



Politecnico  
di Torino



**Politecnico  
di Torino**



BAKI ALI NEFT MƏKTƏBİ  
BAKU HIGHER OIL SCHOOL

Master's degree thesis:

# **Three-dimensional Flow Analysis in Complex Fracture Networks Through Green's Function.**

**Supervisors:**

Prof. Javid SHIRIYEV

Prof. Viberti DARIO

**Candidate:**

Willy Jospin KEMTEU DOUMTSOP

M: S319205

Academic year 2024/2025



## Acknowledgements

First and foremost, I want to extend my heartfelt appreciation to my parents and my whole family. Throughout this adventure, their unwavering affection, support, and encouragement have been my biggest source of strength. Being away from my loved ones has been one of the hardest aspects of this academic journey, so studying abroad has not been simple. However, their unwavering faith in me gave me the fortitude and resolve to keep going. For giving me the fortitude, tenacity, and direction to get through all the obstacles along the route, I also thank God.

I am truly appreciative of the guidance, insightful criticism, and skilled mentorship I received from my thesis advisors, **Professor Javid SHIRIYEV** and **Professor Viberti DARIO**, throughout the course of this study. Their assistance has been essential to the completion of this thesis.

**Professor Raffaele ROMAGNOLI**'s way of teaching was the spark for me, and I want to express my sincere gratitude to him. His humility, compassion, humanity, and unwavering encouragement affected me personally and also influenced my academic career. His leadership and role model are much appreciated by me.

I would like to express my sincere appreciation to the staff of the **Geoenergy Engineering** program at **Politecnico di Torino** and **Baku Higher Oil School** for enhancing my global academic and cultural experience.

I would also like to thank **CARBONTECH SRL**, and particularly **Ing. Armando Ruffini**, for allowing me the chance to use my skills in meaningful, real-world projects by employing me as a Project Engineer.

I want to express my gratitude from the bottom of my heart to all my friends and colleagues who have supported, encouraged, and inspired me.



Politecnico  
di Torino



BAKI ALI NEFT MOKTABI  
BAKU HIGHER OIL SCHOOL



## Abstract

Traditional simulation methods struggle with unconventional reservoirs, which have extremely low permeability and complicated natural fracture systems. Traditional numerical approaches like dual-porosity and dual-permeability models are unable to adequately account for the impact of fracture connectivity and topology, particularly in three-dimensional domains. This paper proposes a semi-analytical solution for simulating transient single-phase flow in fractured porous media using Green's functions. The approach allows for precise and effective pressure and production predictions by explicitly incorporating the geometry and connectivity of intricate fracture networks.

The suggested model includes steady-state and transient fracture flow equations together with Green's functions for point, line, and plane sources obtained for an anisotropic cuboidal reservoir. The model is validated against both traditional analytical solutions and commercial numerical simulators, and superposition principles are used to achieve coupling between the reservoir and the fracture. Real field data from the Barnett Shale, multi-stage transverse fractures, and simple planar fractures are among the case studies.

The findings show that the technique can capture precise flow behaviour and pressure transmission dynamics in arbitrarily complicated fracture systems while greatly decreasing computing time. This study aids in the creation of a reliable and scalable methodology for simulating and optimizing production in unconventional reservoirs.



## Sommario

1. Introduction .....	7
1.1 Motivation .....	7
1.2 Objectives .....	9
1.3 Thesis Structure .....	9
2. Literature review.....	10
2.1 Fractured reservoir simulation methods .....	10
2.2 Dual-porosity and dual-permeability models.....	11
2.3 Discrete fracture modeling approaches .....	11
2.4 Green's Function Method in Reservoir Engineering .....	13
2.5 Summary of prior work and gaps .....	13
3. Theoretical framework .....	14
3.1 Governing equations of flow in porous media .....	14
3.2 Green's function solution: point source .....	17
3.3 Green's function solution: line source .....	18
3.4 Green's function solution: plane source .....	21
3.5 Superposition and boundary handling.....	28
3.6 Elliptic Theta Functions and Analytical Integrals .....	28
4. Model Implementation and Validation.....	30
4.1 Algorithm overview .....	31
4.1.1 Fracture discretization .....	32
4.1.2 Numerical time convolution .....	32
4.1.3 Boundary handling via theta functions .....	32
4.1.4 Validation and benchmarking .....	33
4.1.5 Example .....	33
4.2 Coupled fracture-reservoir flow modeling.....	35
4.3 Finite-difference implementation in fractures .....	37
4.4 Non-darcy flow and newton-raphson solution.....	38
4.5 Numerical stability and convergence .....	39
5. Case Studies and Results .....	41
5.1 Single planar fracture (Script in appendix) .....	41
5.1.1 Bilinear flow behaviour:.....	42



5.1.2	Linear flow behaviour: .....	42
5.1.3	Spherical (or pseudo-Radial) flow behaviour:.....	42
5.2	Multi-Stage transverse fractures (Script in appendix) .....	43
5.3	Orthogonal fracture networks (Script in appendix) .....	47
5.3.1	Fracture network representation: .....	47
5.3.2	Flow regimes identified: .....	47
5.3.3	Log-Log plot interpretation .....	48
5.3.4	Drainage maps.....	48
5.3.5	Implications and benefits of the Green's Function approach for orthogonal fracture networks .....	49
5.4	Real field example: Barnett Shale .....	50
5.4.1	Field background and setup.....	50
5.4.2	Modelling approach .....	51
5.5	Flux Distributions and pressure trends .....	54
5.6	Comparison with numerical simulators.....	56
5.6.1	Benchmarking cases .....	56
5.6.2	Results overview .....	57
6.	Discussion.....	58
6.1	Interpretation of pressure behaviors .....	58
6.2	Flow regimes: Linear, Bilinear, Radial, Elliptical .....	58
6.3	Influence of fracture conductivity .....	59
6.4	Advantages over traditional simulation techniques.....	59
6.5	Practical implications for stimulation design .....	59
7.	Conclusion and future work .....	60
7.1	Summary of Findings .....	60
7.2	Limitations of the current approach .....	60
7.3	Future research directions.....	60
8.	References .....	61
9.	Appendix .....	63
9.1	Case 1: Single fracture .....	63
9.2	Case 2 : Mult-stage fracture .....	66
9.3	Case 3: Orthogonal fracture network .....	69

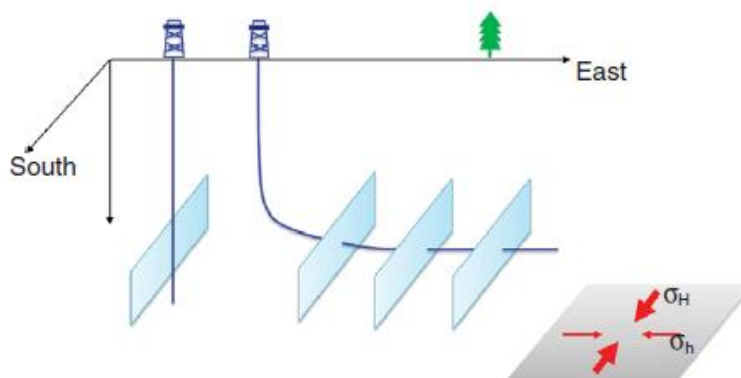
## 1.Introduction

Advanced modelling methods are required to describe fluid movement in complicated geological settings due to the development of unconventional hydrocarbon reservoirs. Unconventional plays, such as shale gas, tight oil, and coalbed methane, depend heavily on significant hydraulic fracturing to increase production, unlike traditional reservoirs. Conventional dual-porosity or structured-grid numerical simulators are unable to accurately replicate the network of fractures, both induced and natural, that frequently results from these treatments.

In order to replicate pressure and production behaviour in complicated fracture networks, this thesis presents a semi-analytical technique based on Green's functions. The formulation allows for quick, physics-based simulations by utilizing the superposition of analytical Green's function solutions for various source types (point, line, and plane) in restricted domains.

### 1.1 Motivation

Fractured reservoirs show a wide range of flow behaviours depending on the fracture geometry, connectivity, and reservoir heterogeneity. The fine gridding required around fractures is difficult for traditional simulators to handle, or they require simplifying assumptions that distort the physical reality of how flow behaves in such systems. Even in three dimensions, these systems may be modelled with a high level of accuracy and computational efficiency by using Green's function.

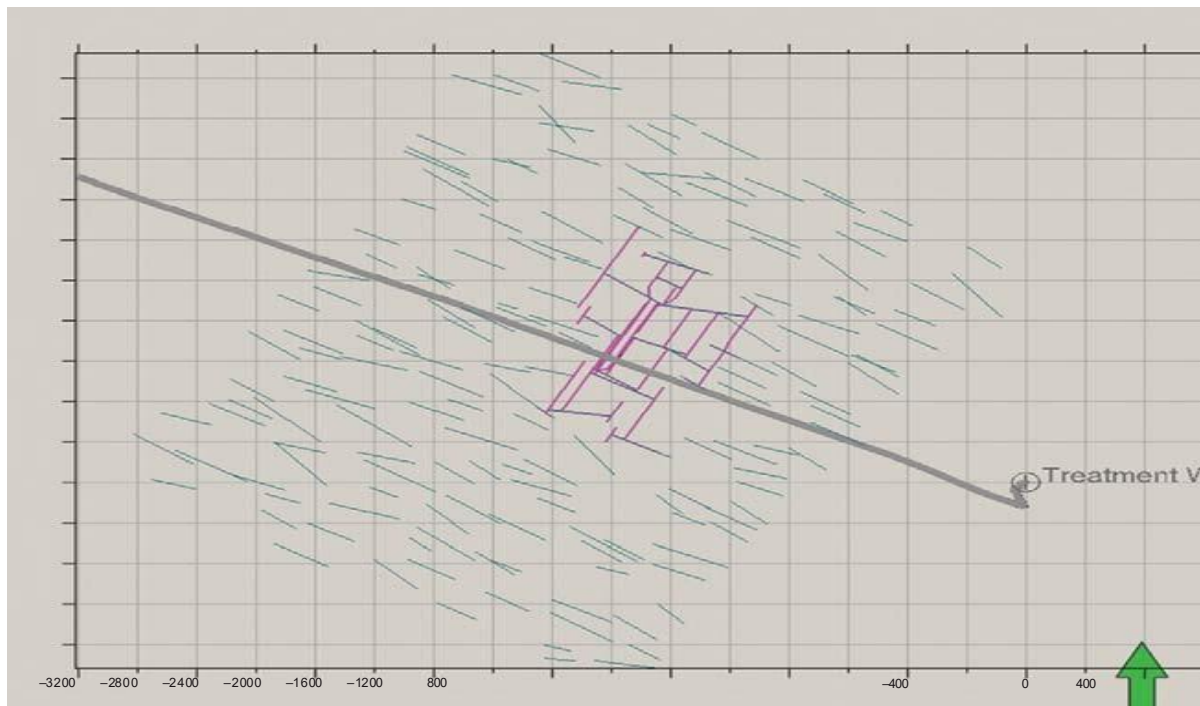


**Figure 1.** Longitudinal and transverse fracture schematics (Zhou 2014)

As shown in **Figure 1.**, hydraulic fractures may take various forms including longitudinal, transverse, and orthogonal geometries. Fig.1 shows the transverse fractures perpendicular to

a horizontal well, typically aligned with the minimum horizontal stress ( $\sigma_h$ ), and longitudinal fractures aligned with the maximum horizontal stress ( $\sigma_H$ ).

