geological models it is computationally expensive. Fast Marching Methods (FMM), being a semi-analytical calculation, is between the two approaches and retains the simplicity of the analytical approach while achieving the desired generality.

The generalization of the concept of depth of investigation to heterogeneous reservoirs utilizes the idea of diffusive time-of-flight and better accounts for the non-uniform pressure fronts that may be distorted due to heterogeneity effects. The pressure front propagation is obtained by solving the Eikonal equation, which is derived from an asymptotic solution of the diffusivity equation. The FMM solves the Eikonal equation very efficiently using a single non-iterative solution, making it very fast. The FMM estimates the drainage volume and the diffusive time of flight can be used as a spatial coordinate to reduce the 3D diffusivity equation into a 1D equation allowing for rapid forecasting of well pressure and rate performance.

In this work, the FMM is implemented into an application plug-in and is integrated with a common commercial E&P software platform. The integration of the FMM Plug-in capitalizes on the simplicity, intuitive appeal, power and utility of the approach, like providing the time-evolution of the drainage volume for visualization, and utilizes the software platform features, like state-of-the-art visualization tools. This work also includes a number of applications that demonstrate the capability of FMM Plug-in to calculate the drainage volume and forecast well pressure or rate performance and validate its results against an industry-reference finite difference simulator. Finally, a study on the scalability of calculations runtime demonstrate the speed advantage that FMM has over finite difference simulators.

DEDICATION

To my parents

For their unconditional love and support

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my committee chair, Dr. Datta-Gupta, for being a great advisor and teacher and for his patience, support, and guidance. I would like to extend my appreciation to my committee members, Dr. King and Dr. Efendiev, for their valuable discussions and constructive remarks. I would also like to thank Dr. Nasrabadi for contributing and substituting for Dr. King during my thesis defense.

Thanks also go to my MCERI colleagues, for their mentorship and friendship, and to the rest of my friends in the petroleum department.

I would like to extend my gratitude to the ExxonMobil Corporation and the Institute of International Education for supporting and financing my academic career.

Last but not least, I would like to express my gratefulness to my parents for their never-ending encouragement, support, and motivation. All my accomplishments are but results to their tireless efforts.

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iv

ACKNOWLEDGEMENTS v

TABLE OF CONTENTS vi

LIST OF FIGURES viii

LIST OF TABLES x

CHAPTER I INTRODUCTION 1

1.1 Background 1
1.2 Thesis Outline 3

CHAPTER II FAST MARCHING METHODS: METHODOLOGY 5

2.1 Depth of Investigation 5
2.2 Asymptotic Solution of Diffusivity Equation 8
2.3 Drainage Volume Calculation Using Fast Marching Method 9
2.4 Geometric Approximation Using Drainage Volume 15
2.5 Coordinate Transformation 19
2.6 Synthetic Example 21

CHAPTER III FAST MARCHING METHOD PLUG-IN 23

3.1 Introduction to Petrel E&P Software Platform 25
3.2 Ocean Software Development Framework 28
3.3 Fast Marching Method Plug-in Workflow 34

CHAPTER IV APPLICATIONS AND TESTING 45

4.1 Example I: Heterogeneous Reservoir with Multi-Stage Hydraulic Fractures 45
4.2 Example II: Dual Porosity 49
4.3 Scalability of Runtime 52

CHAPTER V CONCLUSIONS AND RECOMMENDATIONS 56

5.1 Summary and Conclusions......56
5.2 Recommendations......58

NOMENCLATURE......59

REFERENCES......61

APPENDIX A FMM PLUG-IN USER GUIDE......64

A.1 Installation......64
A.2 User Interface......66
A.3 Keywords in Settings File for a Pressure Calculation......68

LIST OF FIGURES

Page

Figure 1.1: U.S. natural gas production in tcf (EIA, Annual Energy Outlook 2014) 2

Figure 2.1: Illustration of Fast Marching Method (Reprinted with permission from Xie et al. 2012). 11

Figure 2.2: A synthetic example of a heterogeneous reservoir with 5 hydraulic fractures. (a) permeability field; (b) the geometry of five hydraulic fractures; (c) calculated DTOF; (d) drainage volume in 1 month; (e) drainage volume in 30 years. (Reprinted with permission from Zhang et al. 2013) 13

Figure 2.3: A synthetic example of a heterogeneous reservoir. (a) geometry of three hydraulic fractures; (b) permeability field. 22

Figure 2.4: Comparison of production rate calculated by FMM and ECLIPSE 22

Figure 3.1: Petrel user interface 27

Figure 3.2: Significant growth in number of Ocean plug-ins (Reprinted with permission from Wharton 2010) 29

Figure 3.3: Architecture of the Ocean Framework 31

Figure 3.4: Plug-in Modules can access all Ocean API layers 32

Figure 3.5: Access to Petrel data via Domain Objects 33

Figure 3.6: FMM Plug-in workflow 35

Figure 3.7: Process of determining location and direction of completed grid cells 38

Figure 3.8: Ocean API intersects a well trajectory with a reservoir model grid 39

Figure 4.1: FMM Plug-in user-friendly interface 47

Figure 4.2: FMM results displayed using Petrel visualization windows 47

Figure 4.3: Drainage volume at (a) 1 month; (b) 1 year; (c) 10 years; (d) 30 years 48

Figure 4.4: Calculated BHP by FMM compared with results from ECLIPSE 49

Figure 4.5: Fracture permeability field for Dual Porosity case 50

Figure 4.6: Diffusive Time of Flight distribution 50

Figure 4.7: Gas rate forecast by FMM compared with results from ECLIPSE for a Dual Porosity reservoir...51

Figure 4.8: Scalability of CPU runtime (2D) 54

Figure 4.9: Scalability of CPU runtime (3D) 55

Figure A.1: FMM Plug-in installation package 64

Figure A.2: Installation using Ocean Plug-in Manager in Petrel 65

Figure A.3: Successfully-installed FMM Plug-in in Processes Panel..............................66

Figure A.4: FMM Plug-in user interface...67

Figure A.5: Calculations log 68

LIST OF TABLES

Page

Table 2.1: Properties for synthetic horizontal well with three hydraulic fractures 21

Table 3.1: FMM input file types & content 41

Table 3.2: FMM result text files... 43

Table 4.1: Model parameters for multi-stage hydraulically fractured reservoir 46

Table 4.2: Parameters used in the Dual Porosity example 50

Table 4.3: Reservoir properties used for scalability analysis 52

Table 4.4: CPU runtime using FMM and ECLIPSE for 2D cases 53

Table 4.5: CPU runtime using FMM and ECLIPSE for 3D cases 54

CHAPTER I

INTRODUCTION

1.1 Background

Strong demand for natural gas and technological advances, such as horizontal well drilling and multistage hydraulic fracturing, are turning unconventional reservoirs into a key element of the energy supply worldwide, especially in the United States (Dong et al. 2013). The increasing scarcity of conventional resources and low production costs in unconventional reservoirs are also driving the growth of the natural gas industry. While demand for petroleum and other liquids in the United States is expected to see a small decline, natural gas consumption grows by 0.8%/year between 2012 and 2040, according to the 2014 Annual Energy Outlook by the US Energy Information Administration (EIA). Most of the growth in natural gas supply comes from shale gas, which will account for 53% of natural gas production by 2040 as shown in Figure 1.1 (EIA 2014).

It is important to understand the drainage volume, reliably estimate reservoir properties, and predict well behavior in unconventional reservoirs in order to optimize production performance. Several analytical and numerical methods for production forecasting are well described in the literature. Decline curve analysis (Fetkovich 1980, Valko and Lee 2010) and pressure/rate transient analysis (Ilk et al. 2010, Song and Ehlig-Economides 2011) are some of the commonly used analytical methods. Decline curve analysis is widely used in the industry to predict future well behavior by matching production history with trend functions. It is also used to estimate ultimate recovery by extrapolation to abandonment conditions. However, it assumes a constant well bottom-hole pressure, which is not always the case. Moreover, it is highly dependent on the quality and quantity of completion and production data. Pressure transient analysis is developed for radial wells and homogenous reservoir properties with simple boundary conditions. It relies on identifying flow regimes to estimate reservoir parameters.

Conventional numerical simulation techniques have also been used to account for complex fracture geometry and heterogeneity (Cipolla et al. 2009, 2011; Fan et al. 2010). However, numerical simulation tend to be computationally expensive. On the other hand, many analytical methods generally use curve-fitting or use simplified models and are not suitable for heterogeneous and complex reservoir conditions.

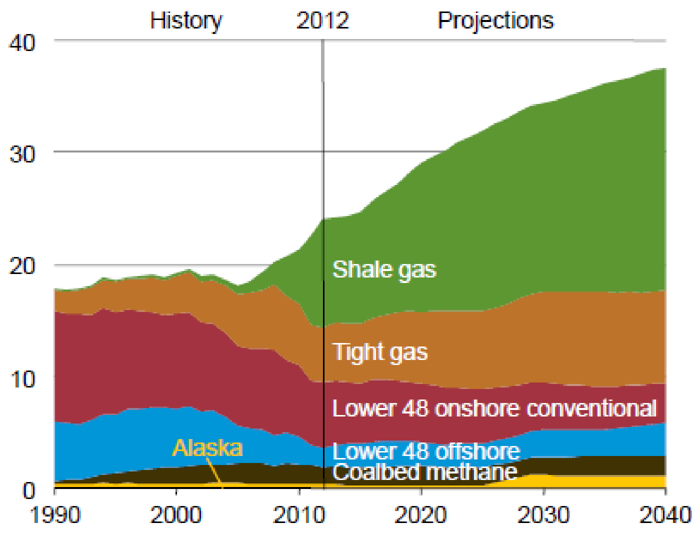


Figure 1.1: U.S. natural gas production in tcf (EIA, Annual Energy Outlook 2014)

Fast Marching Method (FMM) is a novel approach for unconventional reservoir analysis that serves as a bridge between simplified analytical models and expensive conventional numerical simulation (Datta-Gupta et al. 2011; Xie et al. 2012a, 2012b; Zhang et al. 2013, 2014). This thesis summarizes the concept of depth of investigation in homogenous reservoirs, its generalization to heterogeneous reservoirs, and the introduction of the pressure front propagation equation. It also describes the Fast Marching Method, a front tracking method, as an efficient technique of solving the Eikonal equation, which is derived from an asymptotic solution of the diffusivity equation. The Fast Marching Method uses a single non-iterative calculation making it highly efficient and fast. The solution of the Fast Marching Method provides a ‘diffusive time of flight’ distribution that is used to estimate the drainage volume of the reservoir. The calculated diffusive time of flight can be used as a spatial coordinate to reduce the 3D diffusivity equation into a 1D equation allowing for rapid forecasting of well pressure and rate performance. Alternatively, a geometric approximation can be applied once the pressure fronts are determined to yield pressure depletion and well rates.

1.2 Thesis Outline

The main objective of this work is to design and develop a computer application that implements the Fast Marching Method and that is integrated with a common commercial E&P software platform in the form of an application plug-in. While an existing implementation (known as FMM processer) has been developed, alone it lacks the capability to prepare and obtain input data and also lacks the means to visualize results.

This work aims to repackage the FMM processer to overcome the absence of pre-processing and post-processing capabilities. The new computer application, called FMM Plug-in, is integrated with the Petrel software platform using the Schlumberger Ocean software development framework. Integrating with the Petrel platform, gives the FMM Plug-in access to Petrel's extensive data sets and enables it to collect input required to perform the calculations of the Fast Marching Methods algorithm. Moreover, the Petrel platform provides state-of-the-art visualization tools that are used to display results, like diffusive time of flight and physical time of arrival. The visual presentation of results bolster the simplicity and intuitive appeal of the Fast Marching Methods approach.

This work includes an overview of the literature and previous work. Chapter II of the thesis discusses the concept of the radius of investigation and its generalization to heterogeneous reservoirs. It also summarizes the introduction and evolution of the theory of Fast Marching Methods. Chapter III provides details of the design and development of the FMM Plug-in, an overview of the general workflow. Chapter IV includes several applications and examples demonstrating the visualization of the drainage volume and an analysis of scalability and run-time performance. A user guide for the FMM Plug-in is included in Appendix A.

CHAPTER II

FAST MARCHING METHODS: METHODOLOGY*

This chapter summarizes the concept of depth of investigation in homogenous reservoirs, its generalization to heterogeneous reservoirs, and the introduction of the pressure front propagation equation. It also describes the Fast Marching Method (FMM), a front tracking method, as an efficient technique of solving the Eikonal equation, which is derived from an asymptotic solution of the diffusivity equation. The solution of the Fast Marching Method provides a 'diffusive time of flight' distribution that is used to estimate the drainage volume of the reservoir.

2.1 Depth of Investigation

The concept of radius of investigation and depth of investigation is important to well test design and analysis, which are widely used to estimate reserves and understand drainage volume, especially in unconventional reservoirs. Strong demand for natural gas and technological advances are turning unconventional reservoirs, like shale, into a key element of the energy supply in the United States and worldwide (Dong et al. 2013).

* Part of this chapter is reprinted with permission from "From Streamline to Fast Marching: Rapid Simulation and Performance Assessment of Shale Gas Reservoirs Using Diffusive Time of Flight as a Spatial Coordinate" by Zhang, Y., Bansal, N., Fujita, Y., Datta-Gupta, A., King, M.J., and Sankaran, S., 2014. Paper SPE 168997 presented at the SPE Unconventional Resource Conference, The Woodlands, Texas, 1-3 April. Copyright 2014 by SPE.

The idea of radius of investigation refers to the distance travelled into a reservoir formation by a pressure disturbance following a rate change in a well (Lee 1982). There are multiple definitions of the radius of investigation, however, they only vary in detail. Kuchuk (2009) compares and summarizes several of them. Most depend on analytical solutions and homogenous reservoir conditions. Hence, these solutions are limited for heterogeneous reservoirs with complex geology and particularly for unconventional reservoirs with multistage hydraulic fractures.

The most common definition that is widely used in the industry is given by Lee (1982). Lee defines the radius of investigation as the propagation distance of the ‘peak’ pressure disturbance for an impulse source or sink. The analytical solution for the radius of investigation (in field units) can be written as:

$$\mathrm{r} = \sqrt{\frac{kt}{984\,\phi\,\mu\,c_t}} \tag{2.1}$$

where *r* is the propagation distance and *t* is the propagation time. Generalizing the solution for different flow patterns:

$$\mathrm{r} = \sqrt{c\alpha t} \tag{2.2}$$

where *α* is the diffusivity, $\alpha = k/\phi\mu c_t$ (composed of *k*=permeability, φ=porosity, μ=fluid viscosity, c_t=total compressibility), and *c* is a geometric factor that depends on the specific flow geometry (Kim et al. 2009). For example, for linear flow, c=2, for radial flow, c=4, and for spherical flow, c=6. The generalization to heterogeneous reservoirs utilizes the idea of ‘diffusive’ time-of-flight (DTOF), τ, which relies on the observation that the ‘pressure fronts’ propagate with a velocity given by the square root of diffusivity (Datta-Gupta and King, 2007). DTOF has units of square root of time agreeing with the scaling behavior of pressure diffusion and can be related to physical time (Datta-Gupta et al. 2011):

$$\tau = \sqrt{ct} \tag{2.3}$$

However, τ is a measure of distance and is comparable to the radius of investigation. Putting equation (2.2) and equation (2.3) together:

$$\mathrm{r} = \sqrt{\alpha}\tau, \qquad and \ \frac{d\tau}{dr} = \frac{1}{\sqrt{\alpha}} \tag{2.4}$$

Equation (2.4) can be integrated along the trajectory to calculate DTOF for a one dimensional problem with reservoir heterogeneity (Xie et al. 2012b):

$$\tau(r) = \int_0^r \frac{1}{\sqrt{\alpha(r')}}\, dr' \tag{2.5}$$

which can be generalized for two or three dimensional problems as:

$$\sqrt{\alpha(x)}|\nabla\tau(x)| = 1 \tag{2.6}$$

Equation (2.5) and equation (2.6) describe the propagation of pressure front using the concept of depth of investigation with reservoir heterogeneity. Equation (2.6) is a form of the Eikonal equation and will be discussed later in the chapter.

2.2 Asymptotic Solution of Diffusivity Equation

The generalized “depth of investigation” is more suitable for reservoirs with heterogeneous conditions than the radius of investigation. The depth of investigation better accounts for the non-uniform pressure fronts that may be distorted due to heterogeneity effects. One method to obtain the depth of investigation uses a high-frequency asymptotic solution of the diffusivity equation for an impulse sink or source (Vasco et al. 2000). It relies on the equivalence between the propagating pressure front and a propagating wave front. The concept of propagating fronts is present in petroleum engineering and this method has also been applied in several fields like medical, optical, and electromagnetic imaging (Virieux et al. 1994, Datta-Gupta et al. 2001, Datta-Gupta and King 2007).