The evolution of these fractures and their interaction with natural fractures (Fig. 2) leads to highly variable production outcomes. Understanding these dynamics is critical for optimizing fracture treatments and maximizing recovery.



**Figure 2.** Hydraulic fractures interacting with existing fractures

This figure illustrates the complex interaction between induced hydraulic fractures and pre-existing natural fractures, highlighting deviations, intersections, and connectivity evolution within fractured reservoirs. Fracture propagation is influenced by the in-situ stress field and the orientation of pre-existing discontinuities. (*Adapted from Weng et al., 2011*).



## 1.2 Objectives

The core objectives of this thesis are:

- To derive and apply Green's function solutions for transient flow in fractured porous media.
- To implement a numerical scheme coupling analytical reservoir response with steady-state fracture flow.
- To validate the methodology against existing commercial simulators and analytical benchmarks.
- To apply the model to real and synthetic case studies and interpret pressure and production data.

## 1.3 Thesis Structure

This thesis is structured as follows:

- Chapter 2 provides a detailed literature review on fractured reservoir modeling.
- Chapter 3 outlines the theoretical framework, including derivations of point, line, and plane Green's functions.
- Chapter 4 presents the numerical implementation and solution methodology.
- Chapter 5 details simulation results from a variety of synthetic and real-world fracture configurations.
- Chapter 6 discusses implications and interpretation of results.
- Chapter 7 concludes the thesis and proposes future research directions.

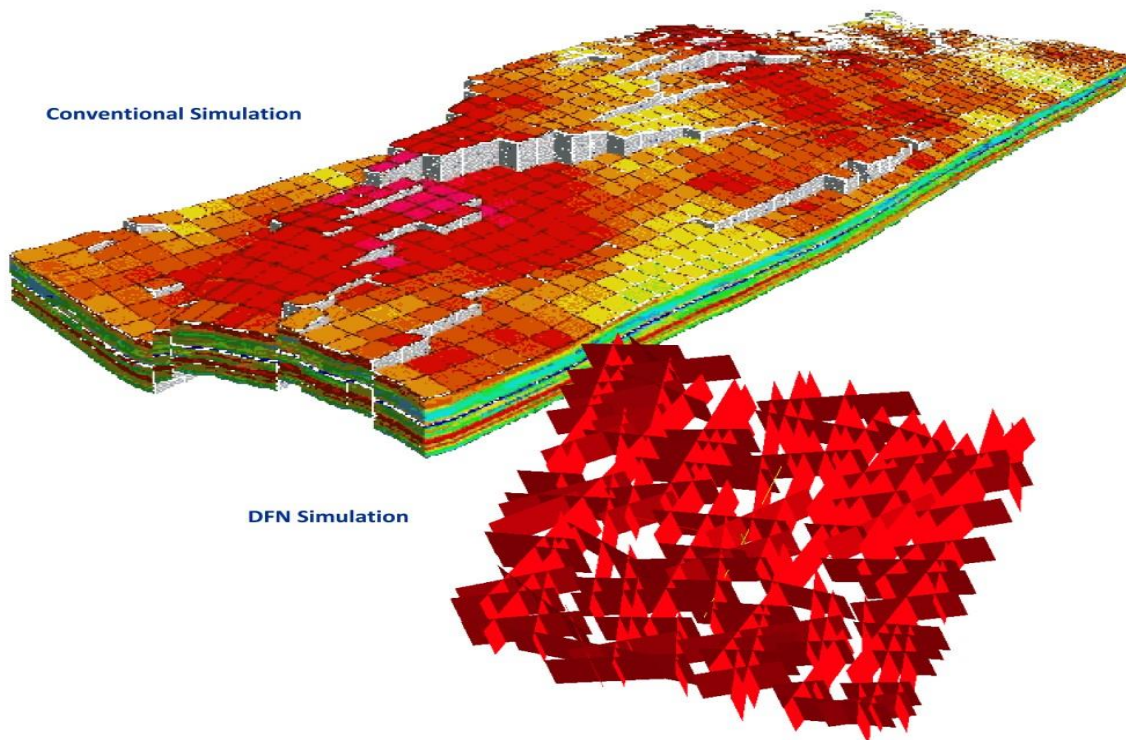
## 2. Literature review

### 2.1 Fractured reservoir simulation methods

Over the course of the last few decades, there has been a lot of progress in the way that fluid flow in fractured reservoirs is modelled. Earlier simulation attempts used average permeability tensors to implicitly model fractures. Later, dual-porosity and dual-permeability models were developed to differentiate between the contributions of the matrix and fractures. The Warren and Root (1963) dual-porosity model posits that fractures operate as high-permeability conduits while the matrix serves as the main storage medium.

Although this conceptual model is effective, it assumes a uniform and orthogonal fracture network, which restricts its use for actual reservoirs that have complicated relationships. Discrete fracture network (DFN) models, which explicitly represent each fracture with its own distinct geometry and characteristics, have been developed.

Although these models give a more accurate depiction of natural and induced fractures, they are computationally demanding and need a lot of data input. Unstructured grids are frequently used in the implementation of DFNs, which enable intricate geometries but also call for sophisticated numerical solvers and precise meshing close to fracture tips.



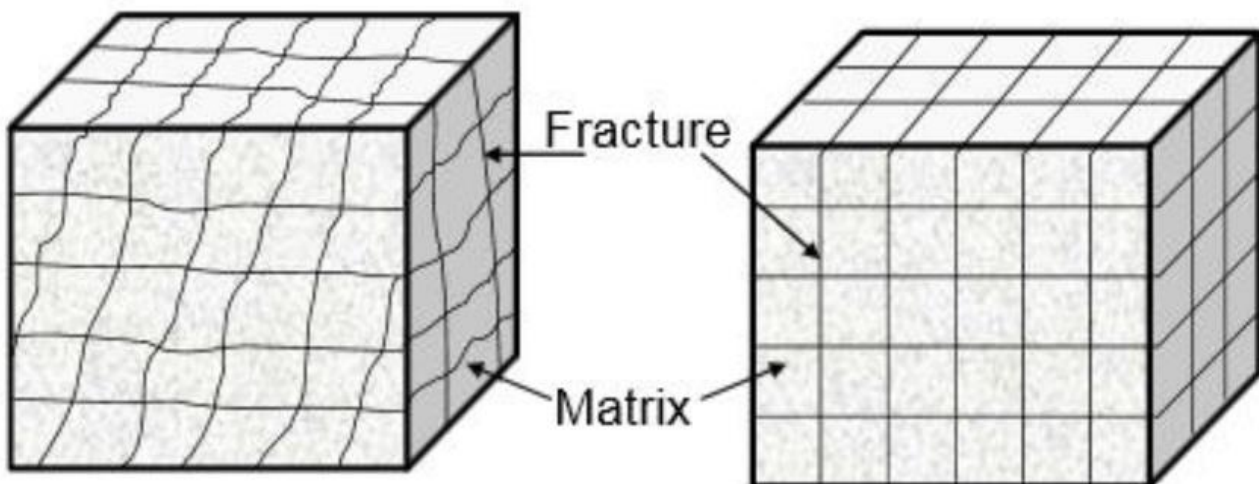
**Figure 3.** Conventional simulation vs DFN simulation (Discrete Fracture Network).

Comparison between conventional structured-grid simulation and discrete fracture network (DFN) simulation techniques. DFN explicitly represents fractures with complex geometries, enabling higher fidelity at the cost of greater computational demands.

## 2.2 Dual-porosity and dual-permeability models

Dual-porosity models treat the rock matrix and fracture system as overlapping continua, with shape factors dictating the flow between the two domains. This model, however, does not address the spatial variability in the interaction between fractures and the matrix, even though it is useful. Dual-permeability models address this by permitting flow in both the fracture and matrix domains, but in fracture systems with high anisotropy or stochastic distribution, the geometric assumptions remain inadequate.

Additionally, both models have trouble replicating realistic fracture geometries, particularly when dealing with vertical, sloping, or multidirectional fractures. Their use in realistic 3D systems is greatly constrained when used in conjunction with structured grid simulators.



**Figure 4.** Conceptual model of dual-porosity and dual-permeability systems.

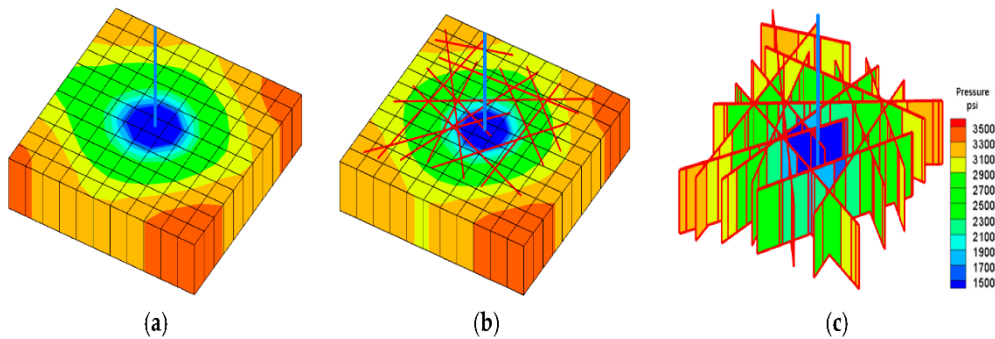
The matrix blocks provide storage, while fractures act as primary conduits. Flow occurs both between and within these continua.

## 2.3 Discrete fracture modeling approaches

The DFN methodology models fractures as explicit geometric features. Two main categories exist:

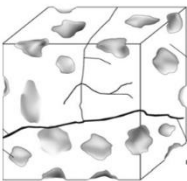
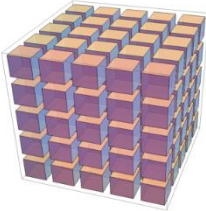
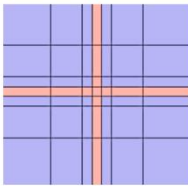
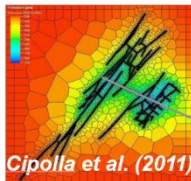
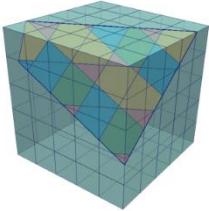
- Embedded Discrete Fracture Models (EDFM)
- Fully Discrete Fracture Models (FDFM).

In EDFM, fractures are incorporated into the matrix using modified transmissibility factors, thus avoiding mesh conformity. In FDFM, mesh is conformal to the fracture network, leading to high accuracy but large computational cost. Despite their flexibility, DFNs are limited by computational scalability and the challenge of mesh generation in field-scale models.



**Figure 5.** Discrete Fracture Modeling (DFM) approach illustrating explicit representation of fractures as individual entities within the reservoir matrix.

To each fracture, specific geometrical and flow properties are assigned allowing high-fidelity simulations.