Starting with the known diffusivity equation for a single phase and slightly compressible fluid in a heterogeneous reservoir and constant fluid properties:

$$\phi(\mathbf{x})\mu c_t \frac{\delta P(\mathbf{x},t)}{\delta t} = \nabla \cdot (k(\mathbf{x})\nabla \mathrm{P}(\mathbf{x},\mathrm{t})) \tag{2.7}$$

where P(**x**,t) represents pressure, φ(**x**) porosity, k(**x**) permeability, μ viscosity, and c_t total compressibility. Then taking a Fourier transform of equation (2.7), we get the following equation in the frequency domain:

$$\phi(\mathbf{x})\mu c_t(-i\omega)\tilde{P}(\mathbf{x},\omega) = k(\mathbf{x})\nabla^2\tilde{P}(\mathbf{x},\omega) + \nabla k(\mathbf{x}) \cdot \nabla\tilde{P}(\mathbf{x},\omega) \tag{2.8}$$

Following previous work by Xie et al. (2012a, 2012b) and by considering a solution in terms of the inverse powers of $\sqrt{-i\omega}$, we get an asymptotic solution which is the pressure equation:

$$\tilde{P}(\mathbf{x},\omega) = e^{-\sqrt{-i\omega}\tau(x)} \sum_{k=0}^{\infty} \frac{A_k(\boldsymbol{x})}{(\sqrt{-i\omega})^k} \qquad (2.9)$$

where τ(**x**) is the propagation time of the pressure front (or diffusive time of flight) and A_k(**x**) is pressure amplitude. The high-frequency solution can be obtained from the initial terms of the asymptotic equation (Vasco and Datta-Gupta 1999, Vasco et al. 2000, Kulkarni et al. 2000):

$$\tilde{P}(\mathbf{x},\omega) = A_0(\boldsymbol{x})e^{-\sqrt{-i\omega}\tau(x)} \qquad (2.10)$$

Combining equation (2.10) with equation (2.8) and collecting terms with highest orders (-iω) gives the equation for pressure propagating front, which is equation (2.6) as seen in the previous section:

$$\sqrt{\alpha(x)}|\nabla\tau(x)| = 1, \quad \text{and} \;\; \alpha(\mathbf{x}) = \frac{k(\mathbf{x})}{\phi(\mathbf{x})\mu c_t}$$

which is a form of the Eikonal equation. The pressure front propagation is independent of the flow rate but it depends on reservoir and fluid properties. The square root of fluid diffusivity, which contains the reservoir heterogeneity information, gives the velocity of the pressure front propagation.

2.3 Drainage Volume Calculation Using Fast Marching Method

Drainage volume can be estimated by calculating the reservoir volume affected by the pressure disturbance from the impulse at the sink or source, which is contained by the ‘pressure front’ at a given time. To determine the pressure front, we solve the Eikonal equation (2.6) which governs the pressure front propagation in the reservoir. The Fast

Marching Method (FMM), which is a class of level set methods that track the position of propagating fronts, can solve the Eikonal equation very efficiently (Sethian 1996, 1999). The solution of the Eikonal equation yields the diffusive time of flight. The approach of FMM is similar to Dijkstra's algorithm which solves the shortest path between two points on a graph (Dijkstra 1959).

The Fast Marching Method follows this procedure (Sethian 1996, 1999; Xie et al. 2012a, 2012b; Zhang et al. 2013):

(1) Label the nodes corresponding to the initial position of the propagating front (e.g. well location) as *accepted*, with $\tau = 0$. Label all other grid nodes as *unknown*, with $\tau = \infty$.

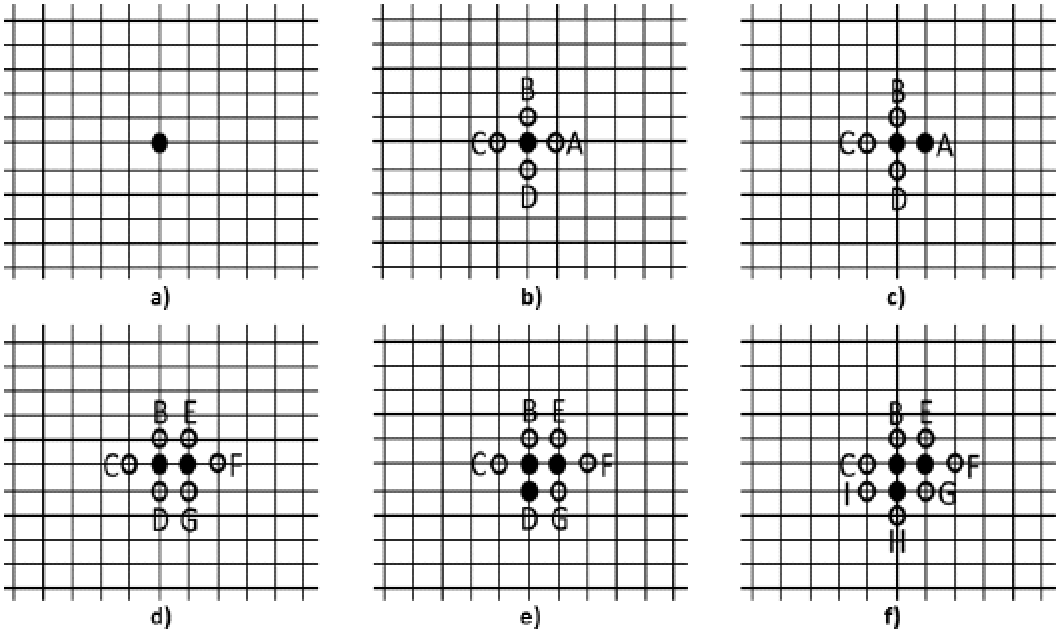
(2) For each node that is *accepted*, locate its immediate neighboring nodes that are *unknown* and label them as *considered*.

(3) For each node labeled *considered*, update its τ based on its *accepted* neighbors using the minimum of local solutions of equation (2.6).

(4) Once all nodes labeled *considered* have been locally updated, the node which has the minimum τ among them is picked and labeled as *accepted*.

(5) Go to step (2) until all nodes are *accepted*.

Xie et al. (2012a, 2012b) illustrate these steps of the Fast Marching Method on a 2D rectangular grid, as shown in **Figure 2.1**. In this illustration, one point is selected as the initial position of the propagating front and is labeled as *accepted* (represented by a solid black circle as in figure 2.1a). Then its immediate neighbors A, B, C, and D are labeled as *considered* (represented by hollow circles as in figure 2.1b). The τ values of A, B, C, and D are updated and the smallest one (suppose it is A) is selected and labeled as *accepted*, as shown in figure 2.1c. Once a point is labeled as *accepted*, its neighbors (E, G, and F, in this example) are labeled as *considered*, as shown in figure 2.1d. The previous steps are repeated for the next *accepted* point (suppose it is D) as shown in figure 2.1e and 2.1f.



a) b) c) d) e) f)

Figure 2.1: Illustration of Fast Marching Method (Reprinted with permission from Xie et al. 2012).

The τ gradient is calculated by using a finite difference discretization of the Eikonal equation on a 2D grid (Sethian 1996):

$$\max\left(\mathrm{D}_{ij}^{-x}\tau, -\mathrm{D}_{ij}^{+x}\tau, 0\right)^2 + \max\left(\mathrm{D}_{ij}^{-y}\tau, -\mathrm{D}_{ij}^{+y}\tau, 0\right)^2 = \frac{1}{\alpha} \tag{2.11}$$

where *D* is a finite difference operator with first-order upwind approximation scheme. In the x-direction, *D* is expanded as:

$$\mathrm{D}_{ij}^{-x}\tau = \frac{(\tau_{i,j} - \tau_{i-1,j})}{\Delta x} \tag{2.12a}$$

$$\mathrm{D}_{ij}^{+x}\tau = \frac{(\tau_{i+1,j} - \tau_{i,j})}{\Delta x} \tag{2.12b}$$

and in the y-direction, *D* is expanded as:

$$\mathrm{D}_{ij}^{-y}\tau = \frac{(\tau_{i,j} - \tau_{i,j-1})}{\Delta y} \tag{2.12c}$$

$$\mathrm{D}_{ij}^{+y}\tau = \frac{(\tau_{i,j+1} - \tau_{i,j})}{\Delta y} \tag{2.12d}$$

Equation (2.11) and equation (2.12) produce a quadratic expression for the diffusive time of flight, τ, which can be solved very efficiently. After estimating the arrival time at each node, $\tau_{i,j}$, the drainage volume at any time can be calculated by summing up the pore volume of all nodes contained within the contour of the pressure front at that given time.

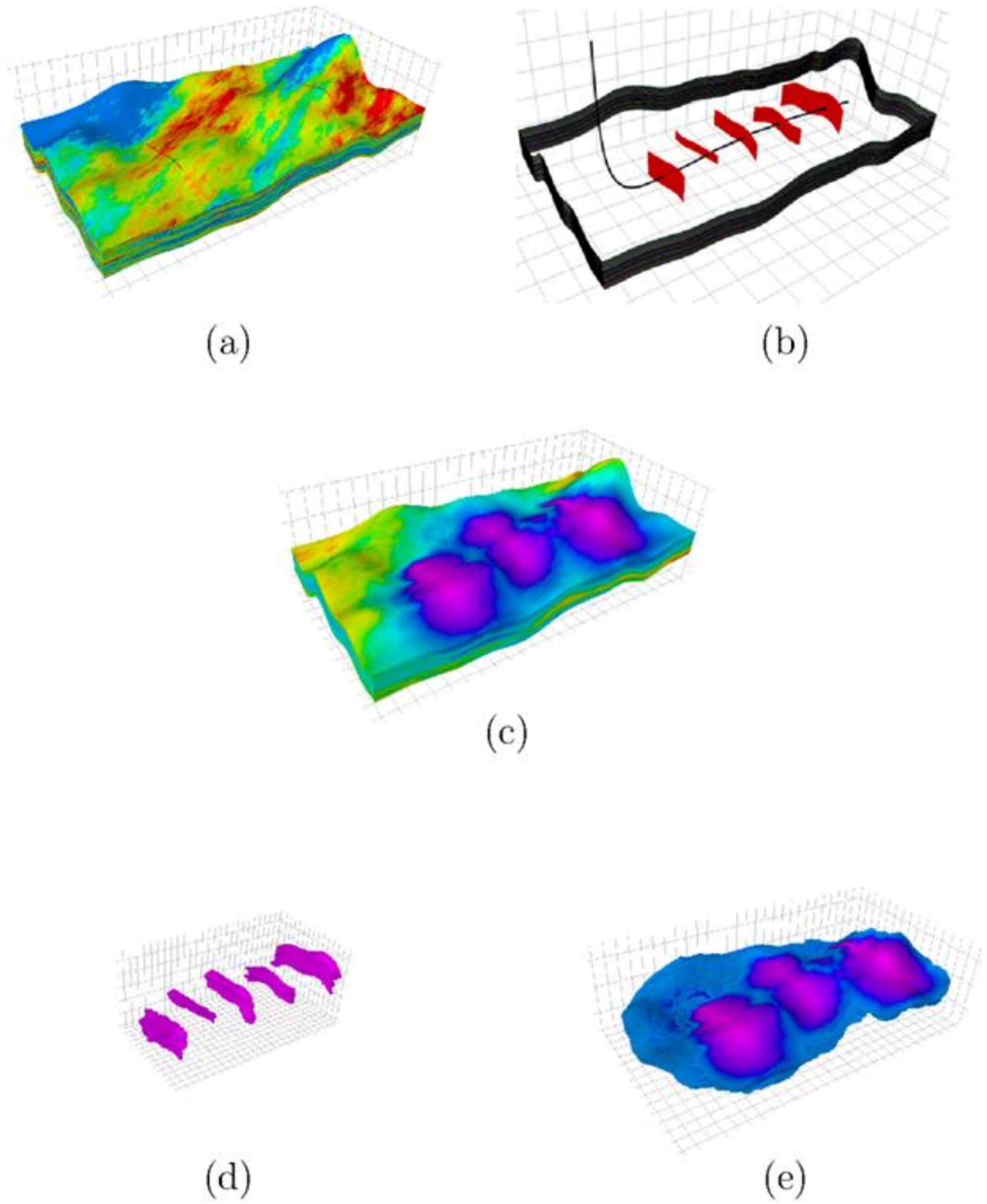


Figure 2.2: A synthetic example of a heterogeneous reservoir with 5 hydraulic fractures. (a) permeability field; (b) the geometry of five hydraulic fractures; (c) calculated DTOF; (d) drainage volume in 1 month; (e) drainage volume in 30 years. (Reprinted with permission from Zhang et al. 2013)

Zhang et al. (2013) extended the Fast Marching Method to complex grids including corner point and unstructured grids. They demonstrated the efficiency of Fast Marching Method in a variety of applications to unconventional reservoirs. The results for drainage volume calculations for one of those applications is shown in **Figure 2.2** (Zhang et al. 2013). The reservoir model shown in the figure below has matrix permeability between 1 nanodarcy and 0.001 millidarcy. It contains 400×200×30 cells with dimension of 4000ft×2000ft×120ft. It has a horizontal well with 5 hydraulic fractures. Figure 2.2c shows the calculated DTOF using the Fast Marching Method. The drainage volume can be visualized for any given time by transforming the DTOF to physical time. The drainage volume is shown for 1 month and 30 years in figure 2.2d and 2.2e, respectively.

2.4 Geometric Approximation Using Drainage Volume

The drainage volume obtained by the Fast Marching Method can be utilized to estimate the pressure and production behavior of a reservoir system. The prediction of the pressure and production behavior is achieved by geometric approximation. With a constant well rate, the pressure dropdown can be well-approximated. Similarly, well rates can be computed under constant bottom-hole flowing pressure (Xie et al. 2012a, 2012b).

2.4.1 Pressure Solution Using Geometric Approximation

After calculating the drainage volume as a function of time, *Vp(r(t)),* the bottom-hole flowing pressure and the reservoir average pressure within the pressure front can be approximated. Xie et al. (2012a) derive the approximation starting with a mixed form of the diffusivity equation and the Darcy flux, Q, for radially symmetric flow.

$$A(r)\phi c_t \frac{\delta P}{\delta t} = \frac{\delta Q}{\delta r} \tag{2.13}$$

$$Q = \frac{kA(r)}{\mu}\frac{\delta P}{\delta r} \tag{2.14}$$

where $A = 2\pi r h$, $r = \sqrt{x^2 + y^2}$ for cylindrical radial flow,

$A = 4\pi r^2$, $r = \sqrt{x^2 + y^2 + z^2}$ for spherical radial flow,

and *A* is a constant, $r = |x|$ for linear flow.

The sign convention being used in this derivation is that Q is inwardly directed flux and Q_w is positive for a producer. Equation (2.14) is rewritten in terms of pore volume, V_p:

$$c_t \frac{\delta P}{\delta t} = \frac{\delta Q}{\delta V_p} \tag{2.15}$$

$$V_p(r) = \phi \int_0^r A(r) dr \tag{2.16}$$

The geometric approximation for the pressure solution assumes that the well is producing at a constant rate, Q_w, that the Darcy flux is negligible outside the drainage volume, and that the pressure within the drainage volume is well approximated by a steady-state solution. Using the assumptions above, equation (2.15) yields the pseudo-steady state solution (Nordbotten et al. 2004, Agarwal 2010, Xie et al. 2012a):

$$\frac{\delta P}{\delta t} \cong \frac{\delta \bar{P}}{\delta t} = -\frac{1}{c_t} \frac{Q_w}{V_p(r(t))} \tag{2.17}$$

The well test derivative can then be calculated from equation (2.17):

$$\Delta P' = -t \frac{\delta P}{\delta t} \cong -t \frac{\delta \bar{P}}{\delta t} = \frac{1}{c_t} \frac{Q_w t}{V_p(r(t))} \tag{2.18}$$

Approximate reservoir pressure distribution and bottom-hole pressure can be obtained by integrating equation (2.17). Xie et al. (2012a, 2012b) have demonstrated several examples and applications of equation (2.17) and equation (2.18).

2.4.2 Rate Calculation Using Geometric Approximation

Well rates can be obtained by geometric approximation using the drainage volume. Xie et al. (2012b) relate well production rate to calculated drainage volume from the Fast Marching Method. The approach assumes the well is producing under constant well bottom-hole flowing pressure. The reservoir is assumed to be under pseudo-steady state conditions inside the drainage volume and at initial conditions outside of the drainage volume. Following the formulation of Xie et al. (2012b) and starting with Darcy's law in radial flow:

$$Q(r,t) = \frac{kA(r)}{\mu}\frac{\delta p(r,t)}{\delta r} \tag{2.19}$$

To express Darcy's law in terms of the drainage volume, where the drainage volume would be used as a spatial coordinate, the following chain rule is used:

$$dV_p = \phi A(r) \cdot \mathrm{dr}$$

$$\frac{\delta}{\delta r} = \frac{dV_p}{dr} \cdot \frac{\delta}{\delta V_p} = \phi A(r)\frac{\delta}{\delta V_p} \tag{2.20}$$

Combining equation (2.20) with equation (2.19) yields:

$$Q(V_p,t) = \frac{k\phi A^2}{\mu}\frac{\delta p(V_p,t)}{\delta V_p} \tag{2.21}$$

where the Darcy flux and pressure are in terms of reservoir drainage volume and both the permeability and porosity are averaged over the surface area, A, along the boundary of the drainage volume. The Darcy flux and the production rate can be related using a dimensionless flux along the drainage volume as follows (Winestock and Colpitts 1965, Xie et al. 2012b):

$$Q\left(V_p,t\right) \cong Q_{well}(t) \cdot Q_D(V_p,t) \tag{2.22}$$

Putting equation (2.21) and equation (2.22) together, and then integrating:

$$Q_{well}(t) \cdot Q_D\left(V_p,t\right) \cong \frac{k\phi A^2}{\mu}\frac{\delta p(V_p,t)}{\delta V_p} \tag{2.23}$$

$$\Delta p = p_i - p_{wf} \cong \ Q_{well}(t)\mu \int_0^{V_p(t)} \frac{Q_D(V,t)}{k\phi A^2} dV \tag{2.24}$$

The integral can be restricted to the depth of investigation because the Darcy flux is negligible outside the drainage volume. The pressure drop, Δp, is considered to be specified since the well is assumed to be at a constant bottom-hole flowing pressure. Under pseudo-steady state condition (inside the drainage volume), the diffusivity equation can be rewritten in terms of the drainage volume and the dimensionless flux becomes a linear function of the drainage volume:

$$c_t \frac{\delta p}{\delta t} = \frac{\delta q}{\delta V_p} = Q_{well} \frac{\delta Q_D(V_p)}{\delta V_p} \tag{2.25}$$

Setting the Darcy flux equal to production rate at the well and equal to zero at the depth of investigation gives the following formula for the dimensionless flux:

$$Q_D(V) = 1 - \frac{V}{V_p(t)} \tag{2.26}$$

Substituting equation (2.26) into equation (2.24) and rearranging:

$$Q_{well}(t) \cong \ \frac{\Delta p}{\mu} \frac{1}{\int_0^{V_p(t)} \frac{1}{k\phi A^2} (1 - \frac{V}{V_p(t)}) dV} \tag{2.27}$$

where the drainage volume, *V_p(t)*, is calculated using the Fast Marching Method. The production rate is at reservoir conditions and can be converted to standard conditions using the oil or gas formation volume factor.

2.5 Coordinate Transformation

Zhang et al. (2014) have proposed the use of the DTOF as a spatial coordinate to transform the diffusivity equation from 3-D space to 1-D space. Transforming the diffusivity equation into a one-dimensional equation allows the spatial variables, like pressure, to be solved along the new DTOF coordinate, τ, using the finite-difference scheme. To achieve the coordinate transformation, Zhang et al. (2014) assume that the pressure gradient direction is aligned with the τ gradient direction. This is equivalent to the assumption that the contour surfaces of pressure are the same as the contour surfaces of τ or that the pressure only depends on τ in space. We incorporate the Eikonal equation (2.6) with this assumption in equation (2.28).

$$\vec{\nabla} P \approx \frac{\delta P}{\delta \tau}\vec{\nabla}\tau = \frac{\delta P}{\delta \tau}\frac{1}{\sqrt{\alpha}}\,\hat{\boldsymbol{n}}_{\tau} \qquad (2.28)$$

where $\hat{\boldsymbol{n}}_{\tau}$ is the unit normal vector to the contour of τ. Equation (2.29) shows the 3-D diffusivity equation after introducing the diffusivity term. We then substitute equation (2.28) into equation (2.29):

$$\phi c_t \frac{\delta P}{\delta t} = \nabla \cdot (\phi c_t \alpha \vec{\nabla} \mathrm{P}) \qquad (2.29)$$

$$\phi c_t \frac{\delta P}{\delta t} = \nabla \cdot (\phi c_t \sqrt{\alpha}\frac{\delta P}{\delta \tau}\,\hat{\boldsymbol{n}}_{\tau}) \qquad (2.30)$$

To perform the coordinate transformation for equation (2.30), we need to rewrite the divergence operator from (x, y, z) to (τ, ψ, κ). Here, τ is the DTOF and ψ and κ are two orthogonal coordinates that are defined on the contour surfaces of the DTOF. In the new system the divergence operator is written as:

$$\nabla \cdot \vec{\mathrm{F}} = \frac{1}{h_\tau h_\psi h_\kappa}(\frac{\delta\left(h_\psi h_\kappa F_\tau\right)}{\delta\tau} + \frac{\delta\left(h_\tau h_\kappa F_\psi\right)}{\delta\psi} + \frac{\delta\left(h_\tau h_\psi F_\kappa\right)}{\delta\kappa}) \qquad (2.31)$$

where h_τ, h_ψ, and h_κ are the length of the covariant vectors. Expanding the divergence operator in equation (2.30), we have:

$$\phi c_t \frac{\delta P}{\delta t} = \frac{1}{h_\tau h_\psi h_\kappa}\frac{\delta}{\delta\tau}(\phi c_t h_\psi h_\kappa \sqrt{\alpha}\frac{\delta P}{\delta\tau}) \qquad (2.32)$$

Terms in ψ and κ coordinates are not present in the equation because only the τ direction normal component was present in the vector function in the divergence. Since h_τ is the length of the covariant vector $\vec{\boldsymbol{t}}_\tau$, we have $h_\tau = \left\|\vec{\boldsymbol{t}}_\tau\right\| = \left\|\frac{\delta\vec{X}}{\delta\tau}\right\| = \sqrt{\alpha}$. The resulting product, $h_\tau h_\psi h_\kappa$, represents the Jacobian of the coordinate transformation. Using these definitions, we can reduce equation (2.32) to:

$$J\phi c_t \frac{\delta P}{\delta t} = \frac{\delta}{\delta\tau}(J\phi c_t \frac{\delta P}{\delta\tau}) \qquad (2.33)$$

While the pressure P, based on our assumption, is only a function of t and τ, the Jacobian J and porosity φ are both functions of (τ, ψ, κ). This allows us to further reduce equation (2.33) by integrating over ψ and κ coordinates, since P can be taken out of the integral.

$$w(\tau)\frac{\delta P}{\delta t} = \frac{\delta}{\delta\tau}(w(\tau)\frac{\delta P}{\delta\tau}) \qquad (2.34)$$

where $(\tau) = \int\int \phi J(\tau,\psi,\kappa)d\psi d\kappa = \frac{dV_p(\tau)}{d\tau}$. Equations (2.33) and (2.34) give a 1-D diffusivity equation where the heterogeneity is fully embedded in the τ-coordinate. Using a finite difference scheme, the 1-D equation can be solved to obtain the drainage volume and pressure depletion in the reservoir. This approach proposed by Zhang et al. (2014) allows for general and rapid numerical simulation along the DTOF as a spatial coordinate.

2.6 Synthetic Example

This section presents a synthetic case of a heterogeneous reservoir model. Using this example, the results of Fast Marching Method could be examined in contrast with results from a commercial finite difference simulator. The simulator of choice for this example is ECLIPSE from Schlumberger, an industry-reference finite difference reservoir simulator. To demonstrate the comparison, reservoir production is simulated at constant bottom-hole pressure for nearly two years using ECLIPSE and Fast Marching Method.

Both simulators use the same reservoir properties and model parameters, which are shown in Table 2.1. The synthetic gas reservoir model has a horizontal well with three stage hydraulic fractures, which have enhanced permeability of 10 millidarcy. The heterogeneous matrix permeability of the reservoir ranges between 1 nanodarcy and 0.1 millidarcy (Figure 2.3).

Table 2.1: Properties for synthetic horizontal well with three hydraulic fractures

Property	Value
Reservoir dimensions (ft)	1000×1500×25
Grid size	100×150×5
Fracture half-length (ft)	300, 450, 350
Reservoir porosity	0.04
Gas viscosity (cp)	0.02
Initial reservoir pressure (psi)	3000
Fracture permeability (mD)	10
Total compressibility (psi^{-1})	1×10^{-5}
BHP control (psi)	1500

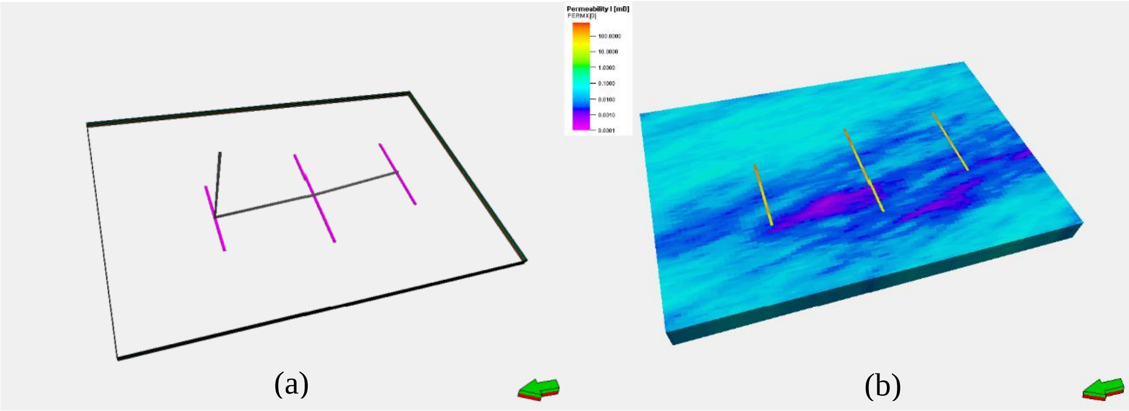


Figure 2.3: A synthetic example of a heterogeneous reservoir. (a) geometry of three hydraulic fractures; (b) permeability field.

The results of the simulation are shown in Figure 2.4 below. By comparing the results, one can observe good agreement between the 1D solution in the Fast Marching Method and the finite difference solution from ECLIPSE simulator.

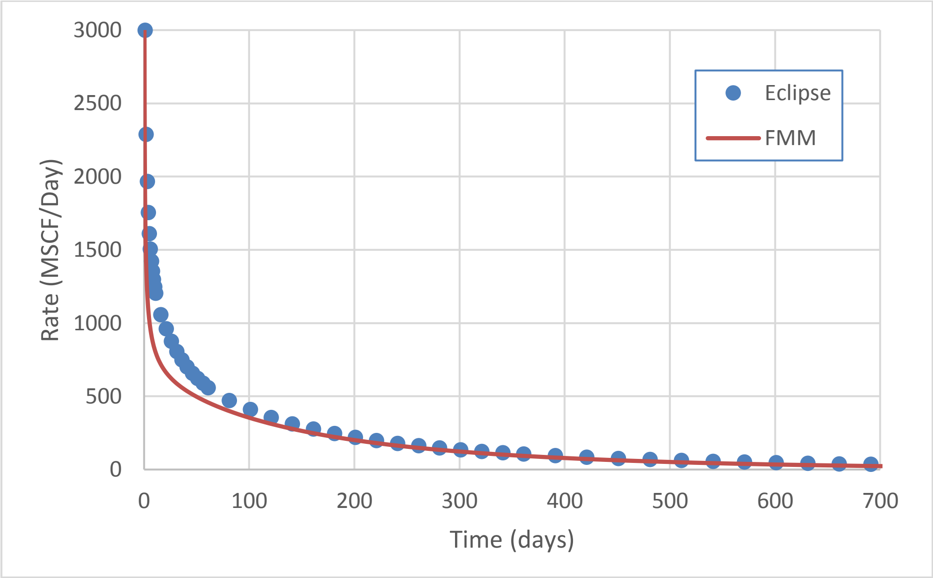


Figure 2.4: Comparison of production rate calculated by FMM and ECLIPSE

CHAPTER III

FAST MARCHING METHOD PLUG-IN

The optimization of production performance in unconventional reservoirs calls for an understanding of the drainage volume and reliable estimates of reservoir properties. The novel approach of the Fast Marching Method evaluates the drainage volume and well performance and is especially well suited for shale reservoirs. With Fast Marching Method, we can quickly generalize the drainage volume and account for complex reservoir heterogeneity, geologic stratigraphy, and multi-stage fractured wells. As discussed in the previous chapter, the asymptotic expansion of the diffusivity equation produces the Eikonal equation that describes the propagation of the pressure front. Fast Marching Method can solve the Eikonal equation efficiently without resorting to detailed fluid flow simulation. The pressure front propagation is obtained from a single non-iterative calculation. Thus, the time evolution of the drainage volume for multimillion-cell geologic models can be computed and visualized in seconds.

There is a wide range of uses and potential applications for Fast Marching Method including well multi-stage fracture spacing optimization, well trajectory and hydraulic fracture location optimization, reservoir model screening and calibration of reservoir parameters, uncertainty analysis, and production data integration (Zhang et al. 2013). Development of Fast Marching Method software can utilize the many benefits of this novel approach and enable its consolidation with existing engineering workflows. This chapter discusses the design and development of the software, which has been built as an application plug-in, and its integration with a common commercial E&P software platform.

The first component of the application plug-in, which is named **FMM Processor**, implements the concepts described earlier in chapter II and is responsible for computing the diffusive time of flight, and running the rate and pressure calculations. Although FMM Processor was built as a separate application using C++ programming language, it still operates as a computing process within the Fast Marching Method Plug-in (**FMM Plug-in**). FMM Processor is incapable of preparing and obtaining input data in a useful format and had previously relied on commercial simulators, like ECLIPSE from Schlumberger, to generate its formatted data input files. Moreover, FMM Processor lacks the means to present and visualize its results. The absence of pre-processing and post-processing capabilities in the FMM Processor renders it short of fully utilizing the Fast Marching Method. Therefore, the main motive behind the development of the FMM Plug-in is to overcome the shortfall in pre-processing and post-processing capabilities.

The FMM Plug-in has been integrated into the Petrel Platform from Schlumberger, which is one of the most commonly used E&P software platforms in the industry. The Petrel platform contains a wide set of features, including state-of-the-art visualization tools, that greatly complements the FMM Plug-in. The Petrel platform is also coupled with the Ocean software development framework from Schlumberger that enables its features and capabilities to be extended and developed in the form of plug-ins.

3.1 Introduction to Petrel E&P Software Platform

Petrel is a Schlumberger owned E&P software platform that operates on Microsoft Windows. It allows effortless collaboration between engineers and geoscientists and features a single, model-centric subsurface representation built from multidiscipline knowledge (Schlumberger 2014a). More and more companies are working with increasingly complex and unconventional reservoirs quickly raising the need for advanced and comprehensive software solutions. The Petrel platform brings an integrated solution from exploration to production with its seismic-to-simulation software, which enables geoscientists and engineers to create and simulate complex 3D models of oil and gas reservoirs (Wharton 2010).

The broad set of tools for planning and evaluation of different development scenarios and field management, along with the powerful simulation environment for all types of simulation studies, makes the Petrel platform popular amongst thousands of geoscientists and engineers worldwide (EAGE Daily News 2013, 2014). Schlumberger advertises its use of the most extensive array of algorithms in the industry in property modeling (Schlumberger 2014b). The Petrel reservoir engineering module also features multiple upscaling methods, quick plots, interactive 3D analysis tools, and unmatched state-of-the-art visualization tools (Akram et al. 2007, Schlumberger 2014a). It is especially well suited for shale and heavy oil reservoirs, that may require high counts of wells with complex well trajectories, due to its well-pad-design optimization capabilities (EAGE Daily News 2012). Figure 3.1 shows the user interface of the Petrel platform.

These powerful sets of tools make the Petrel platform an unrivalled candidate host to complement and efficiently utilize the Fast Marching Method. Moreover, the Petrel platform environment allows for seamless integration. For example, Schlumberger's industry-reference reservoir simulators, like ECLIPSE and INTERSECT, are all supported within the Petrel software platform. In addition, the Ocean software development framework enables the customization of Petrel features and gives software developers the ability to extend the existing functionality of the platform. The Ocean framework makes it feasible to extend and combine Petrel tools with Fast Marching Method software in a single integrated workflow within the Petrel platform.