	Complex fracture	Dual-continuum model	Local grid refinement	Unstructured grids	EDFM
				 <i>Cipolla et al. (2011)</i>	
Accuracy		✗	✓	✓	✓
Flexibility		✗	✗	✓	✓
Gridding		✓	✓	✗	✓
Computational efficiency		✓	✗	✗	✓

**Figure 6.** Comparison of modeling approaches for complex fractures, including dual-continuum models, local grid refinement, unstructured grids, and embedded discrete fracture models (EDFM). The figure contrasts accuracy, flexibility, gridding ease, and computational efficiency across methods.

## 2.4 Green's Function Method in Reservoir Engineering

By utilizing the linearity of the diffusivity equation to depict the reservoir response as a superposition of fundamental solutions, Green's function techniques provide an alternative. As a source term, each fracture segment contributes to the overall pressure field. Based on this theory, Zhou et al. (2014) developed a semi-analytical approach.

The model may replicate transient pressure behaviour without discretizing the whole reservoir by employing Green's functions for point, line, and plane sources in constrained domains. By combining numerical fracture flow simulations with analytical reservoir solutions, the method maintains both speed and precision.

Green's function solution for a point source in a cuboidal reservoir is given by:

$$p(\mathbf{r}, t) = p_{init} - \int_0^t q(t - \tau) G_{point}(\mathbf{r}, \mathbf{r}_o, \tau) d\tau \quad (1) \text{ Zhou 2014}$$

where  $G_{point}(\mathbf{r}, \mathbf{r}_o, \tau)$  is derived from elliptic theta functions to satisfy no-flow boundary conditions in a rectangular geometry. Representing extended fractures and fracture surfaces, respectively, this approach has been applied to line and plane sources. Every Green's function solution is customized to meet boundary conditions and anisotropic diffusivity in the reservoir.

## 2.5 Summary of prior work and gaps

Trade-offs between accuracy, computational speed, and geometric fidelity have been made in prior simulation methods. Classical analytical solutions are quick but restricted in geometry. Numerical simulators provide precise solutions but necessitate significant computational effort and knowledge.

By combining Green's function-based solutions with local numerical fracture flow computations, Zhou (2014) provides a semi-analytical method that bridges this gap. It maintains computational efficiency while facilitating flexible geometric representation and transient analysis.

More studies are needed, nevertheless, even after these improvements:

- Extend the model to multiphase and gas flows.
- Include pressure-dependent permeability and fracture conductivity.
- Handle stress-sensitive reservoir conditions.
- Integrate with real field data for large-scale optimization.

### 3. Theoretical framework

This chapter presents the mathematical foundations of flow in fractured porous media using Green's function solutions. The goal is to construct an accurate, efficient model for predicting pressure behavior in 3D reservoirs with complex fracture networks.

#### 3.1 Governing equations of flow in porous media

The conservation of mass principle and Darcy's law regulate fluid flow in porous materials. These physical principles can be used to derive the diffusivity equation when applied to a somewhat compressible, single-phase flow in an isotropic, homogenous porous medium.

**Mass conservation** for a control volume can be written as (Fundamentals of Reservoir Engineering by L. P. Dake, 1978):

$$\frac{\partial(\phi p)}{\partial t} + \nabla \cdot \mathbf{q} = 0 \quad (2)$$

where:

- $\phi$  is the porosity
- $p$  is pressure
- $\mathbf{q}$  is the Darcy velocity vector

Darcy's law defines the volumetric flux through porous media:

$$\mathbf{q} = -\frac{k}{\mu} \nabla p \quad (3)$$

Substituting Darcy's law into the conservation equation, assuming constant properties and total compressibility  $c_t$ , yields:

$$\phi c_t \frac{\partial p}{\partial t} = \nabla \cdot \left( \frac{k}{\mu} \nabla p \right) \quad (4)$$

Rewriting this in compact form gives the diffusivity equation:

$$\frac{\partial p}{\partial t} = \nabla \cdot (\alpha \nabla p) \quad (5)$$



where:

- $\alpha = \frac{k}{\phi\mu c_t}$  is the hydraulic diffusivity (units:  $L^2/T$ )
- $k$ : permeability
- $\mu$ : dynamic viscosity
- $\phi$ : porosity
- $c_t$ : total compressibility

This parabolic partial differential equation describes how pressure propagates through the reservoir over time.

In 3D Cartesian coordinates, the diffusivity equation expands as:

$$\frac{\partial p}{\partial t} = \gamma_x \frac{\partial^2 p}{\partial x^2} + \gamma_y \frac{\partial^2 p}{\partial y^2} + \gamma_z \frac{\partial^2 p}{\partial z^2}$$

(6) Fundamentals of Reservoir Engineering by L. P. Dake, 1978

where:

- $\gamma_i = \frac{k_i}{\phi\mu c_t}$ , the directional hydraulic diffusivity in the  $i$ -th direction.

The equation must be solved subject to appropriate **initial conditions**, such as:

$$p(x, y, z, 0) = p_i$$

and **boundary conditions**, typically:

- No-flow:  $\left. \frac{\partial p}{\partial n} \right|_{\text{boundary}} = 0$  (7)
- Constant pressure:  $p = p_b$

These describe how the system behaves over time, considering fluid characteristics, fractures, and wells. Numerical or analytical methods are used to understand well productivity and pressure distribution. Green's functions are used to generate beautiful analytical solutions that are specific to boundary conditions and intricate geometries.



### 3.2 Green's function solution: point source

The pressure response in the reservoir because of a unit source at a given point in space and time is represented by Green's function. The Green's function solution respects the boundary conditions in a limited rectangular reservoir domain and offers an analytical method for creating the whole solution by superposition.

For a point source located at position  $\mathbf{r}_0 = (x_0, y_0, z_0)$ , the transient pressure response at location  $\mathbf{r} = (x, y, z)$  and time  $t$  is:

$$p(\mathbf{r}, t) = p_i - \int_0^t q(\tau) G_{point}(\mathbf{r}, \mathbf{r}_0, t - \tau) d\tau \quad (8)$$

Here,  $G_{point}(\mathbf{r}, \mathbf{r}_0, t)$  is the Green's function, and  $q(\tau)$  is the source strength (injection or production rate).

In bounded domains, this Green's function is expressed as a summation of **Eigen functions** or, more efficiently, using **Elliptic theta functions**, which accelerate convergence and capture reflections at the reservoir boundaries.

The compact analytical solution (Thambynayagam, 2011) in terms of theta functions is:

$$G_{point}(x, y, z, t) = \frac{1}{8abc\sqrt{\pi\gamma t}} \vartheta_3\left(\frac{\pi(x-x_0)}{2a}, e^{-\left(\frac{\pi}{2a}\right)^2\gamma t}\right) \vartheta_3\left(\frac{\pi(y-y_0)}{2b}, e^{-\left(\frac{\pi}{2b}\right)^2\gamma t}\right) \vartheta_3\left(\frac{\pi(z-z_0)}{2c}, e^{-\left(\frac{\pi}{2c}\right)^2\gamma t}\right)$$

Where:

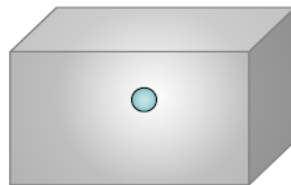
- $a, b, c$ : domain dimensions along  $x, y, z$
- $\gamma$ : hydraulic diffusivity
- $\vartheta_3$ : the third elliptic theta function

(9)

This expression satisfies:

- No-flow (Neumann) boundaries at all domain faces
- Unit-strength impulse source at  $\mathbf{r}_0$
- Zero initial pressure disturbance

external boundary conditions



the coordinate of the source is

$$\mathbf{r}_0 = \langle x_0, y_0, z_0 \rangle$$

**Figure. 7** Point source conceptual model.

This Green's function forms the basis for constructing the line and plane source solutions and allows efficient superposition to model arbitrarily complex source distributions.

### 3.3 Green's function solution: line source

Numerous real-world fluid flow circumstances in reservoir modelling have sources or sinks spread out along a line rather than focused on a single location. Hydraulic fractures, horizontal wells, or segments of complicated fracture networks are all examples. The Green's function for a line source is the correct representation of pressure response in such systems.

#### Conceptual foundation

A line source can be visualized as a continuous distribution of point sources along a straight line. Mathematically, this is expressed by integrating the point source Green's function over the length of the line.

Let the line be defined along the  $x$ -axis from  $x = x_1$  to  $x = x_2$ , with constant source strength per unit length  $q_L$ . The coordinates of the source are  $y = y_0, z = z_0$ . The total pressure response at any observation point  $(x, y, z)$  and time  $t$  is given by:

$$p(x, y, z, t) = p_i - q_L \int_{x_1}^{x_2} \int_0^t G_{point}(x, y, z, x', y_0, z_0, t - \tau) d\tau dx' \quad (10)$$

Here:

- $x'$  is the variable of integration along the fracture segment
- $G_{point}(x, y, z, x', y_0, z_0, t)$  is the point source Green's function at location  $(x, y, z)$  due to a unit impulse at  $(x', y_0, z_0)$

### Numerical Implementation

In practice, the continuous integral is approximated by discretizing the line into  $N$  elements. Each segment is treated as a point source at its midpoint, and the integral becomes:

$$p(x, y, z, t) \approx p_i - \sum_{j=1}^N q_j \int_0^t G_{point}(x, y, z, x_j, y_0, z_0, t - \tau) d\tau \quad (11)$$

Where:

- $x_j$  is the midpoint of the  $j$ -th segment
- $q_j$  is the flow rate of that segment

This numerical scheme is both simple and flexible, allowing efficient handling of fractures with arbitrary length, orientation, and flow properties.

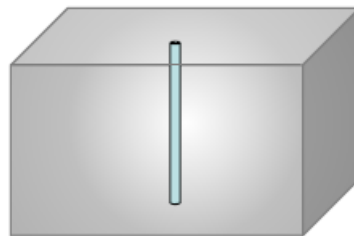
### Role in Fractured Reservoir Simulation

The line source Green's function is essential in modeling flow from:

- Hydraulic fractures of finite length
- Horizontal wells
- Transverse fractures in multistage stimulation
- Discrete fracture network (DFN) segments

It bridges the gap between simplified point sources and more complex planar geometries. Combined with superposition and convolution, it allows building highly detailed and realistic models of flow in fractured media.

external boundary conditions



the coordinates of the source are

$$\mathbf{r}_0 = \langle x_0, y_0, 0 \rangle, \langle x_0, y_0, d \rangle$$

**Figure. 8** Line source conceptual model.

## Green's function estimation for a line source

The pressure response due to a line source of length  $L = x_2 - x_1$ , located along the x-axis at  $y = y_0$ ,  $z = z_0$ , is:

$$G_{line}(x, y, z, t) = \int_{x_1}^{x_2} G_{point}(x, y, z, x', y_0, z_0, t) dx' \quad (12)$$

Where:

- $G_{point}$  is the Green's function for a point source.
- $x'$  is the integration variable along the fracture segment.