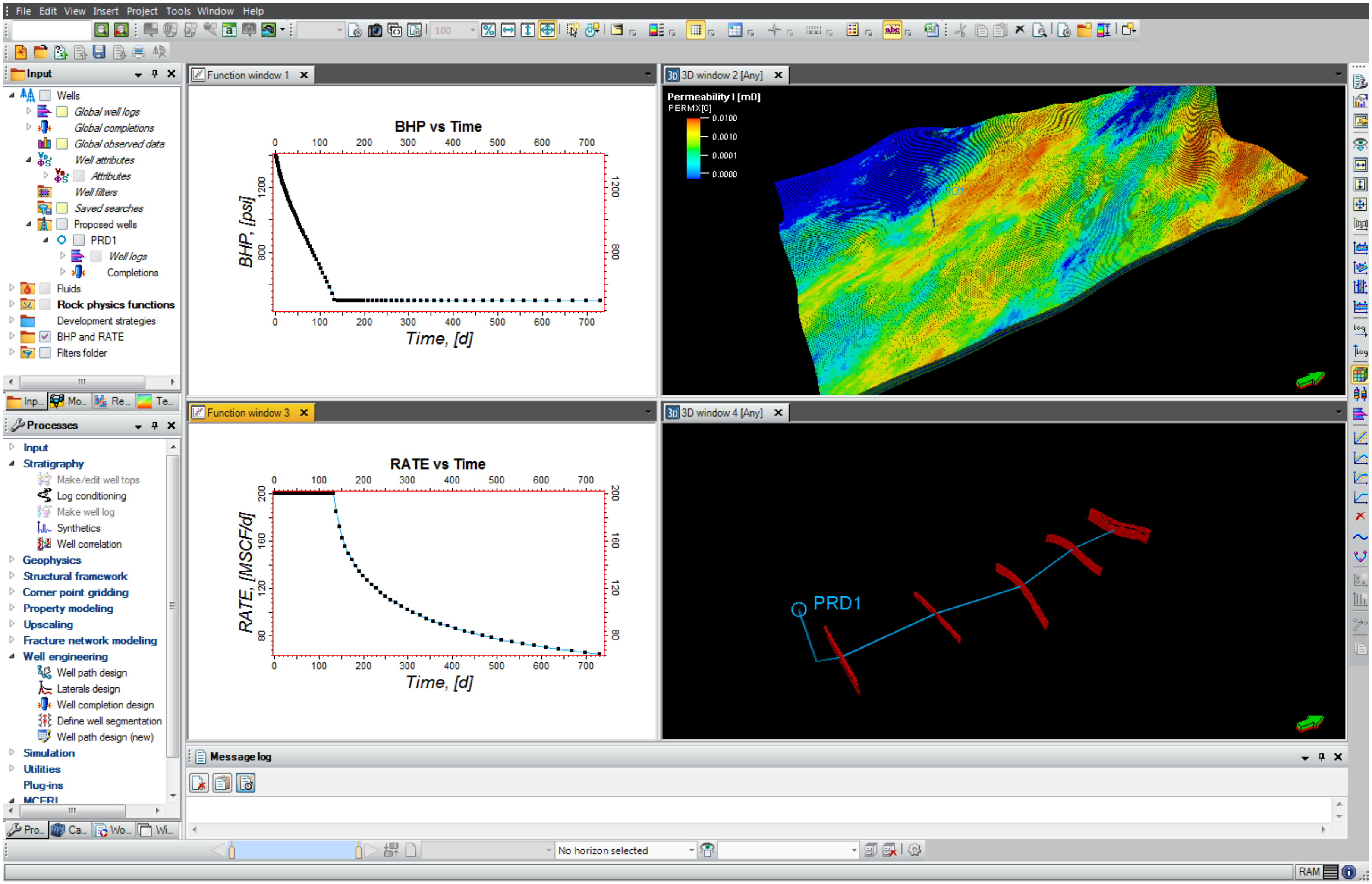


Figure 3.1: Petrel user interface

3.2 Ocean Software Development Framework

Ocean is a Schlumberger owned software development framework built in the Windows.NET environment. It unlocks elements and components of the Petrel reservoir engineering and production engineering modules to allow software developers to expand its capabilities with new features and workflows (Schlumberger 2014c). Being built in the .NET environment, the Ocean framework provides a more productive and friendlier environment for software developers due to the support of the Microsoft Developer Network libraries (MSDN). While the 'Ocean for Petrel' framework was released nearly eight years ago, Schlumberger is expanding the Ocean framework to other Schlumberger software products, like Techlog and Avocet platforms, by 2015.

E&P companies and researchers in academia continue to face new challenges that require quick innovative solutions. In the past, commercial applications had often lagged behind the industry's requirements and many energy companies developed hard-to-integrate and maintain software. What the industry needed was the ability to innovate and develop customized software solutions on the fly without having to wait for the commercial software vendors to catch up. Wharton (2010) argues that the Ocean framework brings exactly that advantage. He asserts that the Ocean framework facilitates accelerated development of innovative software solutions and allows immediate deployment on top of currently released commercial applications. As of 2010, the Ocean framework had been adapted by 22 oil and gas companies, 30 software development companies, and 19 universities around the world, which translated in a noticeable rise in the number of Ocean plug-ins developed (shown in Figure 3.2; Wharton 2010).

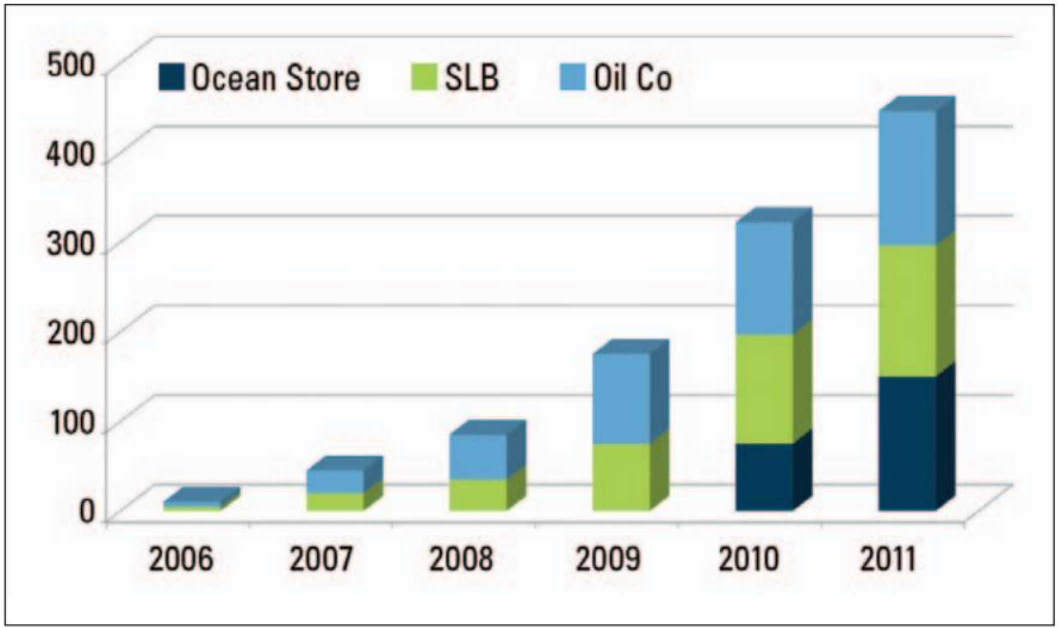


Figure 3.2: Significant growth in number of Ocean plug-ins (Reprinted with permission from Wharton 2010)

The Ocean framework offers a large set of application programming interfaces (APIs) that give programming access to different components inside the Petrel platform. Even though later sections of this chapter detail the components used in FMM Plug-in, we highlight some of the important Ocean framework APIs here. The fluid API allows programmatic creation of Petrel black-oil fluid models and compositional fluid models. Similarly, several rock physics functions can be created and modified through code, including saturation, compaction, and adsorption functions. Perhaps one of the most used sets of the Ocean framework APIs in FMM Plug-in is to access grids. The API enables grid creation and modification with full access to grid cell properties, allowing plug-ins to create, read, update, and delete cell properties. Most importantly, the Ocean framework APIs enable plug-ins to access Petrel visualization and charting tools, allowing the use of its different window renderers (i.e. 2D, 3D, function window, charting, map, …etc). The 2014 release of Ocean framework also included development strategy API with new functionality allowing manipulation of wells, groups, rules, control date and other elements within the Petrel development strategy domain. However, most of the development strategy functionality was unavailable in the 2013 release of the Ocean framework that was used for building the FMM Plug-in.

3.2.1 The Architecture of the Ocean Framework

The Ocean framework is built on top of the Microsoft .NET environment and is arranged into three API layers (Schlumberger 2013). The Ocean Core API is built directly on the .NET framework and serves as the infrastructure of the Ocean framework. The Ocean Services API is built on top of Ocean Core and is dependent only on the Ocean Core for its functionality, which makes it independent of what product family (like Petrel) it is being used for. Finally, the Product Family API layer (Ocean for Petrel, in this case), is built at the top-level of the Ocean framework as shown in Figure 3.3 below. Any module built in Ocean, like the module of FMM Plug-in, has access to the functionalities of all three components of the Ocean framework.

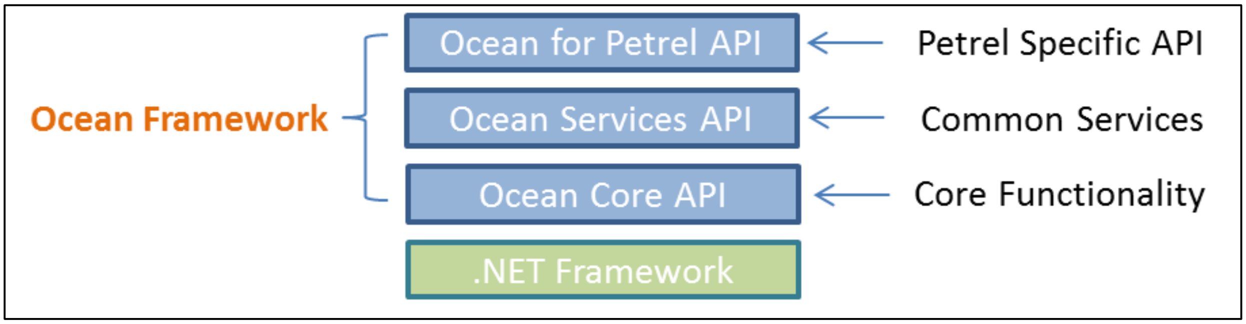


Figure 3.3: Architecture of the Ocean Framework

The Ocean Core serves as the infrastructure of the framework and manages all modules of the Ocean plug-ins. It handles the lifecycle of plug-in modules like initialization, integration into the Petrel system during startup, and later disintegration and removal at the end of the cycle. The Core also controls the transfer of information from data sources to different modules and provides other lower level functionality to the rest of the Ocean API.

The Ocean Services are a group of APIs that include a standardized common set of functionalities. Built on top of the Ocean Core, the services are thus application-independent and can operate on all product family applications of Schlumberger. These standardized services include API for unit systems and unit conversion, coordinate systems and coordinate conversion, general data types, specialized geometry data types, and others.

Ocean for Petrel is the product family API layer for the Petrel platform. It provides access to different domain objects within Petrel (like wells, completions, well logs, grids, faults, horizons, etc), graphical tools (like visualizations and charting window renderers), and Petrel user interface (like toolbars, menus, windows, etc). It allows for seamless integration with the Petrel interface for optimal user experience.

Ocean plug-ins are containers of a set of one or more modules, which include all the implemented plug-in functionality. The plug-in is used as an organizational entity giving the application an identity and is also responsible for deployment tasks (like installation and distribution). On the other hand, the module is the basic component containing the plug-in functionality and processes. The module also has access to all Ocean API layers, as illustrated in Figure 3.4.

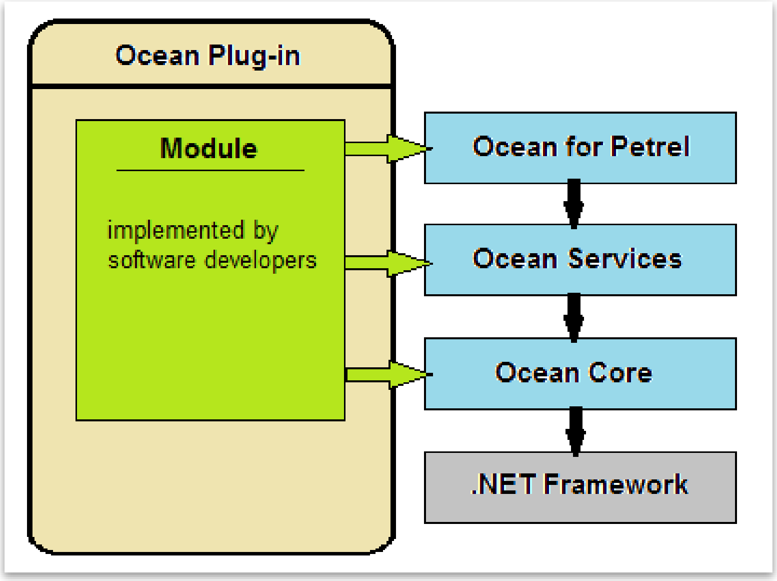


Figure 3.4: Plug-in Modules can access all Ocean API layers

3.2.2 Data Access via Ocean API

Recall that one of the main objectives of the Ocean framework is to allow access to components of the Petrel platform and enable extension of its features and usability. Achievement of these goals usually mandates access to Petrel's extensive sets of data (wells, well logs, seismic data, grids properties, faults, horizons, etc). Although this information might be stored in a complicated and sophisticated system within the Petrel platform, Ocean provides API that allows data access at a high-level of abstraction, as illustrated in Figure 3.5. Ocean uses the concept of *Domain Objects* to encapsulate the Petrel validation and usage rules and hide the complexity of data storage (Schlumberger 2013). Domain Objects expose Petrel data. However, the API is constrained by Petrel and Ocean does not offer unlimited data access to Petrel data.

With Ocean, Petrel data is grouped into separate domains designated by parent Domain objects including *PillarGrid* (which gives access to grids, grid geometries, grid properties …), *Well* (gives access to boreholes, markers, casings, completions, …), Shapes (points, polylines, surfaces, fracture networks, …), and many others.

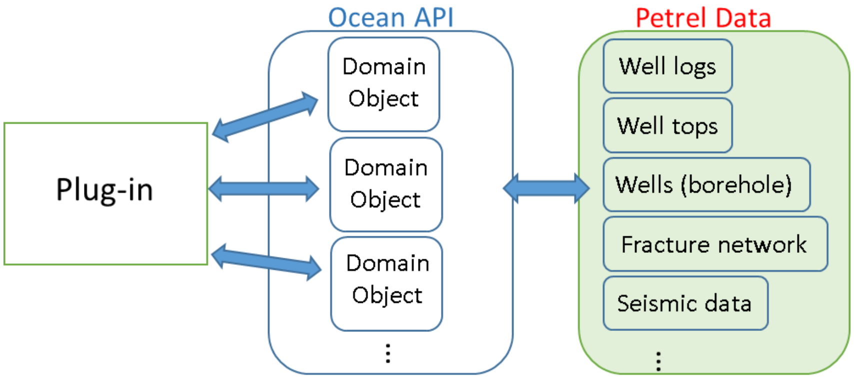


Figure 3.5: Access to Petrel data via Domain Objects

3.3 Fast Marching Method Plug-in Workflow

The general workflow of the FMM Plug-in is divided into three stages. The first is the pre-processing stage which includes preparing the data required for the following computations. Next, the Fast Marching Method algorithm is executed. Finally, at the post-processing stage, the result data is imported and put into suitable format and is then presented using available visualization tools. A flowchart illustrating the general workflow of the FMM Plug-in is shown in Figure 3.6.

There are two allowed options for calculations in the FMM Plug-in. The first is to only calculate the drainage volume for a selected reservoir model. Selecting this option provides the diffusive time of flight and the physical arrival time at each grid block, allowing for visual observation of the time-evolution of the drainage volume in the selected reservoir model. The second option is to calculate the diffusive time of flight and use it in a 3D to 1D coordinate transformation to calculate bottom-hole pressure or well rate under pre-specified conditions. However, performing a 1D simulation requires the user to provide additional input parameters (discussed in later sections). In both options, the drainage volume is calculated for a selected reservoir model by solving the Eikonal equation discussed in chapter II (equation 2.6):

$$\sqrt{\alpha(x)}|\nabla\tau(x)| = 1 , \quad \text{and } \alpha(\mathbf{x}) = \frac{k(\mathbf{x})}{\phi(\mathbf{x})\mu c_t}$$

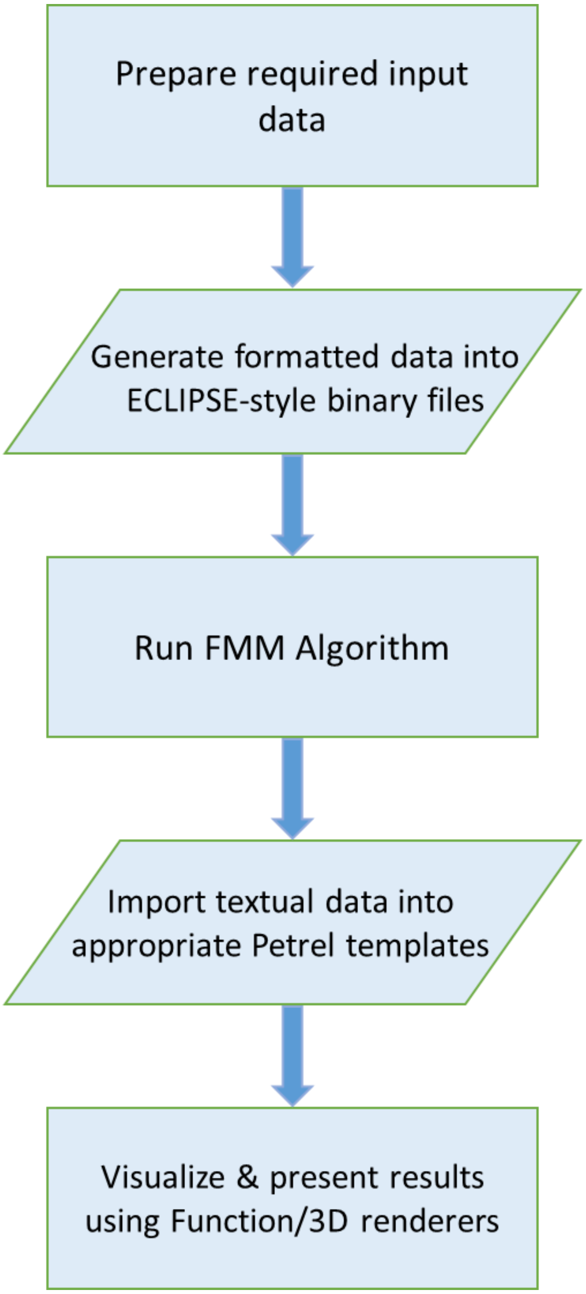

Prepare required input data
Generate formatted data into ECLIPSE-style binary files
Run FMM Algorithm
Import textual data into appropriate Petrel templates
Visualize & present results using Function/3D renderers

Figure 3.6: FMM Plug-in workflow

To solve for the diffusive time of flight using the Eikonal equation, diffusivity information and geometrical information must be provided. Following the above definition of the diffusivity, the solution requires providing reservoir permeability, porosity, total compressibility, and fluid viscosity. The geometrical information needed includes the reservoir grid geometry (Cartesian or corner-point) and well completion data (location of completed grid cells and direction of the completion inside the cell).

3.3.1 Preparation of Required Input Data

Reservoir information of a Petrel case could be stored over different domains inside Petrel. During the preparation stage of the required input data, information is obtained from the appropriate Petrel domain objects. However, all Petrel data that is obtained through Ocean is stored in the SI metric unit system and thus has to be implicitly or explicitly converted to the appropriate unit system (the current version of FMM Plug-in converts data to field units by default).

The PillarGrid group (also known as the PillarGrid *namespace* in object-oriented programming terminology) contains several domain objects that provide most of the input data used in the Fast Marching Methods algorithm. The *Grid* domain object provides the geometrical information later used in discretizing the Eikonal equation when calculating the diffusive time of flight. The FMM Plug-in retrieves IJK information of the grid, dimensions, and XYZ coordinates of all eight corners for each grid cell and converts all distance units to feet. Additionally, four *Grid Property* domain objects from the PillarGrid group are retrieved and later used to calculate the diffusivity for each grid cell, namely, reservoir porosity, I-direction permeability, J-direction permeability, and K-direction permeability. However, solving the diffusivity also requires information about the total compressibility and fluid viscosity. The FMM Plug-in interface allows the plug-in user to specify numerical values for both the total compressibility and fluid viscosity.

The remaining input that is required for the Fast Marching Method Algorithm is the well completions information (specifically, location and direction of completed cells). This information is necessary to determine the initial position of the propagating pressure front at the beginning of the Fast Marching Method calculations. Unfortunately, the 2013 version of the Ocean framework does not include full functionality of the Development Strategy API. However, the FMM Plug-in is able retrieve the needed input using information from the *Grid, Borehole, Completion,* and *Perforation* domain objects after some data manipulation, as illustrated in Figure 3.7.

Initially, trajectory points (points that define the trajectory of the well in the XYZ space) are obtained from each well borehole object. These points are then transformed into a 3D polyline object in XYZ space. Once the 3D polyline of the well trajectory is created, it is intersected with the grid object of the reservoir model using an intersection service available through the Ocean API. This Ocean service computes the intersection points of the well trajectory with any cell face inside the grid, see Figure 3.8. From these intersection points, we can determine an ordered list of grid cells containing the well borehole path.

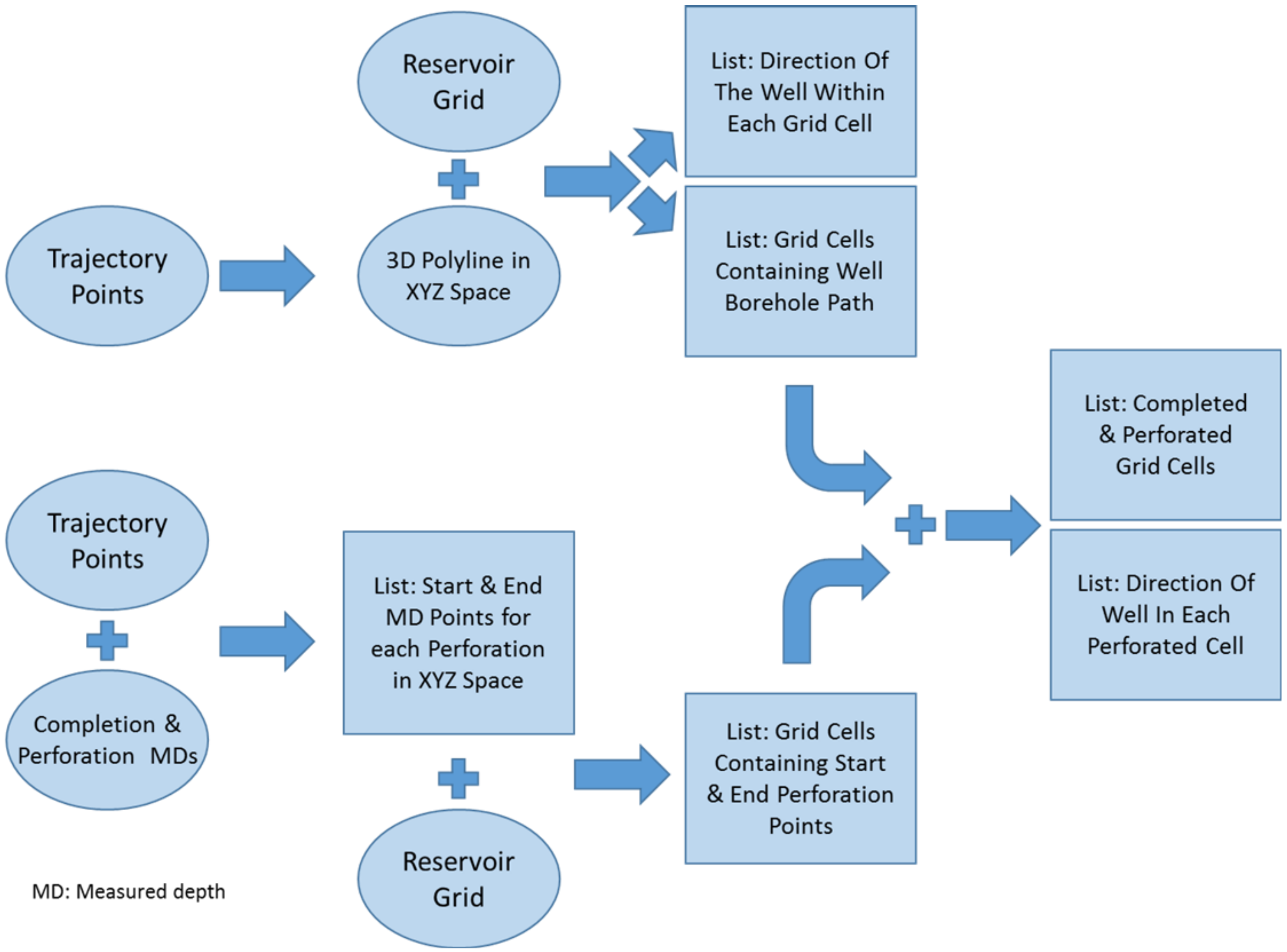

Trajectory Points
Reservoir Grid
3D Polyline in XYZ Space
List: Direction Of The Well Within Each Grid Cell
List: Grid Cells Containing Well Borehole Path
Trajectory Points
Completion & Perforation MDs
List: Start & End MD Points for each Perforation in XYZ Space
Reservoir Grid
List: Grid Cells Containing Start & End Perforation Points
List: Completed & Perforated Grid Cells
List: Direction Of Well In Each Perforated Cell
MD: Measured depth

Figure 3.7: Process of determining location and direction of completed grid cells

In addition, using the intersection points of both intersected cell faces for each cell to create a vector yields the direction of the well within that cell.

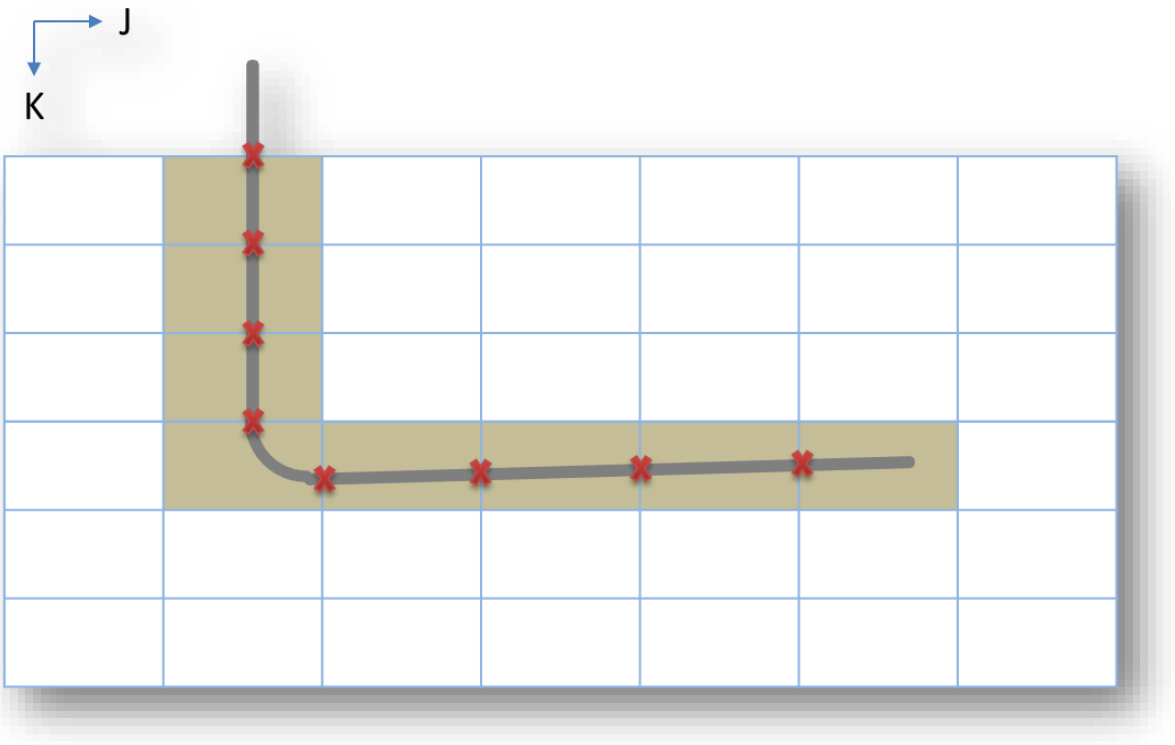


Figure 3.8: Ocean API intersects a well trajectory with a reservoir model grid

However, we are only interested in grid cells containing completed and perforated sections of the borehole since these are the locations where fluid flow occurs between the reservoir and the well. To determine which of the grid cells (among the list of grid cells established earlier) are perforated, we use the completion and perforation objects. They provide start and end *Measured Depth* of each perforation along the well trajectory. The well trajectory records we initially obtained from the borehole object include both XYZ coordinates and

Measured Depth and thus can be used as a mapping reference. The perforations start and end Measured Depths are then linearly interpolated using the mapping reference to obtain their counterpart points in the XYZ space. Once the start and end points of each perforation is known in the XYZ space, we use another Ocean API to find out which grid cells these start and end points are located inside. Finally, these grid cells are referenced against the list of ordered grid cells to obtain a final list with completed and perforated grid cells.

If we elect to perform pressure or rate calculations using 3D to 1D coordinate transformation after getting the diffusive time of flight, then additional parameters are needed for the simulation. These parameters are entered in a readable ASCII text file and its file path is provided as part of the inputs. The format of the text file containing additional parameters needed for the 1D simulation is covered in Appendix A.

3.3.2 Generation of Formatted Data into ECLIPSE-Style Binary Files

The FMM Processor has been designed to accept input pre-processed as encoded binary files. Binary data files, although not viewable by text editors, have significantly smaller file sizes relative to text files with the same amount of information. Using binary data files also yields faster file read/write times. Binary data files have to be created in a pre-determined file format. FMM Plug-in follows ECLIPSE-style binary file formats which are an industry-standard for reservoir model data files.

The FMM generated input is organized into three binary files and one text file. The first file, named grid file, contains the geometrical information of the reservoir model grid and has a file extension of .GRID (ECLIPSE file type for grid files). It contains the IJK dimensions of the grid and XYZ coordinates for the eight corner-points of each grid cell. The second file, named Init file, contains the reservoir grid properties (like porosity, permeability in three directions, …) and has a file extension of .INIT (ECLIPSE file type). The third binary file, named the restart file, contains well and completion information and has a file extension of .X0000 (ECLIPSE file type). Finally, the text file, named FMM input header file, contains information of other files included as input (like grid file, init file, and restart file), and has a file extension of .FIP (This file uses ASCII formatting and is viewable using ordinary text editors). The last file also includes information about the total compressibility and fluid viscosity, and name of the file containing the additional parameters for pressure or rate calculations (if a 1D simulation was to be performed). These files are summarized in Table 3.1 below.

Table 3.1: FMM input file types & content

File Type	**File Content**
*filename***.GRID**	Grid geometry
*filename***.INIT**	Grid block properties (porosity, permeability)
*filename***.X0000**	Well and completion information
FMM**.FIP**	Input files header and total compressibility and fluid viscosity
*filename***.TXT** (optional)	Additional parameters needed for pressure/rate calculations

3.3.3 Running the FMM Algorithm

The Fast Marching Method algorithm and the solution of the Eikonal equation, as described in chapter II, are implemented in the FMM Processor, which has been built as a separate executable application. Once the required input has been prepared and generated as formatted binary and text files, the FMM Processor is started as a system process within the FMM Plug-in.

The FMM Processor initially goes through the FMM input header file and then reads all the formatted input data. It uses the locations of completed grid cells as initial positions for the pressure front. The local diffusivity for a grid cell can be calculated using the grid cell properties (porosity and XYZ permeability), total compressibility, and fluid viscosity. Then, the diffusive time of flight is calculated, using the local grid cell diffusivity, by solving the discretized Eikonal equation. The calculation starts at the initial positions (completed cells) and continues until all grid cells are reached, following the steps of the algorithm in section 2.3. After calculating the diffusive time of flight values for grid cells, the physical arrival time can be obtained.

After estimating the arrival time at each cell the drainage volume at any given time can be calculated by summing up the pore volume of all cells contained within the contour of the pressure front at that given time. If a pressure or rate calculation has been requested, FMM Processor uses the additional parameters provided (Phase information, PVDG table, Dual

Porosity settings, Scheduling information, …) with the diffusive time of flight to perform coordinate transformation and 1D simulation, as described in chapter II.

3.3.4 Importing and Visualization of Results

In the post-processing stage, FMM Plug-in begins to import results of the calculations performed by FMM Processor. These results are organized into two text files, or four if a pressure/rate calculation was performed. The first two text files contain the calculated diffusive time of flight and the physical time of arrival values for every grid cell. The third and fourth text files contain well bottom-hole pressure information and well rate information, respectively, at pre-defined time steps. Table 3.2 summarizes the result text files and their content.

Table 3.2: FMM result text files

File Name	File Content
1. *casename*.Tau**.GRDECL**	Diffusive time of flight
2. *casename*.Time**.GRDECL**	Physical time of arrival
3. *casename*_BHP	Bottom-hole pressure vs. Time
4. *casename*_RATE	Well rate vs. Time

FMM Plug-in automatically reads and imports all results from the text files after the calculations are completed. The diffusive time of flight and the physical time of arrival solutions are both used to create Grid Property domain objects and are then merged with the existing grid properties. Converting the solutions to grid property objects makes them compatible with the rendering tool in Petrel and are then easily visualized in a 3D window. The 3D window renderer uses *Open Inventor* technology, a third party software, and offers state-of-the-art visualization capabilities.