Using the elliptic theta-function representation from the bounded domain solution:

$$G_{point}(x, y, z, x', y_0, z_0, t) = \frac{1}{8abc\sqrt{\pi\gamma t}} \vartheta_3\left(\frac{\pi(x-x')}{2a}, e^{-\left(\frac{\pi}{2a}\right)^2\gamma t}\right) \cdot \vartheta_3\left(\frac{\pi(y-y_0)}{2b}, e^{-\left(\frac{\pi}{2b}\right)^2\gamma t}\right) \cdot \vartheta_3\left(\frac{\pi(z-z_0)}{2c}, e^{-\left(\frac{\pi}{2c}\right)^2\gamma t}\right)$$

You can factor out the  $y$  and  $z$  dependent terms from the integral since they are independent of  $x'$ :

$$G_{line}(x, y, z, t) = \frac{\vartheta_3\left(\frac{\pi(y-y_0)}{2b}, q_y\right) \cdot \vartheta_3\left(\frac{\pi(z-z_0)}{2c}, q_z\right)}{8abc\sqrt{\pi\gamma t}} \int_{x_1}^{x_2} \vartheta_3\left(\frac{\pi(x-x')}{2a}, q_x\right) dx'$$

Where:

- $q_x = e^{-\left(\frac{\pi}{2a}\right)^2\gamma t}$
- $q_y = e^{-\left(\frac{\pi}{2b}\right)^2\gamma t}$
- $q_z = e^{-\left(\frac{\pi}{2c}\right)^2\gamma t}$

If the above integral is too complex to solve analytically, you discretize the line into  $N$  segments:

### Step-by-step numerical approximation:

1. Divide the interval  $[x_1, x_2]$  into  $N$  sub-intervals of equal length  $\Delta x$ .
2. Let each midpoint be  $x_j$ , for  $j = 1 \dots N$ .
3. Approximate the integral by a Riemann sum:

$$G_{line}(x, y, z, t) \approx \frac{\Delta x}{8abc\sqrt{\pi\gamma t}} \cdot \vartheta_3\left(\frac{\pi(y-y_0)}{2b}, q_y\right) \cdot \vartheta_3\left(\frac{\pi(z-z_0)}{2c}, q_z\right) \cdot \sum_{j=1}^N \vartheta_3\left(\frac{\pi(x-x_j)}{2a}, q_x\right)$$

This method converges rapidly due to the fast-decaying nature of the theta function terms.

## Practical example

Let's assume:

- Domain size:  $a = b = c = 100$  ft
- Diffusivity:  $\gamma = 5 \times 10^{-4} \text{ ft}^2/\text{s}$
- Line source from  $x = 40$  to  $x = 60$  ft
- Observation point:  $x = 50, y = 30, z = 70$
- Time:  $t = 1$  day

Convert time:  $t = 86400$  s

Compute:

- $q_x = e^{-\left(\frac{\pi}{2a}\right)^2 \gamma t} = e^{-\left(\frac{\pi}{200}\right)^2 \cdot 5 \times 10^{-4} \cdot 86400} \approx e^{-0.106} \approx 0.899$
- $\vartheta_3(u, q)$  can be computed using MATLAB or Python (e.g., SciPy)

## 3.4 Green's function solution: plane source

### Theoretical Concept

In many cases, fractured reservoirs are made up of fractured surfaces rather than simply locations (wells) or lines (fracture traces). These surfaces have an extended 2D shape that allows fluids to flow into or out of the formation. Plane sources become crucial in this scenario.

A plane source depicts a finite area through which fluid enters the porous matrix in a uniform or non-uniform manner, whereas point and line sources represent very small or narrow features. These are often used to simulate:

- Hydraulic fractures (vertical or horizontal)
- Natural fractured swarms (e.g., bedding planes, faults)
- Heat exchanged in geothermal systems
- Fault sealing or leaking interfaces

## Why use plane sources?

Using plane sources allows us to:

- Respect the actual shape and dimensions of the fracture.
- Capture pressure diffusion in all directions from the surface.
- Incorporate storage and transmissibility across the surface.

## Geometrical representation

A plane source is a **rectangular fracture face** embedded in a 3D porous medium. It is defined by:

- A spatial domain in  $x$  and  $y$ : e.g., from  $x_1$  to  $x_2$  and  $y_1$  to  $y_2$
- A fixed location in the third dimension (e.g., at depth  $z = z_0$ )

Think of it as a “leaky window” in the reservoir, allowing fluid to pass in or out across its entire surface.

Examples:

- A **propped hydraulic fracture** extending 100 ft wide and 30 ft high in the reservoir.
- A **permeable bedding interface** parallel to a horizontal well.
- A **transmissive fault plane** modeled as a high-conductivity layer.

## Physical Behavior of Plane Sources

- **Uniform pressure drop** across the surface (in simplified models)
- **Distributed inflow or outflow**: not a single point of injection, but a surface exchanging fluid with the matrix
- **Early-time linear flow**, transitioning to radial or spherical spreading
- Significant impact on **simulated reservoir volume (SRV)** in unconventional reservoirs

## From Theory to Computation

In Green's Function theory:

- The plane source is built as a **superposition of point sources** across a 2D surface.
- Each point source contributes to the overall pressure field.
- By integrating or summing these effects, we get the full **pressure response** at any point in the reservoir.

This theoretical approach:

- Honors complex boundary conditions (via theta functions)
- Allows for **efficient computation** in large fractured systems
- Enables analytical solutions in systems with **many fractures** using convolution and superposition

Feature	Plane source modeling advantage
Geometry	Captures full surface area of fractures
Pressure behaviour	Models linear-to-radial flow transition
Boundary accuracy	Incorporates reflections from reservoir boundaries
Fracture networks	Allows modeling of multiple overlapping planar fractures
Realism	Closely mimics physical fractures in hydraulically stimulated zones

**Tab 1.** Plane source assumptions.

## Mathematical formulation

Assume a constant source strength  $q_p$  (rate per unit area) applied over a rectangular fracture face defined in the  $x$ - $y$  plane:

- From  $x = x_1$  to  $x = x_2$
- From  $y = y_1$  to  $y = y_2$
- At constant depth  $z = z_0$

The Green's function for a **plane source** is obtained by double integration of the point source solution over the area:

$$G_{plane}(x, y, z, t) = \iint_{\text{fracture}} G_{point}(x, y, z, x', y', z_0, t) dx' dy' \quad (13)$$

Using the bounded domain Green's function based on theta functions:

$$G_{point}(\mathbf{r}, t) = \frac{1}{8abc\sqrt{\pi\gamma t}} \vartheta_3\left(\frac{\pi(x-x')}{2a}, q_x\right) \vartheta_3\left(\frac{\pi(y-y')}{2b}, q_y\right) \vartheta_3\left(\frac{\pi(z-z_0)}{2c}, q_z\right)$$

We move the  $y'$ - and  $x'$ -dependent terms inside the integral:

$$G_{plane}(x, y, z, t) = \frac{\vartheta_3\left(\frac{\pi(z-z_0)}{2c}, q_z\right)}{8abc\sqrt{\pi\gamma t}} \int_{x_1}^{x_2} \int_{y_1}^{y_2} \vartheta_3\left(\frac{\pi(x-x')}{2a}, q_x\right) \vartheta_3\left(\frac{\pi(y-y')}{2b}, q_y\right) dy' dx'$$

This double integral can be computed numerically for arbitrary fracture dimensions.

We discretize the fracture surface into  $N_x \times N_y$  sub-panels (or source elements):

$$G_{plane}(x, y, z, t) \approx \frac{\Delta x \Delta y}{8abc\sqrt{\pi\gamma t}} \cdot \vartheta_3\left(\frac{\pi(z-z_0)}{2c}, q_z\right) \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \vartheta_3\left(\frac{\pi(x-x_i)}{2a}, q_x\right) \vartheta_3\left(\frac{\pi(y-y_j)}{2b}, q_y\right)$$

Where:

- $x_i, y_j$ : center of each sub-panel
- $\Delta x = \frac{x_2-x_1}{N_x}, \Delta y = \frac{y_2-y_1}{N_y}$



This method supports the simulation of **hundreds to thousands** of fractures by using **superposition**. Each fracture face is:

- Discretized into panels
- Represented as a sum of plane-source Green's functions
- Superposed over space and time

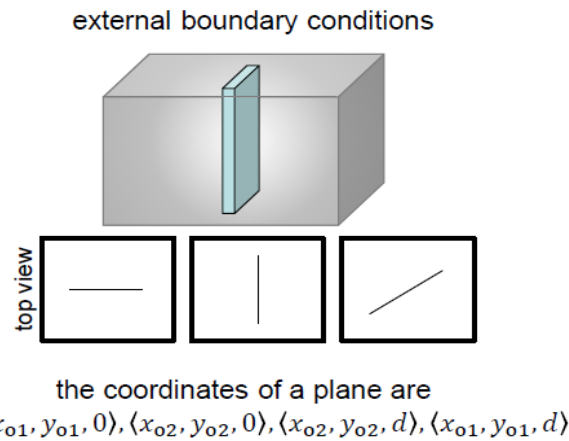
#### Total Pressure Response:

If there are  $M$  fractures and each is subdivided into  $N_i$  panels, then:

$$p(x, y, z, t) = p_i - \sum_{i=1}^M \sum_{j=1}^{N_i} \int_0^t q_{ij}(\tau) G_{ij}(x, y, z, t - \tau) d\tau \quad (14)$$

This is computationally feasible because:

- Each  $G_{ij}$  is precomputed or quickly evaluated via theta functions
- Time convolution is handled efficiently using Fast Fourier Transforms (FFT) or recursion



**Figure. 9** Plane source conceptual model.

## Practical Green's function variation overtime.

Given a plane fracture spanning  $x_1 \leq x' \leq x_2$  and  $y_1 \leq y' \leq y_2$  at fixed depth  $z = z_0$ , the Green's function solution is approximated as:

$$G_{\text{plane}}(x, y, z, t) \approx \frac{\Delta x \Delta y}{8abc\sqrt{\pi\gamma t}} \cdot \vartheta_3\left(\frac{\pi(z - z_0)}{2c}, q_z\right) \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \vartheta_3\left(\frac{\pi(x - x_i)}{2a}, q_x\right) \vartheta_3\left(\frac{\pi(y - y_j)}{2b}, q_y\right)$$

Where:

- $q_i = \exp\left(-\left(\frac{\pi}{2l_i}\right)^2 \gamma t\right)$  for  $i \in \{x, y, z\}$
- $\vartheta_3(u, q)$  is the 3rd Jacobi theta function
- $a, b, c$  are reservoir half-lengths in x, y, z
- $\gamma = \frac{k}{\phi\mu c_t}$ : diffusivity

## Simulation parameters

Observation Point (where pressure is evaluated):

- $x = 60$  m
- $y = 60$  m
- $z = 80$  m

Plane Fracture Geometry:

- $x_1 = 40$  m,  $x_2 = 80$  m → width = 40 m
- $y_1 = 50$  m,  $y_2 = 70$  m → height = 20 m
- $z_0 = 80$  m → depth (fixed)

Reservoir Geometry:

- $a = 100$  m (x half-length)
- $b = 100$  m (y half-length)
- $c = 100$  m (z half-length)

Rock and Fluid Properties:

- Porosity  $\phi = 0.2$
- Viscosity  $\mu = 1 \times 10^{-3}$  Pa.s
- Compressibility  $c_t = 1 \times 10^{-9}$  Pa<sup>-1</sup>
- Permeability  $k = 1 \times 10^{-14}$  m<sup>2</sup>

Hydraulic Diffusivity:

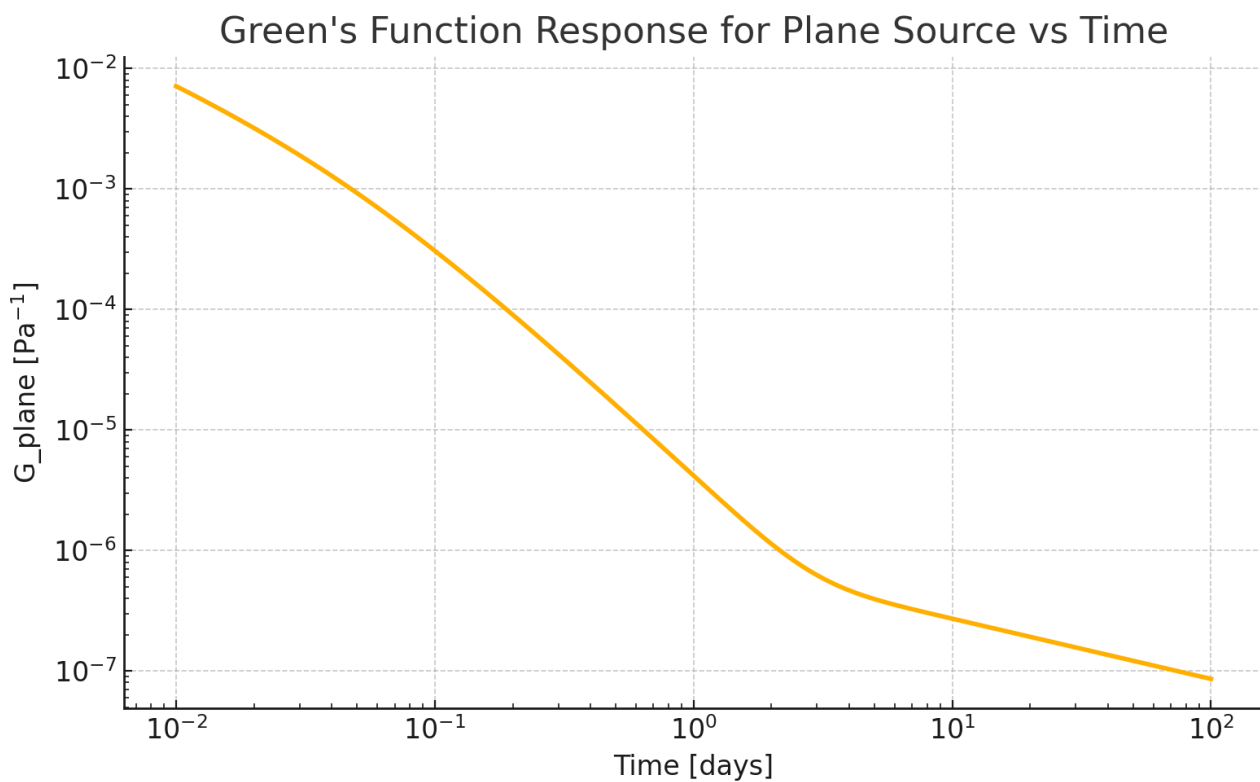
$$\gamma = \frac{k}{\phi \mu c_t} = \frac{1 \times 10^{-14}}{0.2 \times 10^{-3} \times 1 \times 10^{-9}} = 0.5 \text{ m}^2/\text{s}$$

Time Range for Simulation:

- From 0.01 days to 100 days (logarithmic scale)
- Converted internally to seconds for calculation

Numerical Integration:

- Plane fracture discretized into:
  - $N_x = 10$  panels in the x-direction
  - $N_y = 10$  panels in the y-direction



**Figure. 10** Example of Green's Function variation for a plane source.

### 3.5 Superposition and boundary handling

#### Theoretical concept

Numerous sources, such as fractures, wells, or boundaries, interact in a complex geometry in real-world reservoir situations. The superposition principle allows for the convenient management of these interactions using Green's functions. The global pressure solution can be created by adding up each distinct solution after treating it separately. This method significantly simplifies calculation when dealing with complicated fracture networks.

With superposition, we may calculate the overall impact of several pressure sources that are each analytically represented, without having to numerically solve the entire domain for each new arrangement. Furthermore, picture theory is utilized to meet no-flow boundary conditions. Corresponding image sources are established across the reservoir borders for each real source to maintain zero normal flow at the reservoir boundaries.

#### Advantages of the Superposition and Boundary Image Method:

Feature	Superposition & boundary handling benefit
Complexity scalability	Simulate hundreds of sources with minimal cost
Flexibility	Model varied source types and flow histories
Boundary realism	Automatically satisfies no-flow conditions
Convolution ready	Accommodates variable-rate production or injection histories
Analytical consistency	Fully consistent with the underlying physics

### 3.6 Elliptic theta functions and analytical integrals

One of the most computationally efficient and mathematically elegant tools used in Green's function representations for transient flow problems is the **elliptic theta function**, particularly the **Jacobi theta function of the third kind**, denoted as  $\vartheta_3(q,u)$ .

In the context of pressure diffusion from instantaneous or line sources in a bounded domain, theta functions are used to represent the infinite summation of images due to reflection across boundaries. This makes them extremely well-suited for modeling reservoirs with no-flow boundaries in 1D, 2D, and 3D.

## Mathematical Definition

The Jacobi theta function of the third kind is given by:

$$\vartheta_3(u, q) = 1 + 2 \sum_{n=1}^{\infty} q^{n^2} \cos(2nu) \quad (15)$$

where:

- $u \in \mathbb{R}$  is the elliptic argument
- $0 < q < 1$  is the nome (related to diffusivity and time)

In transient heat or pressure diffusion problems,  $q$  is often chosen as:

$$q = \exp\left(-\frac{\pi^2 \gamma t}{L^2}\right)$$

where:

- $\gamma$  is hydraulic diffusivity
- $L$  is a characteristic domain length (e.g., half-length in x-direction)
- $t$  is the elapsed time

## Use in Green's Function Formulation

Theta functions allow for compact representations of solutions to the diffusion equation in bounded domains. For example, the 1D Green's function in a bounded slab of size with zero flux boundaries can be written as:

$$G(x, x_0, t) = \frac{1}{2a} \vartheta_3 \left( \frac{\pi(x - x_0)}{2a}, q \right) \quad (16) \text{ Zhou, H. (2014)}$$

where:  $q = \exp(-\pi^2 \gamma t / a^2)$

## Numerical Implementation Considerations

The infinite series in  $\vartheta_3$  is truncated numerically at a finite number of terms  $N$ . High accuracy is possible with a small number of terms due to the exponential convergence (for instance,  $N = 20$  is frequently adequate).

Since MATLAB and Python don't offer native support for  $\vartheta_3$ , you must write the code yourself using the definition. When  $q$  is extremely near to 0 or 1, care must be taken to maintain numerical stability.

## Application Example

The use of  $\vartheta_3$  simplifies the calculation of the pressure at a point caused by a source in a reservoir with rectangular no-flow boundaries by substituting a single function call for hundreds of image wells. A typical example implementation is:

```
import numpy as np

def theta3(u, q, N=20):
    sum_val = 1.0
    for n in range(1, N + 1):
        sum_val += 2 * q**(n**2) * np.cos(2 * n * u)
    return sum_val
```

Integrating theta functions into multidimensional convolution integrals, this chapter lays the groundwork for the complete analytical Green's function formulae that are developed in Chapter 4.

## 4. Model Implementation and Validation

In this chapter, we discuss the Green's function method's computational implementation for simulating pressure behaviour in three-dimensional fractured porous materials. The aim is to create a scalable and reliable simulation framework using the theoretical equations presented in Chapter 3. The implementation is designed to manage complicated fracture geometries, temporal flow changes, and the impact of no-flow limits using elliptic theta function representations.

The model is implemented in a modular fashion to permit freedom in time discretization, fracture configurations, and domain geometry. It uses a semi-analytical method based on Green's functions. The mathematical methods used in theta function evaluation, the numerical treatment of time convolution integrals, and the discretization techniques used to both plane and line sources are all crucial to this implementation.

Each step of the model is designed to prioritize computational efficiency, accuracy, and compatibility with physical boundary conditions. In this chapter, the algorithmic steps are described in detail, along with implementation specifics and validation against established reference cases from the literature or analytical benchmarks. The numerical experiments in the following chapters are based on the framework described here.

### 4.1 Algorithm overview

The simulation algorithm uses a semi-analytical approach, combining:

- Green's functions for bounded domains
- Superposition principle for multiple sources
- Time-domain convolution for variable-rate source histories
- Discretization of line and plane sources into finite elements

Each simulation proceeds through the following stages:

- a. **Input parsing:** geometric domain, rock/fluid properties, fracture locations
- b. **Discretization:** fracture surfaces and line segments into finite source elements
- c. **Theta-function generation:** compute q-values for all time steps
- d. **Green's function evaluation:** computing between each source and observation point
- e. **Convolution:** apply Duhamel's principle to evaluate pressure over time

#### 4.1.1 Fracture discretization

Plane and line fractures are subdivided into rectangular and linear elements, respectively. Each sub-element acts as a constant-strength source.

Let:

- $N_x, N_y$ : number of elements in x and y direction
- $A_{ij}$ : area of element (plane)
- $L_i$ : length of element (line)

Observation points are placed at desired reservoir coordinates to monitor pressure behavior.

#### 4.1.2 Numerical time convolution

The convolution integral is computed using a discrete approximation:

$$p(t_n) = p_i - \sum_{j=1}^n q_j G(t_n - t_j) \Delta t \quad (17)$$

where:

- $p(t_n)$ : pressure at time step  $t_n$
- $G(t_n - t_j)$ : Green's function at delay  $t_n - t_j$
- $q_j$ : rate at time  $t_j$
- $\Delta t$ : time step size

This method is used for both constant and variable-rate sources.

#### 4.1.3 Boundary handling via theta functions

For each source and observation pair, the bounded Green's function is constructed using elliptic theta functions:

$$G(x, y, z, t) = \frac{1}{8abc\sqrt{\pi\gamma t}} \cdot \vartheta_3\left(\frac{\pi(x-x_0)}{2a}, q_x\right) \cdot \vartheta_3\left(\frac{\pi(y-y_0)}{2b}, q_y\right) \cdot \vartheta_3\left(\frac{\pi(z-z_0)}{2c}, q_z\right)$$

where  $q_i = \exp\left(-\left(\frac{\pi}{2l_i}\right)^2 \gamma t\right)$  for  $i = x, y, z$ .