On the other hand, bottom-hole pressure and well rate data is converted into Petrel function domain objects (used to describe linear regression functions or smooth functions), so that they are compatible with and can be viewed in the Function Window in Petrel. The bottom-hole pressure and well rate are also assigned appropriate Petrel *data templates* to fully integrate the data into Petrel and allow it to show units and be usable with other Petrel features. The bottom-hole pressure is assigned the Pressure Template from the Petrophysical Group. The well rate is assigned the ProductionRateGas Template from the Production Group. Finally, the time steps for both items are assigned the SimulationTime Template from the Petrophysical Group. Examples of bottom-hole pressure and well rate results displaced in Petrel Function windows can be seen in Figure 3.1 above.

CHAPTER IV

APPLICATIONS AND TESTING

This chapter shows several applications of the Fast Marching Methods. Using the FMM Plug-in, we illustrate the capacity and utility of this novel approach. First, we present two examples of unconventional reservoirs with horizontal wells. We visualize the depth of investigation at different times and we use the estimated drainage volume to calculate the bottom-hole pressure and well rate. These examples display the user-friendly interface of the FMM Plug-in and demonstrate the simplicity and intuitive appeal of the results. The results produced are validated through comparison with an industry-referenced finite difference simulator.

This chapter also includes a study of scalability of CPU runtime of the Fast Marching Methods. The study covers the impact on runtime due to increasing number of grid cells of a reservoir model in 2D and in 3D cases. The scalability behavior is also presented in contrast of the scalability behavior of an industry-reference finite difference simulator.

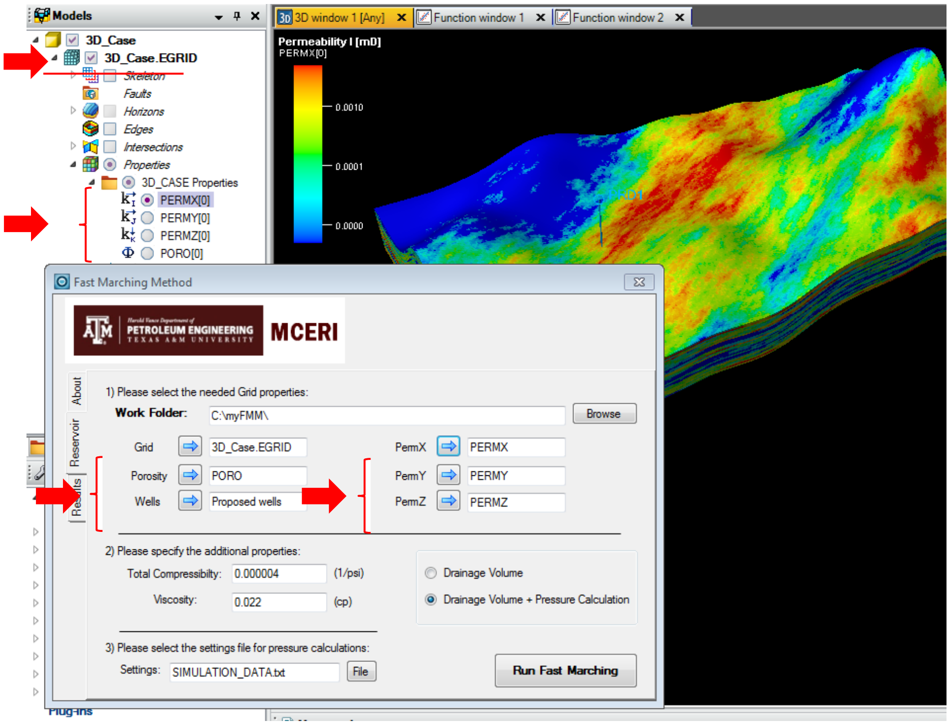
4.1 Example I: Heterogeneous Reservoir with Multi-Stage Hydraulic Fractures

This is a synthetic example of an unconventional reservoir that uses the same permeability field used by Zhang et al. (2013) as previously referenced in Chapter II. With this model, we illustrate the effectiveness of the Fast Marching Methods approach in drainage volume calculations and flow visualization. The model has a horizontal well with five transverse hydraulic fractures. The reservoir model has matrix permeability between 1 nanodarcy and 0.001 millidarcy. It contains 400×200×30 or 2.4 million grid cells with dimensions of 4000ft×2000ft×120ft. Table 4.1 lists the parameters used for this model. In this example, we calculate the bottom-hole pressure under well rate constraints.

Table 4.1: Model parameters for multi-stage hydraulically fractured reservoir

Parameter	Value
Fractures half length (ft)	250
Reservoir porosity	0.15
Gas viscosity (cp)	0.022
Initial reservoir pressure (psi)	5000
Matrix permeability (md)	0.0007
Fracture permeability (md)	10
Reservoir temperature (F)	175
Total compressibility (psi^{-1})	4×10^{-6}
Production rate (Mscf/day)	100

After launching the reservoir case within the Petrel platform, the required properties can be entered into FMM Plug-in (see Figure 4.1). The interface of the FMM Plug-in is simple and user-friendly and allows entering input data with simple drag and drop method from the Models Panel directly to the plugin window. Appendix A includes a user guide for the FMM Plug-in. After running the Fast Marching Methods calculation, the plug-in quickly produces the diffusive time of flight and physical time of arrival distributions and automatically displays the calculated bottom-hole pressure and well rate graphs (Figure 4.2).



Fast Marching Method

1) Please select the needed Grid properties:

Work Folder: C:\myFMM\ Browse

Grid	3D_Case.EGRID	PermX	PERMX
Porosity	PORO	PermY	PERMY
Wells	Proposed wells	PermZ	PERMZ

2) Please specify the additional properties:

Total Compressibilty: 0.000004 (1/psi)

Viscosity: 0.022 (cp)

○ Drainage Volume

● Drainage Volume + Pressure Calculation

3) Please select the settings file for pressure calculations:

Settings: SIMULATION_DATA.txt File

Run Fast Marching

Figure 4.1: FMM Plug-in user-friendly interface

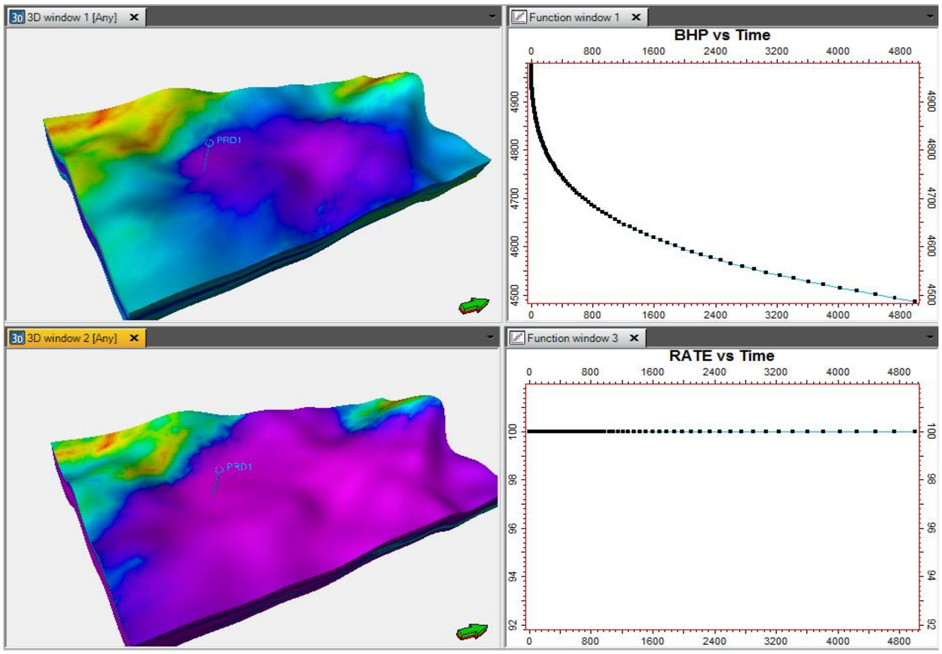


Figure 4.2: FMM results displayed using Petrel visualization windows

Integrating the calculated diffusive time of flight and physical time of arrival with the existing grid properties in Petrel, enables the use of the results with various Petrel tools. Here we use the 1D filter tool from Petrel with the calculated physical time of arrival as threshold criteria to visualize the flow at different time intervals. Figure 4.3 shows the evolution of the drainage volume at 1 month, 1 year, 10 years, and 30 years. The bottom-hole pressure calculated and compared with the results from ECLIPSE show reasonable match, as shown in Figure 4.4.

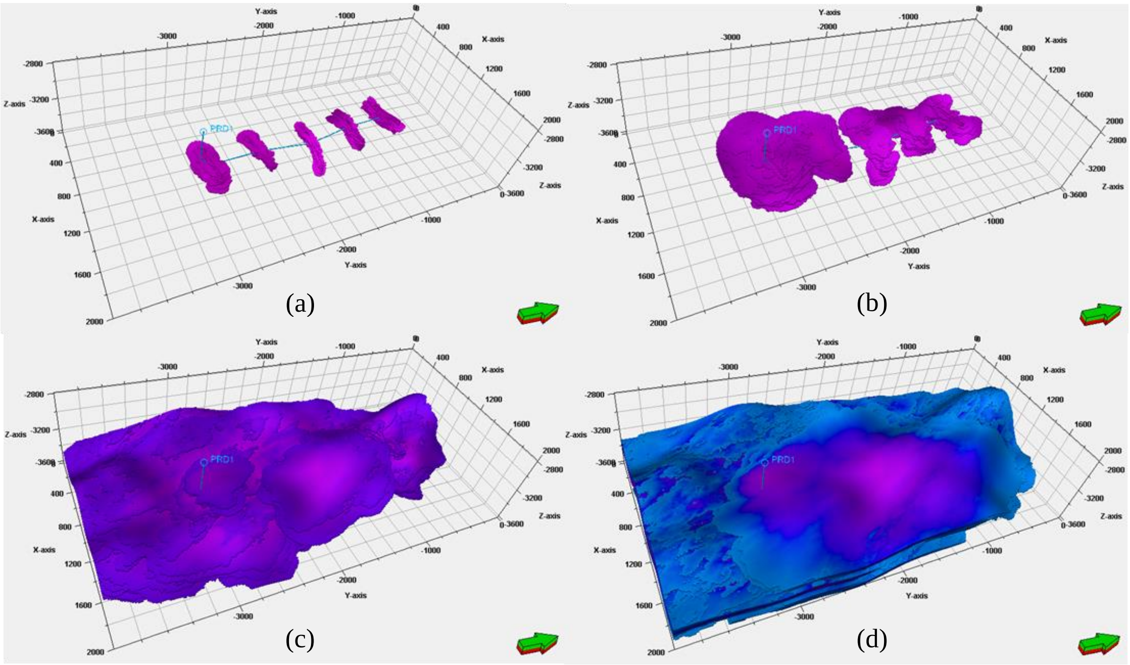


Figure 4.3: Drainage volume at (a) 1 month; (b) 1 year; (c) 10 years; (d) 30 years

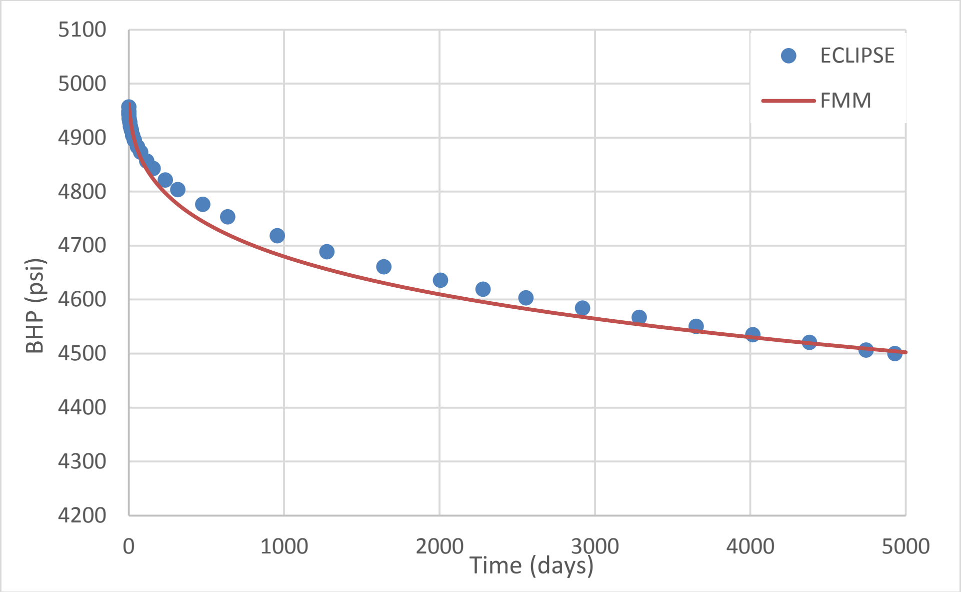


Figure 4.4: Calculated BHP by FMM compared with results from ECLIPSE

4.2 Example II: Dual Porosity

In this example, we present another synthetic model of an unconventional reservoir. This model uses dual porosity to represent the distribution of natural fractures in the reservoir. The FMM Plug-in is used to calculate the drainage volume and estimate the gas production rate over three years. The results are then compared and validated with results from ECLIPSE industry-reference finite difference simulator using its Dual Porosity option. This model includes a horizontal well with 40 hydraulic fractures grouped into ten equal clusters. The fracture permeability field of the reservoir is shown in Figure 4.5. The parameters used for this calculation are summarized in Table 4.2.

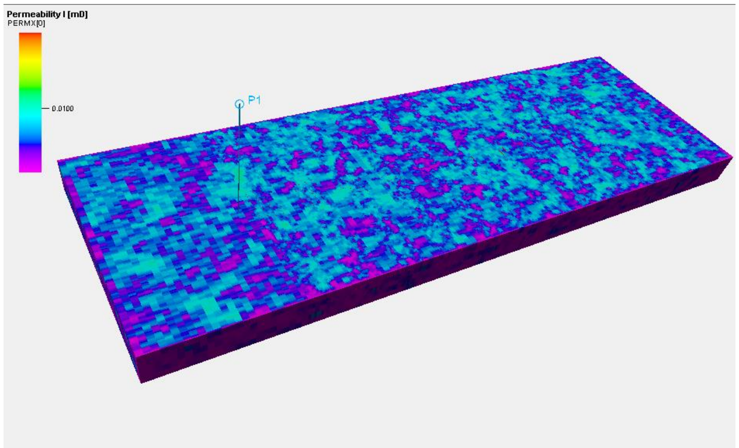


Figure 4.5: Fracture permeability field for Dual Porosity case

Table 4.2: Parameters used in the Dual Porosity example

Parameter	Value
Grid dimensions (cells)	100 × 394 × 5
DX, DY, DZ (ft)	10, varies logarithmically, 20
Initial reservoir pressure (psi)	4970
Matrix porosity	0.076
Matrix permeability (md)	kx = ky = 0.0001, kz = 0.00001
Hydraulic fracture permeability (md)	kx = ky = 10, kz = 1

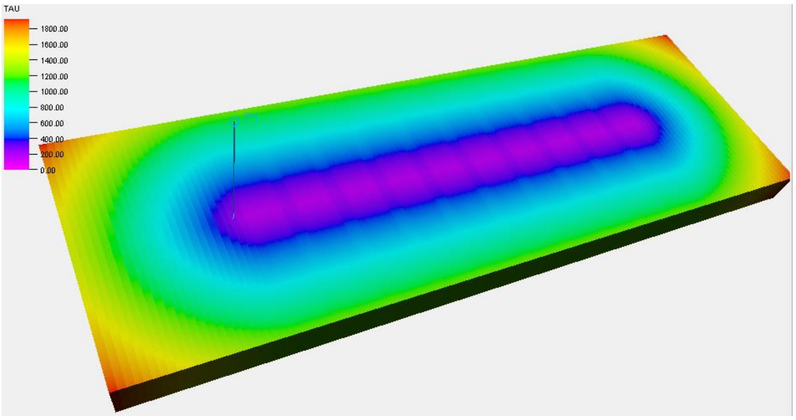


Figure 4.6: Diffusive Time of Flight distribution

In the FMM calculation uses the fracture permeability and fracture porosity to obtain the diffusive time of flight distribution, shown in Figure 4.6, and the drainage volume. This due to the fact that matrix pores are only connected through the fractures the matrix is considered secondary permeability and porosity. The fluid flow occurs through between the fractures grids. Once the diffusive time of flight and the drainage volume is calculated, FMM can incorporate the constant matrix porosity and permeability and perform the 1D simulation to calculate well production rate. The results are compared with ECLIPSE in Figure 4.7 below. We can see reasonable agreement with the finite difference simulator demonstrating FMM capacity to handle dual porosity models.