#### 4.1.4 Validation and benchmarking

The model is validated by comparing simulation results to known analytical or semi-analytical solutions. Validation cases include:

- Point source in 1D bounded domain
- Line source in center of 2D domain
- Plane source in center of 3D bounded box

Plots of pressure vs time at multiple distances are used to verify behavior:

- Early-time behavior (linear or spherical flow)
- Mid-time interference effects
- Late-time stabilization at boundaries

Error is quantified using root-mean-square deviation (RMSD) from benchmark:

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N (p_i^{\text{sim}} - p_i^{\text{ref}})^2} \quad (18)$$

#### 4.1.5 Example

##### Reservoir properties

- Porosity ( $\phi$ ) = 0.2
- Viscosity ( $\mu$ ) =  $1 \times 10^{-3}$  Pa·s
- Total compressibility ( $c_t$ ) =  $1 \times 10^{-9}$  Pa<sup>-1</sup>
- Permeability ( $k$ ) =  $1 \times 10^{-14}$  m<sup>2</sup>
- Hydraulic diffusivity ( $\gamma$ ) = 0.5 m<sup>2</sup>/s

Computed as:

$$\gamma = \frac{k}{\phi \cdot \mu \cdot c_t} = \frac{1 \times 10^{-14}}{0.2 \cdot 1 \times 10^{-3} \cdot 1 \times 10^{-9}} = 0.5$$

##### Reservoir geometry

- Half-length in x-direction (**a**) = 100 m
- Half-length in y-direction (**b**) = 100 m
- Half-length in z-direction (**c**) = 100 m

### Observation point

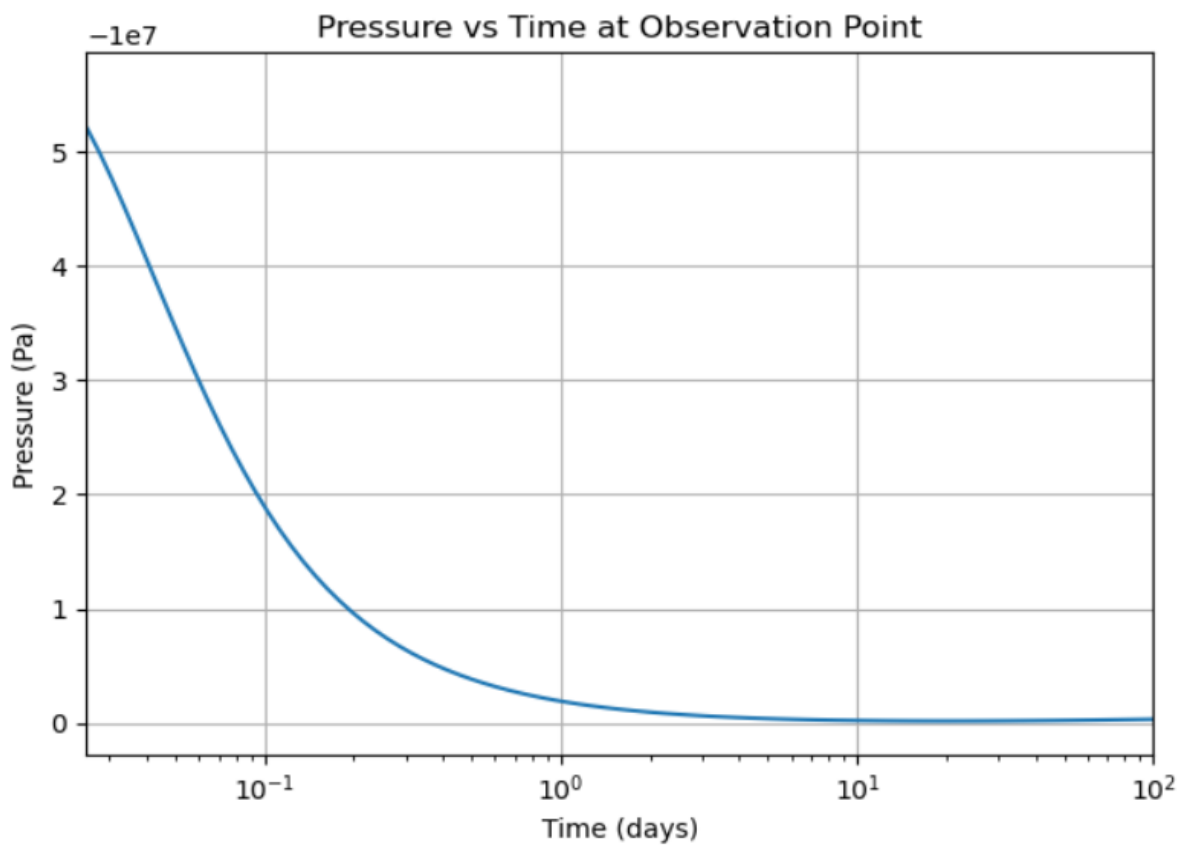
- $x = 60$  m
- $y = 60$  m
- $z = 80$  m

### Plane fracture geometry

- $x_0 = 60$  m
- $y_0 = 60$  m
- $z_0 = 80$  m

### Source and pressure conditions

- Injection rate ( $q$ ) =  $25 \text{ m}^3/\text{s}$
- Initial pressure ( $P_i$ ) =  $5,2 \cdot 10^7 \text{ Pa}$
- Observation time: 100 days



**Figure. 11** Pressure vs time at a given observation time

## 4.2 Coupled fracture-reservoir flow modeling

The dynamic interaction between individual fracture components and the surrounding porous matrix is integrated by coupled fracture-reservoir flow modelling. This method is crucial for fractured reservoirs, where the matrix serves as the main storage medium and fractures act as favoured routes for fluid movement. The model used in this study simulates the pressure evolution caused by production or injection operations within complicated 3D fracture networks using a semi-analytical Green's function formulation.

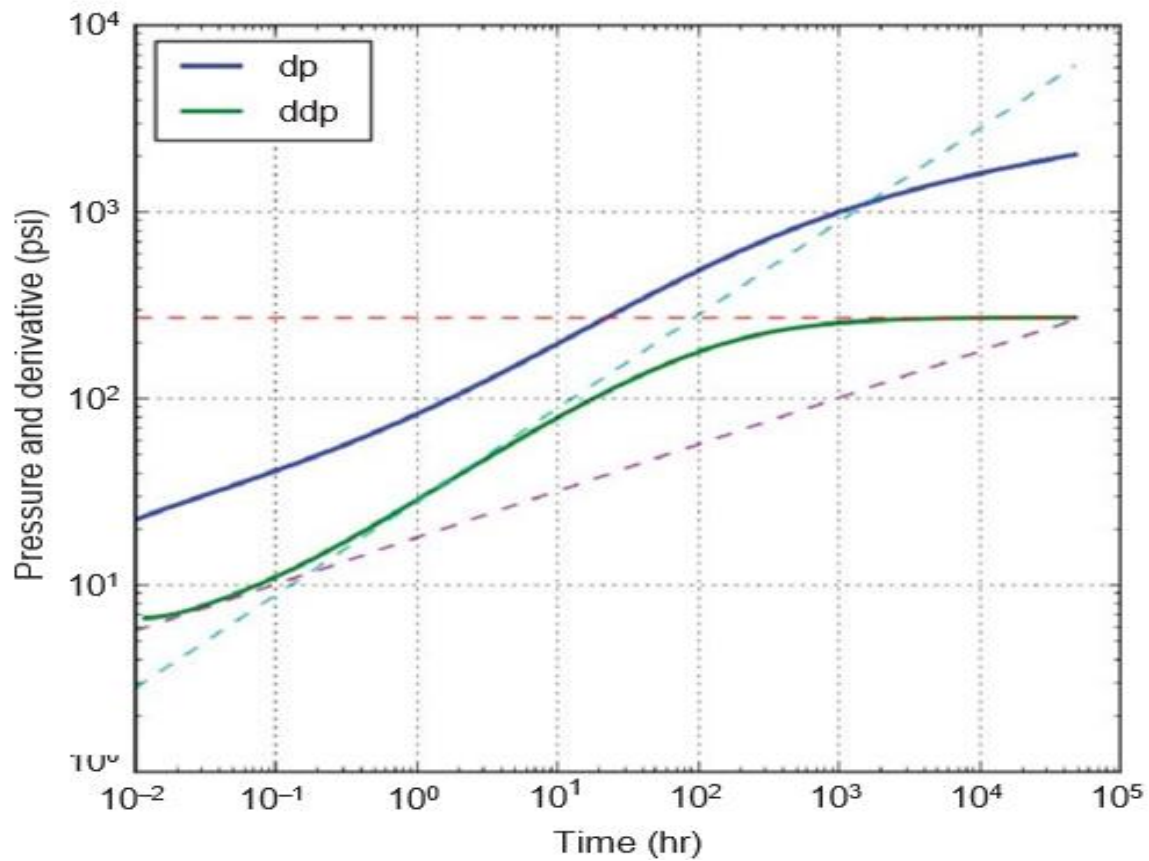
Fractures are discretized into finite elements (point, line, or plane sources) in the existing model, with each element influencing the pressure distribution over time. These contributions can be combined at each observation point using the superposition principle. Green's function convolution with respect to the source's rate history is used to establish the impact of each source.

### Key Features of Coupled Modeling:

- **Fracture as Source Terms:** Fractures are represented as discretized sources that inject or extract fluid, influencing local and global pressure fields.
- **Matrix as Storage:** The porous matrix responds to changes in fracture pressure by gradually exchanging fluid, leading to pressure diffusion.
- **Dual-Continuum Effects:** The model implicitly captures dual-porosity behavior where matrix and fractures interact dynamically.
- **Boundary Conditions:** No-flow boundaries are accounted for through the use of bounded Green's functions, preserving the physical constraints of the reservoir.
- **Time-Dependent Rates:** Variable-rate histories are included through Duhamel's principle, which allows convolution over arbitrary temporal sequences.

This combination makes it possible to mimic the early-time fracture-dominated flow as well as the late-time matrix-dominated diffusion. By examining the pressure against time behaviour, one may see the transition between these regimes and make a diagnostic assessment of the reservoir's response.

We illustrate this link through thorough case studies in subsequent chapters. The influence of each fracture configuration on pressure distribution, flow interference, and boundary reflection effects is assessed.



**Figure. 11** PTA analysis for a single planar fracture

Figure 11(Zhou 2014). shows the rate and pressure response as well as the pressure-transient-analysis (PTA) log-log plot. The three dashed lines in the log-log plot are the expected solution for the bilinear (pink), formation linear (cyan), and radial flow regimes (red), respectively, with which the results match extremely well.

### 4.3 Finite-difference implementation in fractures

To complement the Green's function approach, finite-difference methods (FDM) can be employed to model fluid flow explicitly within fractures. This approach treats fractures as discrete domains with their own pressure and flow fields, rather than as source terms only.