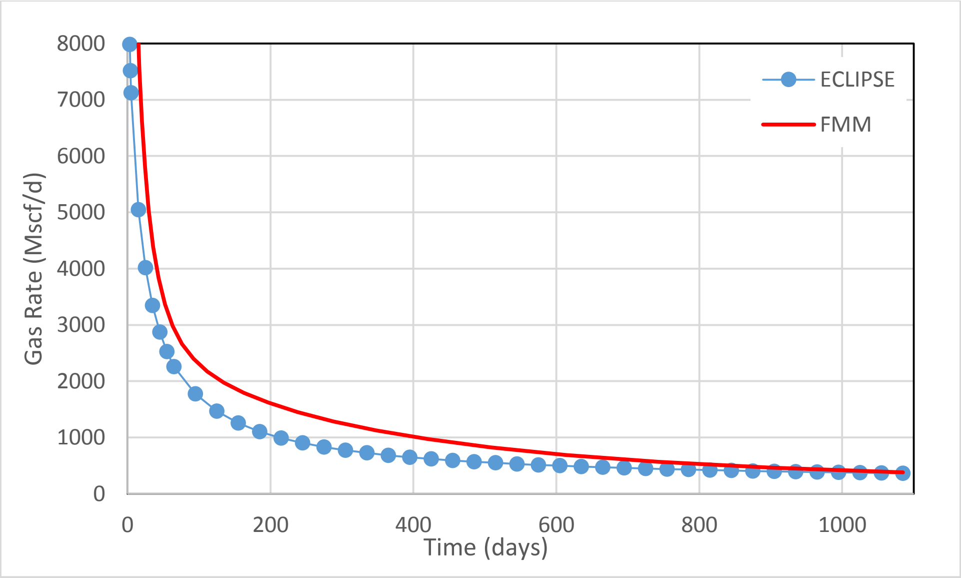


Figure 4.7: Gas rate forecast by FMM compared with results from ECLIPSE for a Dual Porosity reservoir

4.3 Scalability of Runtime

In this section, we study the scalability of CPU runtime of the FMM calculations under the impact of increasing number of grid cells. The result are presented in comparison with the scalability behavior of ECLIPSE industry-reference finite difference simulator. The synthetic model used for this study is a homogenous reservoir with dimensions of 1000ft × 1000ft × 30ft and includes one producer. The reservoir has a homogeneous permeability of 0.001 millidarcy and 5% porosity. The reservoir parameters are summarized in Table 4.3 below.

Table 4.3: Reservoir properties used for scalability analysis

Parameter	Value
Dimensions (ft)	1000 × 1000 × 30
Reservoir permeability (md)	0.001
Porosity	0.05
Initial reservoir pressure (psi)	5000
Reservoir temperature (F)	175
Gas viscosity (cp)	0.02
Total compressibility (psi^{-1})	4×10^{-6}
BHP constraint (psi)	1500

FMM is used to calculate the drainage volume of the reservoir model and estimate the well production rate performance over ten years. The same model is discretized with increasing resolution and every time the calculation is performed CPU runtime is measured. All calculation are then repeated using ECLIPSE finite difference simulator over ten years as well.

The discretization process was done in two fashions. In the first case, the reservoir was designed as a 2D model with increasing resolution. Table 4.4 details the results of calculations runs the sizes of the models used. The results of the scalability is displayed on a log-log plot in Figure 4.8. In the second case, the reservoir model was discretized in a 3D pattern. Again, the same cases were run with both FMM and ECLIPSE and the runtime results are summarized in Table 4.5. The results of the scalability of runtime for 3D cases are compared with ECLIPSE on a log-log plot in Figure 4.9.

Table 4.4: CPU runtime using FMM and ECLIPSE for 2D cases

NX*NY	# of Cells	ECLIPSE	FMM
		CPU Time (s)	CPU Time (s)
40*40	1,600	0.0348	0.119
80*80	6,400	0.0616	0.149
160*160	25,600	0.2464	0.259
320*320	102,400	1.6430	0.725
640*640	409,600	13.9315	2.616
1000*1000	1,000,000	51.6252	6.246
2000*2000	4,000,000	268.3030	25.324
3000*3000	9,000,000	844.1703	59.560

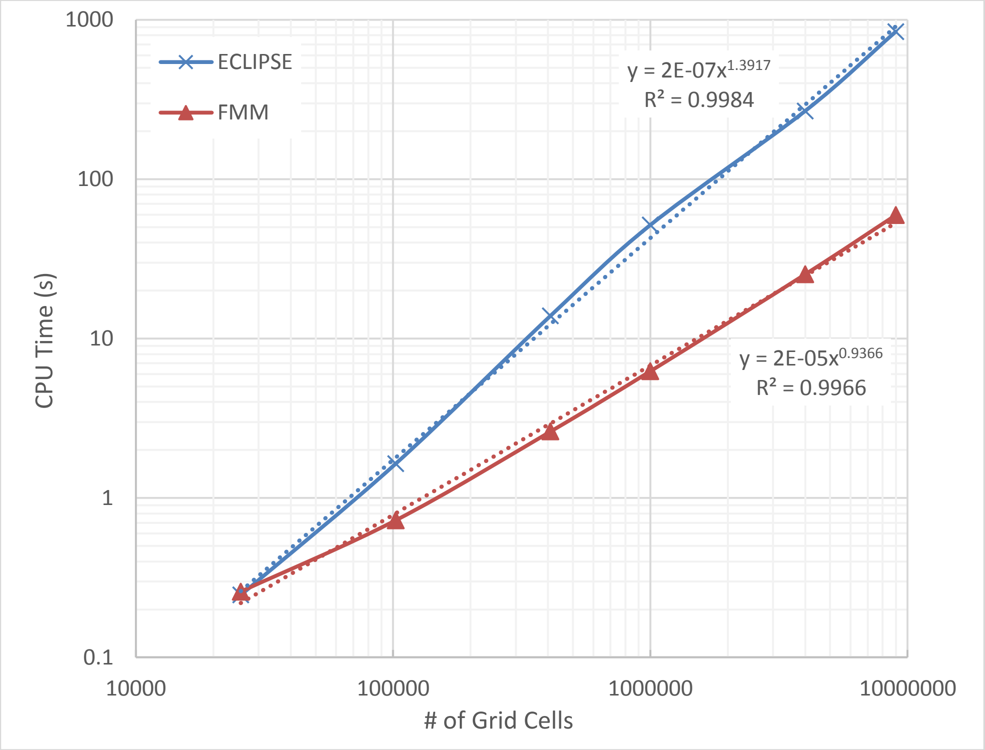


Figure 4.8: Scalability of CPU runtime (2D)

Table 4.5: CPU runtime using FMM and ECLIPSE for 3D cases

NX*NY*NZ	# of Cells	ECLIPSE	FMM
		CPU Time (s)	CPU Time (s)
40*40*4	6,400	0.0775	0.168
80*80*8	51,200	0.5133	0.578
160*160*16	409,600	6.3315	4.082
320*320*10	1,024,000	25.6433	10.100
500*500*10	2,500,000	87.7002	25.480
1000*1000*10	10,000,000	601.3557	110.460

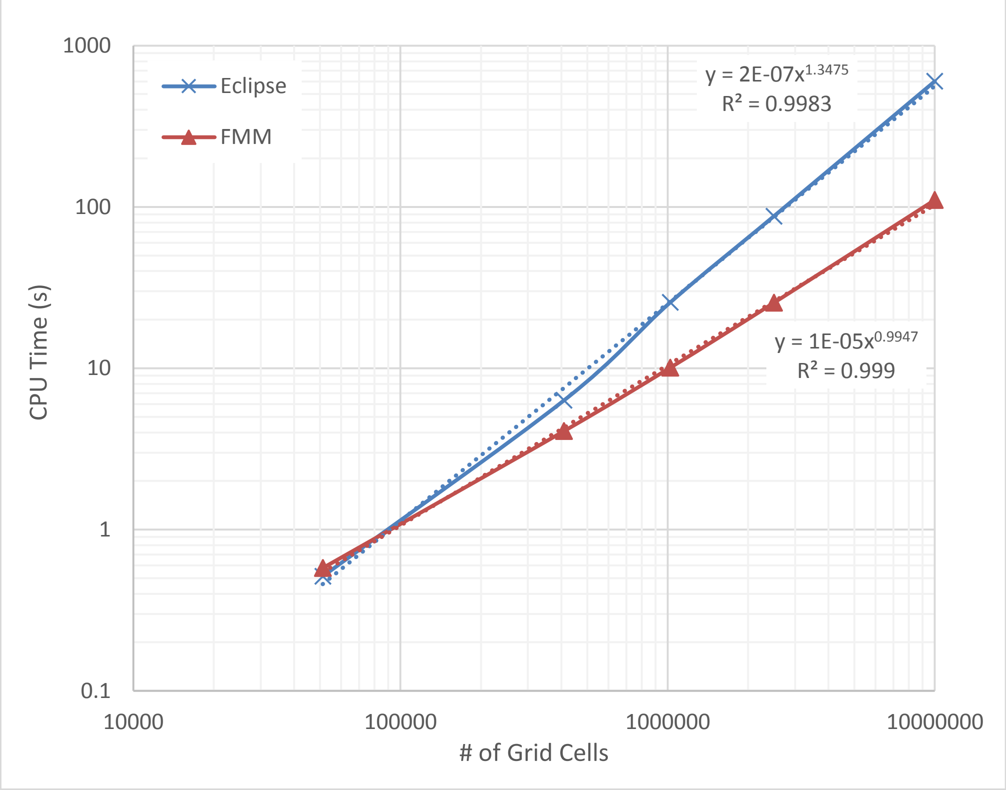


Figure 4.9: Scalability of CPU runtime (3D)

The results of the comparisons in both 2D and 3D scenarios show the clear advantage in the speed of FMM over the finite difference simulator. In fact the gain in efficiency goes up with increasing number of grid cells in both 2D and 3D with speed advantage of up to an order of magnitude. With very small grid sizes (<10,000 cells), however, FMM lagged behind the finite difference simulator by a fraction of second. This is due to small constant overhead in the FMM calculation of about 0.15 seconds that does not change with grid size.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary and Conclusions

This thesis reviews the novel approach of Fast Marching Methods and demonstrates its capacity to estimate the drainage volume and forecast well performance through several applications. This work also includes the integration of the Fast Marching Methods with a common commercial E&P software platform in the form of an application plug-in to effectively utilize its tools and maximize efficiency. A study on scalability of runtime performance for FMM and a comparison with an industry-reference finite difference simulator is also included in this work.

We saw that the generalization of the concept of depth of investigation to heterogeneous reservoirs utilizes the idea of 'diffusive' time-of-flight (DTOF), τ, which relies on the observation that the 'pressure fronts' propagate with a velocity given by the square root of diffusivity. The depth of investigation better accounts for the non-uniform pressure fronts that may be distorted due to heterogeneity effects. To obtain the propagating pressure front, we solve the Eikonal equation, which is derived from an asymptotic solution of the diffusivity equation. The Eikonal equation can be solved very efficiently using the Fast Marching Methods, which is a front tracking method.

Once the diffusive time of flight distribution is calculated, the drainage volume at any time can be calculated by summing up the cell pore volumes within the contour of the propagating front. Our proposed approach performs a coordinate transformation to reduce the diffusivity equation to 1D using the calculated DTOF to estimate pressure dropdown under constant well rate and to forecast well production rate under a constant bottom-hole pressure constraint.

Implementing the concepts of Fast Marching Methods into a software application is necessary to take advantage of this novel approach. Integrating the application with the Petrel commercial E&P software platform enables utilizing the benefits of FMM more effectively via the extension of the Petrel's tools into the FMM workflow. Petrel's state-of-the-are visualization renderers facilitate capitalizing on the simplicity and the intuitive appeal of the Fast Marching Methods.

FMM Plug-in has been applied to examples with several unconventional reservoirs containing multi-stage hydraulically fractured horizontal wells and dual porosity and has produced results agreeing with an industry-reference finite difference simulator. A study on scalability of CPU runtime for FMM showed clear advantage over ECLIPSE finite difference simulator with an advantage in speed reaching an order of magnitude.

5.2 Recommendations

The current state of the Fast Marching Methods is very promising. There are a few areas that could benefit from further development:

- Further studies of the early time near-well disagreement with finite difference simulation to increase accuracy.
- Development and extension of the FMM software to include the compositional and multiphase models.
- Combining the FMM Processor C++ code directly with the FMM Plug-in C# code to reduce massive overhead. Currently information transfer occurs by writing data to the hard drive then reading it back, which is a highly time consuming process. Merging the two components could reduce the total runtime (calculations + data transfer) by over 50%.

NOMENCLATURE

A	Surface area
A_k	k^{th} order pressure amplitude in Fourier domain
c	Geometric factor
c_t	Total compressibility
D	Finite difference operator
J	Jacobian of the coordinate transformation
k	Permeability
p	Pressure
$\bar{p}$	Average pressure
$\tilde{p}$	Pressure in frequency domain
p_i	Initial reservoir pressure
p_{wf}	Flowing bottom-hole pressure
Δp	Pressure drop
$\Delta P'$	Pressure derivative with respect to log(t)
Q	Darcy flux
Q_D	Dimensionless rate
Q_{well}	Well production rate
r	Radius of investigation
t	Time
V_p	Drainage volume

$w(\tau)$	Derivative of the drainage pore volume w.r.t. the DTOF
x	Spatial location
α	Diffusivity
λ_t	Total mobility
ω	Frequency in Fourier domain
ϕ	Porosity
τ	Diffusive time of flight
μ	Viscosity

REFERENCES

Akram, F., Watson, J., Rivas, D. 2007. *Thermal Simulation and Visualization in Petrel*. AAPG Search and Discovery. CSPG/CSEG Conference, Calgery, Canada.

Cipolla, C. L., Lolon, E. P., Erdle, J. C., and Tathed, V. 2009. *Modeling Well Performance in Shale-Gas Reservoirs*. SPE 125532-MS.

Cipolla, C. L., Weng, X., Mack, M.G., Ganguly, U., Gu, H. et al. 2011. *Integrating Microseismic Mapping and Complex Fracture Modeling to Characterize Hydraulic Fracture Complexity*. SPE 140185-MS.

Datta-Gupta, A., Kulkarni, K.N., Yoon, S., and Vasco, D.W. 2001. *Streamlines, Ray Tracing and Production Tomography: Generalization to Compressible Flow*. Petroleum Geoscience **7**: 75-86.

Datta-Gupta, A., and King, M.J. 2007. *Streamline Simulation: Theory and Practice*. Society of Petroleum Engineers, Richardson, Texas.

Datta-Gupta, A., Xie, J., Gupta, N., King, M.J., and Lee W.J. 2011. *Radius of Investigation and its Generalization to Unconventional Reservoirs*. Journal of Petroleum Technology **63** (7): 52-55.

Dijkstra, E.W. 1959. *A Note on Two Problems in Connection with Graphs*. Numerische Mathematik **1**: 269-271.

Dong, Z., Holditch, S.A., McVay, D.A.2013. *Resource Evaluation for Shale Gas Reservoirs*. SPE Journal **5**(1):5-16. SPE 152066-PA.

EAGE Daily News. 2012. *The Platform For E&P Decision Making*. Hart's E&P, June 2012. Houston, Texas.

EAGE Daily News. 2013. *E&P Software Platform Improves Workflow and Productivity*. Hart's E&P, June 2013. Houston, Texas.

EAGE Daily News. 2014. *New User Experience to Deliver Productivity Step Change*. Hart's E&P, June 2014. Houston, Texas.

Fan, L., Thompson, J. W., and Robinson, J. R 2010. *Understanding Gas Production Mechanism and Effectiveness of Well Stimulation in the Haynesville Shale through Reservoir Simulation*. SPE 136696-MS.

Fetkovich M. J. 1980. *Decline Curve Analysis Using Type Curves*. Journal of Petroleum Technology **32** (6): 1065-1077.

Ilk, D., Stotts, G. W. J., Anderson, D. M., Mattar, L., and Blasingame, T.A. 2010. *Production Data Analysis – Challenges, Pitfalls, Diagnostics*. SPE Reservoir Evaluation & Engineering **13** (3): 538-552. SPE 102048-PA.

Kuchuk, F. J., *Radius of Investigation for Reservoir Estimation from Pressure Transient Well Tests*, SPE 120515 presented at SPE Middle East Oil and Gas Show and Conference, 15-18 March 2009, Bahrain, Bahrain.

Lee, W.J. 1982. *Well Testing*. Society of Petroleum Engineers, Richardson, Texas.

Schlumberger. 2013. *Ocean Development Framework for Petrel 2013.1. Fundamentals Training.* Schlumberger Information Solutions. Houston, Texas.

Schlumberger. 2014. *Petrel Reservoir Engineering*. Schlumberger Software Product Sheets. Houston, Texas.

Schlumberger. 2014. *Petrel Geology and Modeling*. Schlumberger Software Product Sheets. Houston, Texas.

Schlumberger. 2014. *Ocean Software Development Framework*. Schlumberger Software Product Sheets. Houston, Texas.

Sethian, J.A. 1996. *Level Set Method*. Cambridge University Press, New York City.

Sethian, J.A. 1999. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, New York City.