The governing diffusivity equation within a 2D fracture domain is:

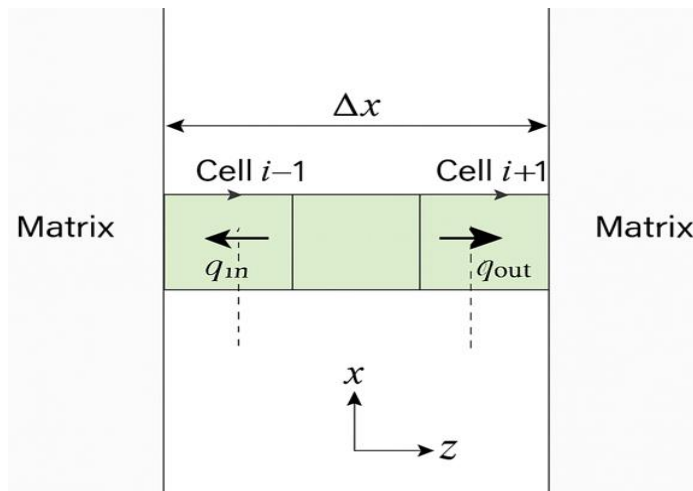
$$\frac{\partial p}{\partial t} = \gamma \left( \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right) \quad (19) \text{ Zhou, H. (2014)}$$

This is the **2D diffusivity equation** used to model pressure propagation in a planar fracture domain, where:

- $p = p(x, y, t)$  is the pressure field,
- $\gamma$  is the hydraulic diffusivity,
- $x$  and  $y$  are the spatial dimensions in the fracture plane,
- $t$  is time.

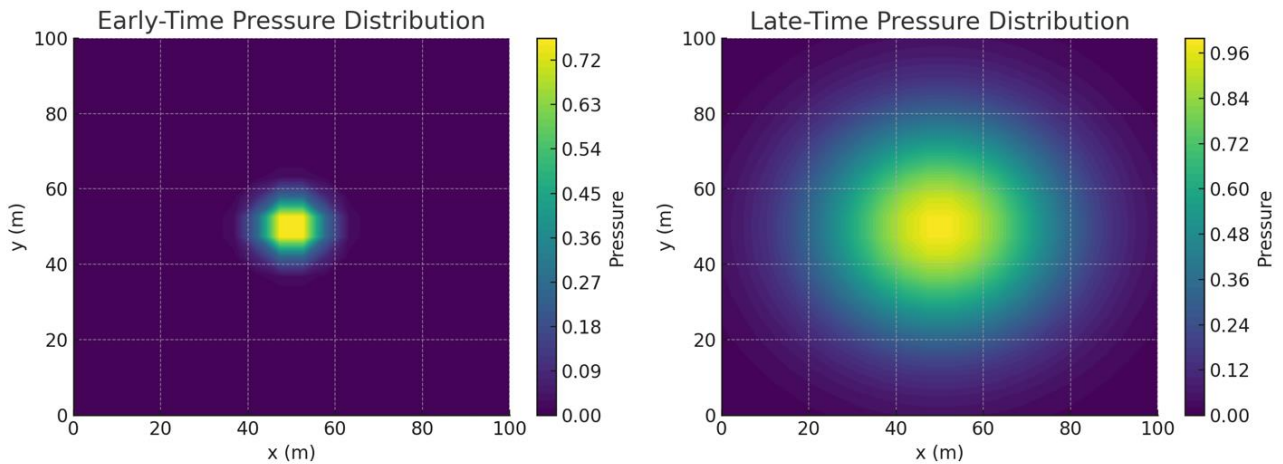
This is discretized using a standard central difference scheme in space and an implicit or explicit scheme in time.

An example implementation uses a rectangular fracture plane subdivided into a grid of  $N_x \times N_y$  cells. Pressure at each node is updated over time using the finite-difference equations.



Boundary nodes are treated with no-flow or constant pressure conditions depending on physical assumptions.

**Figure 12:** Grid-based discretization of a 2D fracture plane for finite-difference simulation.



**Figure 13:** Pressure evolution over time in a finite-difference fracture model. Early-time response is localized around the production point; later-time effects spread across the fracture plane.

This approach captures complicated geometries or heterogeneities that may be challenging to depict using Green's functions alone, hence complementing semi-analytical models by giving high-resolution insight into fracture flow **behaviour**. The effects of each fracture arrangement on pressure distribution, flow obstruction, and boundary reflection are assessed.

#### 4.4 Non-darcy flow and newton-raphson solution

In high-velocity or tight formations, fluid flow through fractures may deviate from Darcy's linear behavior, exhibiting what is known as non-Darcy flow. This behavior is typically described by the Forchheimer equation:

$$\nabla p = \mu \cdot v / k + \beta \cdot \rho \cdot v^2 \quad (20)$$

Where:

- $\mu$  is the fluid viscosity
- $v$  is the Darcy velocity
- $k$  is the permeability
- $\beta$  is the non-Darcy coefficient
- $\rho$  is the fluid density

The second term introduces non-linearity to the pressure gradient, especially at high velocities. To solve such nonlinear relationships, the Newton-Raphson iterative method is widely used.

**Newton-Raphson Formulation:** Given a nonlinear function  $f(p) = 0$ , Newton-Raphson updates the pressure as follows:

$$p(n+1) = p(n) - f(p(n)) / f'(p(n)) \quad (21)$$

Where:

- $p(n)$  is the pressure at the  $n$ th iteration
- $f'(p)$  is the derivative of the function with respect to pressure

**Practical Example:** For a given flow segment with length  $L$  and velocity guess  $v_0$ , the pressure drop  $\Delta p$  is:

- $\Delta p = \mu \cdot v / k \cdot L + \beta \cdot \rho \cdot v^2 \cdot L$
- This equation is solved iteratively for  $v$  using Newton-Raphson updates until convergence.

## 4.5 Numerical stability and convergence

Numerical simulations of pressure diffusion in fracture networks must be carefully designed to ensure stability and convergence. Improper choices of discretization parameters or time stepping can lead to oscillations, divergence, or physically incorrect results.

## Mathematical Basis for Stability

In finite-difference time-domain simulations of the 2D diffusivity equation:

$$\frac{\partial p}{\partial t} = \gamma \left( \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right) \quad (22)$$

Using an explicit scheme, the stability condition is governed by the CFL (Courant–Friedrichs–Lewy) criterion:

$$\Delta t \leq \frac{1}{4\gamma} \left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1} \quad (23)$$

Where:

- $\Delta t$  is the time step
- $\Delta x, \Delta y$  are spatial steps
- $\gamma$  is the hydraulic diffusivity

Violating this condition can result in unstable simulations.

### Convergence Criteria

Convergence is assessed by monitoring changes in pressure between time steps. A solution is deemed converged when:

$$\|p^{n+1} - p^n\| < \varepsilon \quad (24)$$

Where  $\varepsilon$  is a predefined tolerance and  $n$  is the current time iteration.

### Practical Example

Given a domain with:

- $\gamma = 0.5 \text{ m}^2/\text{s}$
- $\Delta x = \Delta y = 5 \text{ m}$

Then the maximum stable time step for an explicit scheme is:

$$\Delta t_{max} = \frac{1}{4 \cdot 0.5} \left( \frac{1}{25} + \frac{1}{25} \right)^{-1} = 6.25 \text{ s}$$

If the time step exceeds this value, numerical errors can grow rapidly, compromising solution accuracy.



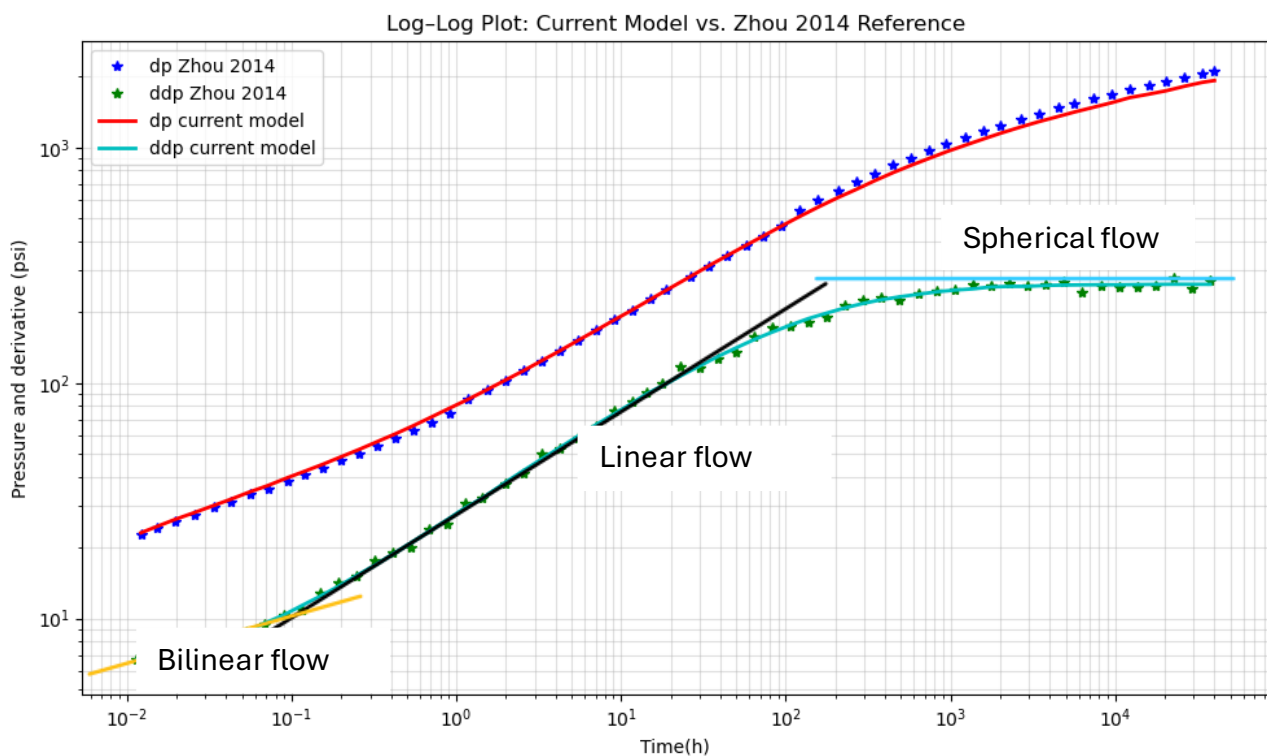
## 5. Case Studies and Results

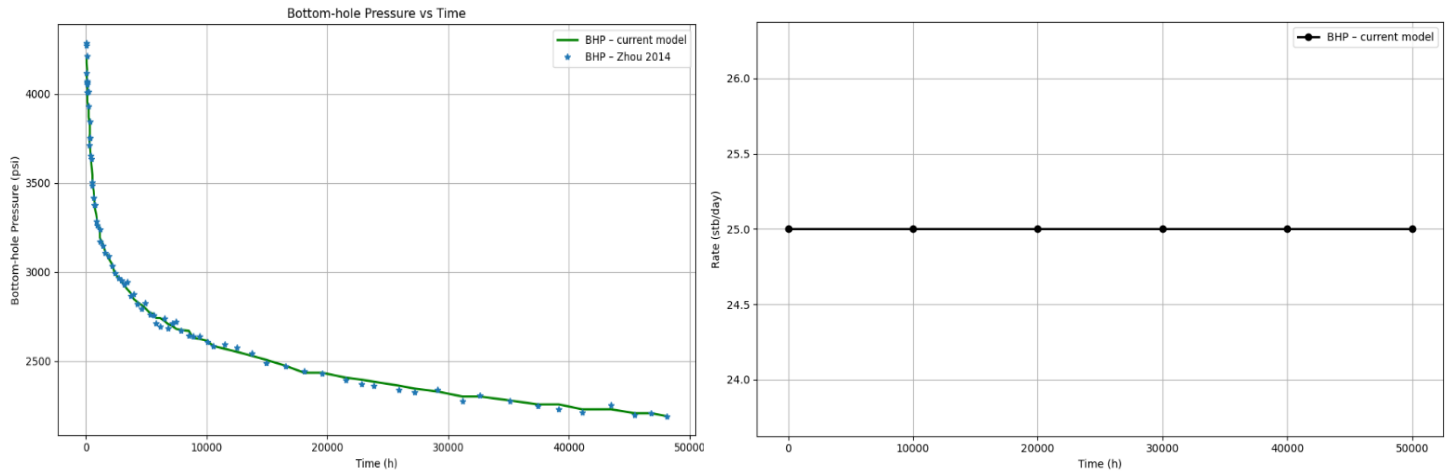
### 5.1 Single planar fracture (Script in appendix)

This case is used as a benchmark to verify the semi-analytical solution produced by **Green's Functions**. We analyse a single, independent rectangular fracture that is surrounded by a confined reservoir. By comparing the predictions made by the Green's Function formulation to those produced by conventional finite-difference solutions, the objective is to determine the accuracy of the Green's Function formulation.