Song B., Economides, M. J., and Ehlig-Economides, C. 2011. *Design of Multiple Transverse Fracture Horizontal Wells in Shale Gas Reservoirs*. SPE 140555-MS.

U.S. Energy Information Administration. 2014. *Annual Energy Outlook 2014 with Projections to 2040*, http://www.eia.gov/forecasts/aeo/pdf/0383(2014).pdf.

Valko P. P. and Lee W. J. 2010. *A Better Way To Forecast Production From Unconventional Gas Wells*. SPE 134231-MS.

Vasco, D.W., and Datta-Gupta, A. 1999. *Asymptotic Solutions for Solute Transport: A Formalism for Tracer Tomography*. Water Resource Research **35** (1): 1-16.

Vasco, D.W., Keers, H., and Karasaki, K. 2000. *Estimation of Reservoir Properties Using Transient Pressure Data: An Asymptotic Approach*. Water Resource Research **36** (12): 3447-3465.

Vasco, D.W., and Finsterle, S. 2004. *Numerical Trajectory Calculations for the Efficient Inversion of Transient Flow and Tracer Observations*. Water Resource Research **40** (1) W01507.

Virieux, J., Flores-Luna, C., and Gilbert, D. 1994. *Asymptotic Theory for Diffusive Electromagnetic Imaging*. Geophysical Journal International **119** (3): 857-868.

Wharton, D. 2010. *Application Enables Customizable Workflows*. Hart's E&P, 83(10).

Zhang, Y., Bansal, N., Fujita, Y., Datta-Gupta, A., King, M.J., and Sankaran, S. 2014. *From Streamline to Fast Marching: Rapid Simulation and Performance Assessment of Shale Gas Reservoirs Using Diffusive Time of Flight as a Spatial Coordinate*. Paper SPE-168997 presented at the SPE Unconventional Resource Conference, The Woodlands, Texas, USA, 1-3 April.

APPENDIX A

FMM PLUG-IN USER GUIDE

The current version of the FMM Plug-in is compatible with the 2013 release of the Petrel platform. This user guide covers the installation of the FMM Plug-in, the user interface, and explanation of the important keywords.

A.1 Installation

There are two methods available to install the FMM Plug-in using the installation package, shown in Figure A.1. The first method is an asissted installation using a pre-configured Miscrosoft Installer (MSI). For this method, the "FMM2013Installer.msi" and "setup.exe" files are required. Before starting the Petrel application, the user can double click on "setup.exe" to start the assisted installation wizard. However, using this method might require the user to have administrator access to the target computer for a successful installation.

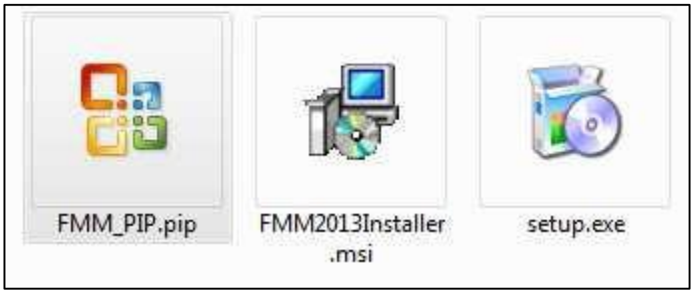


Figure A.1: FMM Plug-in installation package

The second installation method uses the Ocean Plug-in Manager tool in Petrel and the "FMM_PIP.pip" file from the FMM installation package. After starting Petrel, the user can click on "Help" from the Menu Toolbar then clicks on "Ocean Plug-in Manager", as shown in Figure A.2. Then the user can click "Install Plug-in" and select the "FMM_PIP.pip" file from the installation package. Once the installation is complete, the Petrel platform needs to be restarted for the installation to take effect. The FMM Plug-in will be available in the "Processes Panel", as shown in Figure A.3.

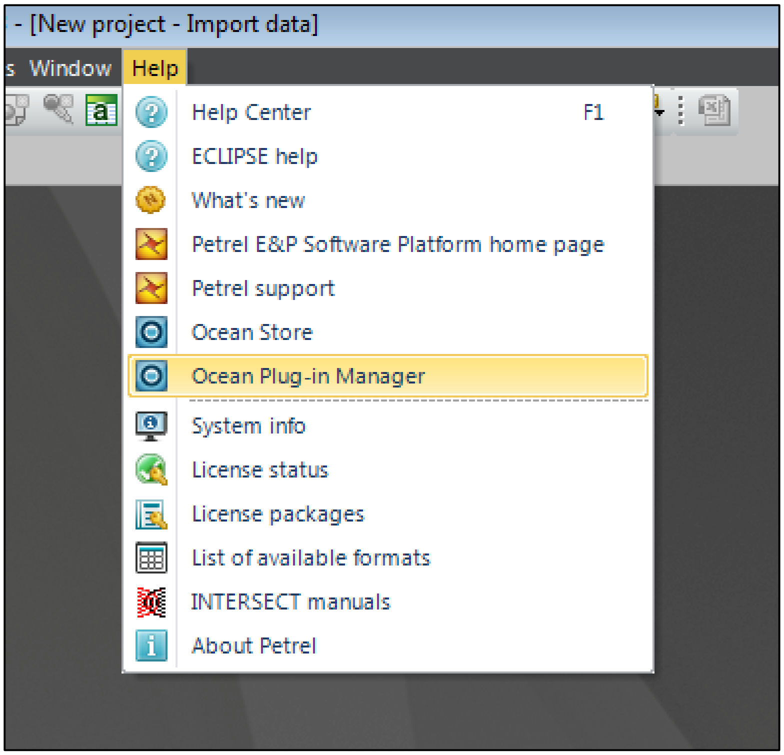
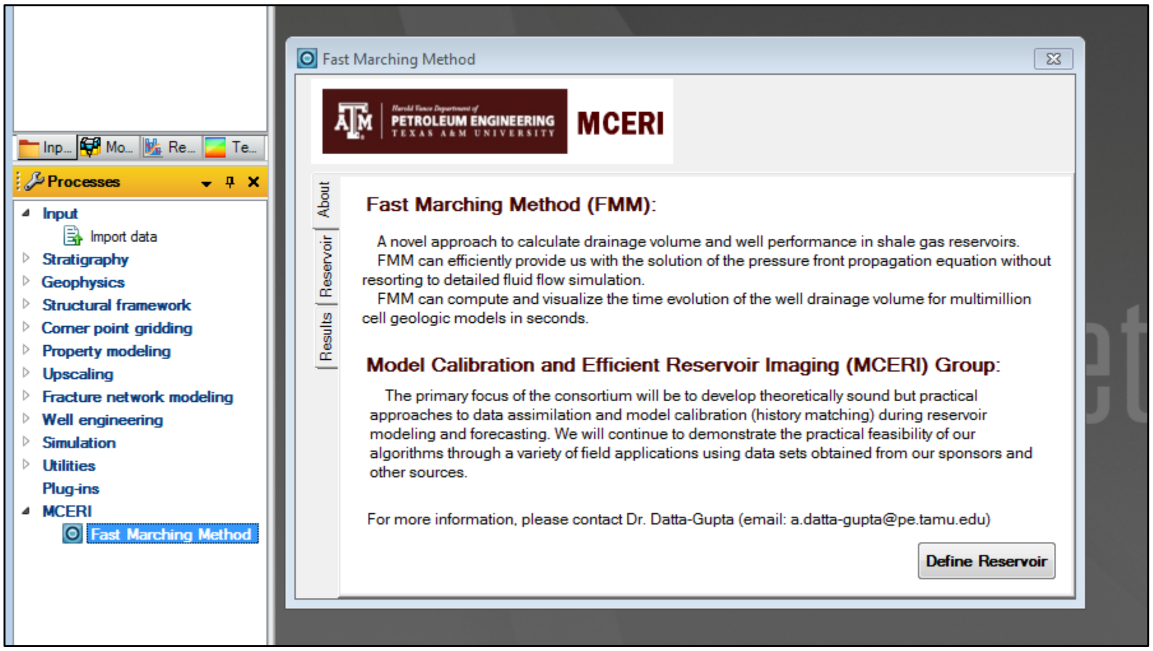


Figure A.2: Installation using Ocean Plug-in Manager in Petrel



Fast Marching Method (FMM):

A novel approach to calculate drainage volume and well performance in shale gas reservoirs.
FMM can efficiently provide us with the solution of the pressure front propagation equation without resorting to detailed fluid flow simulation.
FMM can compute and visualize the time evolution of the well drainage volume for multimillion cell geologic models in seconds.

Model Calibration and Efficient Reservoir Imaging (MCERI) Group:

The primary focus of the consortium will be to develop theoretically sound but practical approaches to data assimilation and model calibration (history matching) during reservoir modeling and forecasting. We will continue to demonstrate the practical feasibility of our algorithms through a variety of field applications using data sets obtained from our sponsors and other sources.

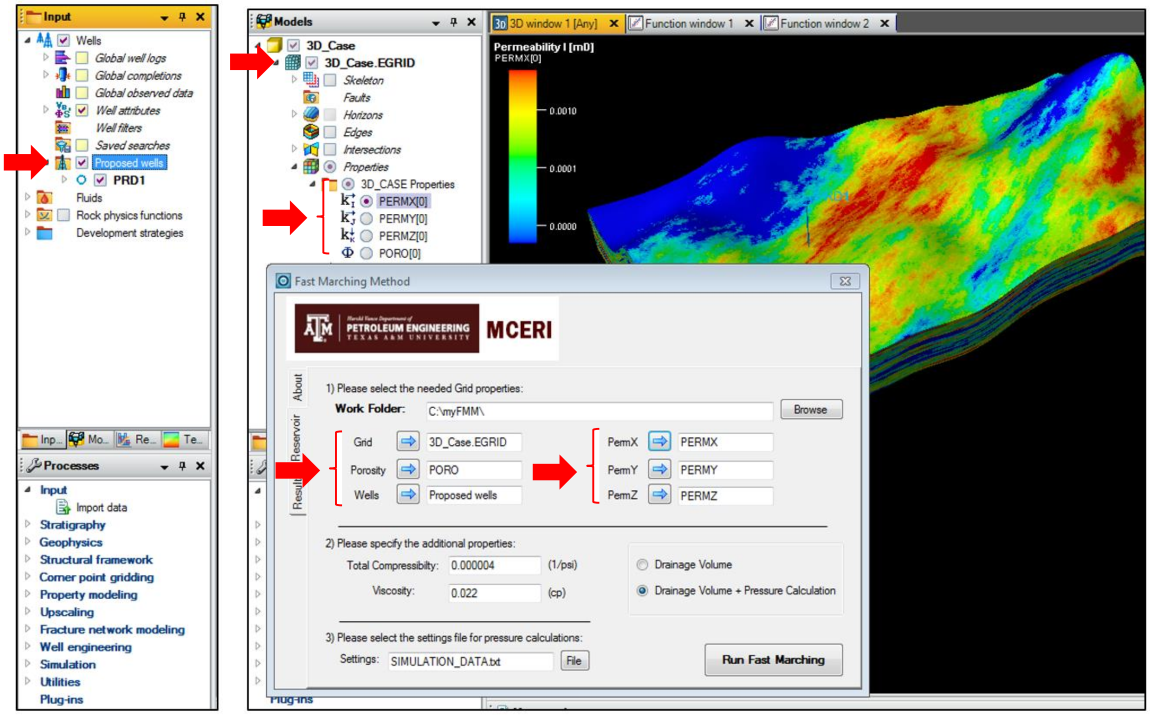
For more information, please contact Dr. Datta-Gupta (email: a.datta-gupta@pe.tamu.edu)

Figure A.3: Successfully-installed FMM Plug-in in Processes Panel

A.2 User Interface

FMM Plug-in has an intuitive and user-friendly interface. For most data input, the interface contains convenient “dropdown buttons” that are familiar to any Petrel user, as shown in Figure A.4. The user can use the dropdown buttons to select the input “Grid” model and its four grid properties (Porosity, PermX, PermY, and PermZ) that are required for the Fast Marching Methods calculations, as highlighted in Figure A.4. The plug-in user can also enter the required well input data by selecting the wells folder from the Petrel “Input Panel” and dropping it in the appropriate field in the FMM Plug-in interface, as highlighted in Figure A.4. Finally, the user can enter numerical values for the total compressibility and fluid viscosity fields. After selecting all input, the user can click “Run Fast Marching” button to start the calculations. Once the calculations are complete, the results will be automatically imported and displayed and a runtime report appears, as shown in Figure A.5.

The user can also perform a 1-D pressure calculation by providing the settings text file for the simulation run (the settings file is described in a following section).



Fast Marching Method

1) Please select the needed Grid properties:

Work Folder: C:\myFMM\

Grid: 3D_Case.EGRID

Porosity: PORO

Wells: Proposed wells

PermX: PERMX

PermY: PERMY

PermZ: PERMZ

2) Please specify the additional properties:

Total Compressibilty: 0.000004 (1/psi)

Viscosity: 0.022 (cp)

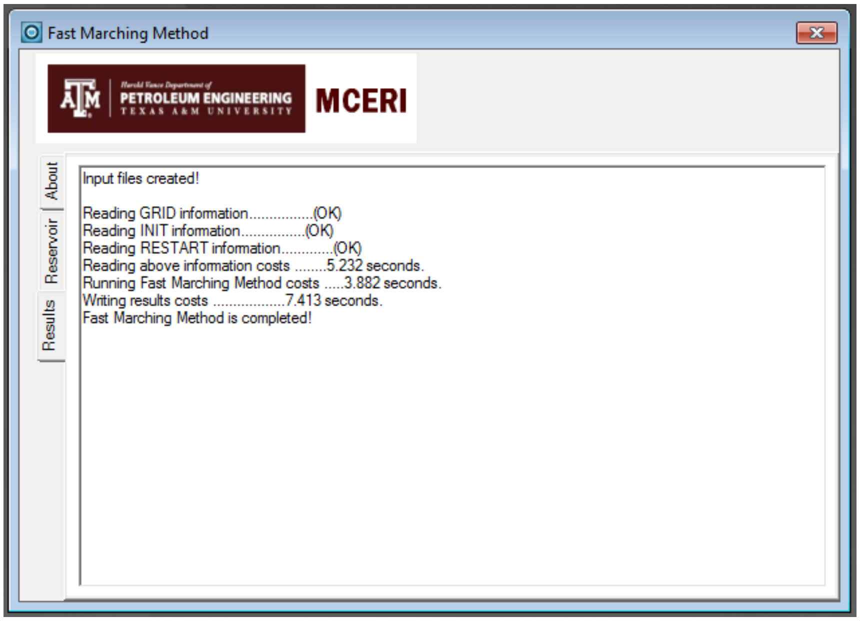
○ Drainage Volume

● Drainage Volume + Pressure Calculation

3) Please select the settings file for pressure calculations:

Settings: SIMULATION_DATA.txt

Figure A.4: FMM Plug-in user interface



```
Input files created!

Reading GRID infomation................(OK)
Reading INIT infomation................(OK)
Reading RESTART infomation............(OK)
Reading above information costs ........5.232 seconds.
Running Fast Marching Method costs .....3.882 seconds.
Writing results costs .................7.413 seconds.
Fast Marching Method is completed!
```

Figure A.5: Calculations log

A.3 Keywords in Settings File for a Pressure Calculation

NTAU 1-D discretization number.

RESERVOIR The RESERVOIR section contains specified phase information using a set of keywords (OIL, VAPOIL, WATER, GAS, DISGAS). In the RESERVOIR section, a set of options can be enabled for the simulation by adding its keyword, like: DUALPORO for dual porosity effect, ADSORB for adsorption, KEROGEN for kerogen, and KLINKENBERG for shear slippage effect.

PRESSURE Initial reservoir pressure (RESERVOIR section).

SWL Initial water saturation (RESERVOIR section).

KR Relative permeability (RESERVOIR section).

PROPS The PROPS section includes reservoir properties tables like PVDG, ROCKTAB and DENSITY. All these tables follow ECLIPSE formatting for property tables.

DUAL POROSITY This section contains the homogenous matrix property information for dual porosity simulation that is specified using keywords, like: MTRXPERM, MTRXPORO, MTRXCOMPRE, SIGMA, SWLM, and KRM.

COMPONENT This keyword is used to specify the molecular fractions and molecular weights needed for the Knudsen diffusion slippage calculation if enabled.

LANGMUIR ISOTHERM This section specifies the properties needed for the adsorption keyword.

WELLSPEC This section includes the well specification which can be entered using keywords of WELLRADIUS, WELLPERM, WELLPORO, and SKIN.

SCHEDULE The SCHEDULE sections specifies the operations to be simulated, the period, and the times at which output reports are requested.

WCONPROD This keyword specifies the control data for production wells. It is followed by record containing the following data: Time (in days), Well constraint (GRAT, ORAT, or BHP), BHP (in psia), Rate (in

STB/day or MSCF/day), and number of steps (used in log distribution).