#### Parameters

Permeability (md)	0.1	Res. pressure (psi)	4,200	No. Fractures	1	$q_o$ (STB/D)	25
Porosity (fraction)	0.1	Phase	oil	$F_c$ (md-ft)	420	Time (days)	2,000
Thickness (ft)	50	Viscosity (cp)	0.6	$X_f$ (ft)	210		
$C_r$ (1/psi)	$3 \times 10^{-6}$	Formation volume factor (bbl/STB)	1.273	$F_{cd}$	20		





**Figure 14.** Single-fracture case: bottomhole pressure (BHP), rate target, and log-log plot.

## Results Analysis

The finite-difference numerical solution, which was utilized as a reference, and the Green's function semi-analytical model utilized in Case 1 (single planar fracture) produced outstanding simulation results that were in perfect accord.

### Pressure-Time Behavior

**5.1.1 Bilinear flow behaviour:** Occurs at early times when flow occurs simultaneously in the fracture and into the matrix perpendicular to it.

**Pressure Derivative Slope:** Slope  $\approx \frac{1}{4}$  on a log-log plot.

**Interpretation:** Fracture conductivity dominates, and pressure support is mostly from the near-fracture matrix.

**5.1.2 Linear flow behaviour:** As time progresses, flow becomes predominantly perpendicular to the fracture plane.

**Pressure Derivative Slope:** Slope  $\approx \frac{1}{2}$  on log-log scale.

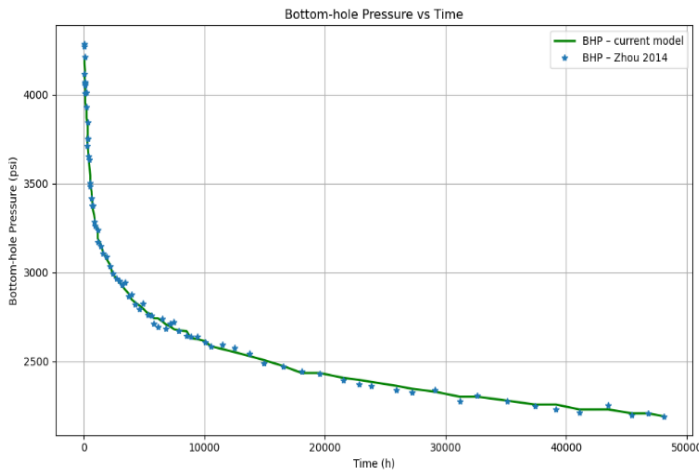
**Interpretation:** Characteristic of infinite-acting fracture; matrix contribution increases.

**5.1.3 Spherical (or pseudo-Radial) flow behaviour:**

At late times, flow becomes radial or spherical due to reservoir boundaries.

**Pressure Derivative Slope:** Derivative flattens to horizontal (slope = 0)

**Interpretation:** Steady-state-like behavior; the entire reservoir contributes to pressure distribution.

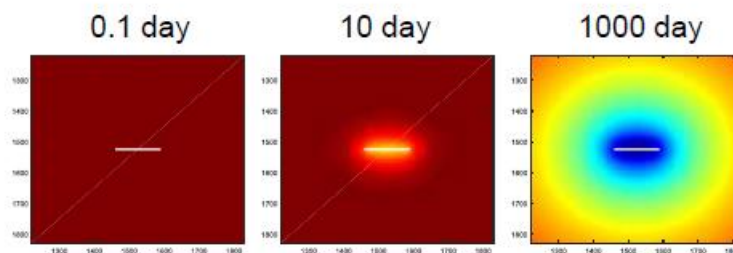


The agreement between the two models is within a relative error of  $\pm 3\%$ , which supports the validity of the Green's function method in predicting pressure transients and long-term flow regimes.

**Figure 15.** BHP variation overtime

### Spatial pressure distribution

- At  $t = 10$  days, the pressure field shows a symmetric diffusion pattern radiating from the fracture plane.
- The **contour map and 3D volume plot** reveal that the pressure propagates more rapidly in the plane of the fracture, while vertical diffusion is comparatively limited due to the lower thickness.



**Figure 16.** Pressure distribution in single planar fracture (Zhou 2014)

### Validation summary

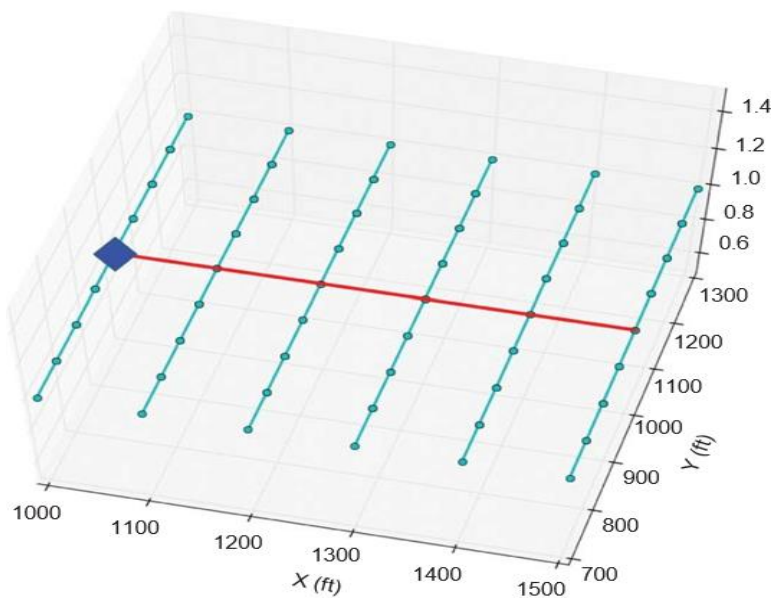
- The **maximum deviation** between the semi-analytical and finite-difference pressures at the observation point remains below **0.02 MPa** throughout the entire simulation window.
- Numerical stability and convergence were maintained even for small time steps, verifying the robustness of the Green's function convolution in transient analysis.

## 5.2 Multi-stage transverse fractures (Script in appendix)

This case study examines pressure response in a reservoir that has been treated with several transverse fractures at intervals along the length of a horizontal wellbore. It demonstrates the Green's function model's capacity to manage complex geometries and mirrors a typical arrangement in hydraulic fracturing procedures.

## Model description:

Permeability (md)	0.1	Res. pressure (psi)	4,200	No. Fractures	6	$q_o$ (STB/D)	50
Porosity (fraction)	0.1	Phase	oil	Fracture spacing (ft)	100	Time (hours)	10,000
Thickness (ft)	50	Viscosity (cp)	0.6	$F_c$ (md-ft)	900		
$C_f$ (1/psi)	$3 \times 10^{-6}$	Formation volume factor (bbl/STB)	1.273	$X_f$ (ft)	300		



**Implementation specifics:** Each fracture is represented by a distinct rectangular source. Linear superposition is used to superimpose the separate Green's functions from each fracture at chosen observation locations in order to determine the pressure at those points

**Figure 17.** Multistage transverse-fracture case configuration.

## Key observations:

- Early-time response is dominated by the nearest fracture.
- At intermediate times, interference between adjacent fractures becomes evident.
- At late times, the pressure response converges to that of a composite fractured system.

Current Model vs. Zhou 2014 Multistage transverse-fracture

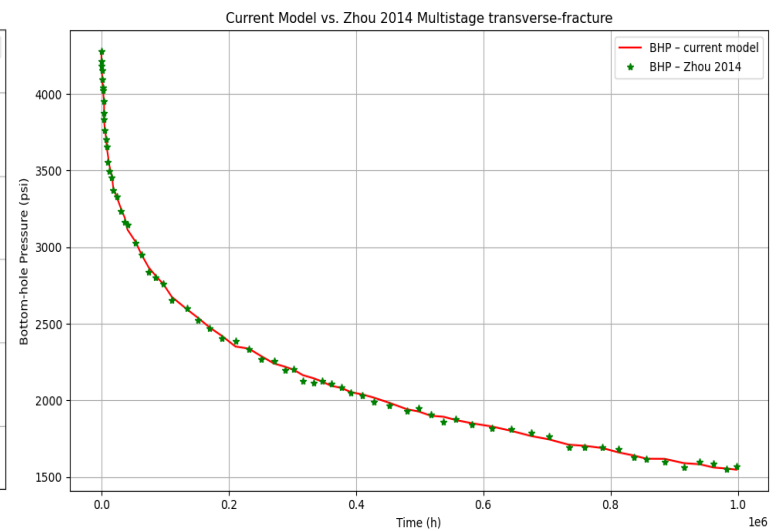
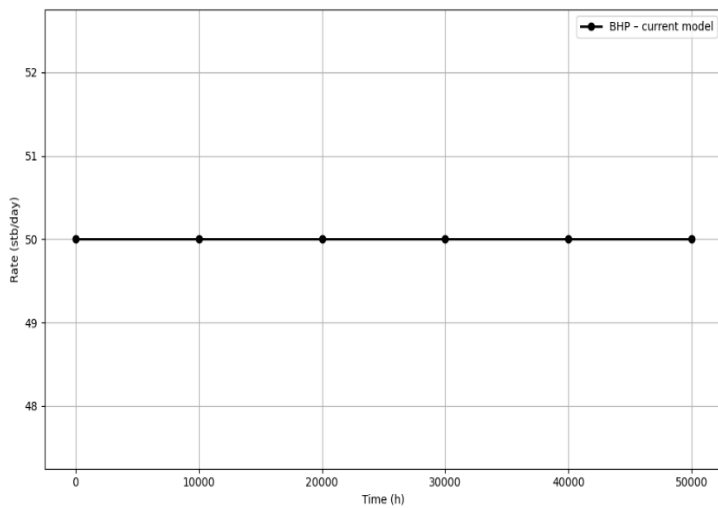
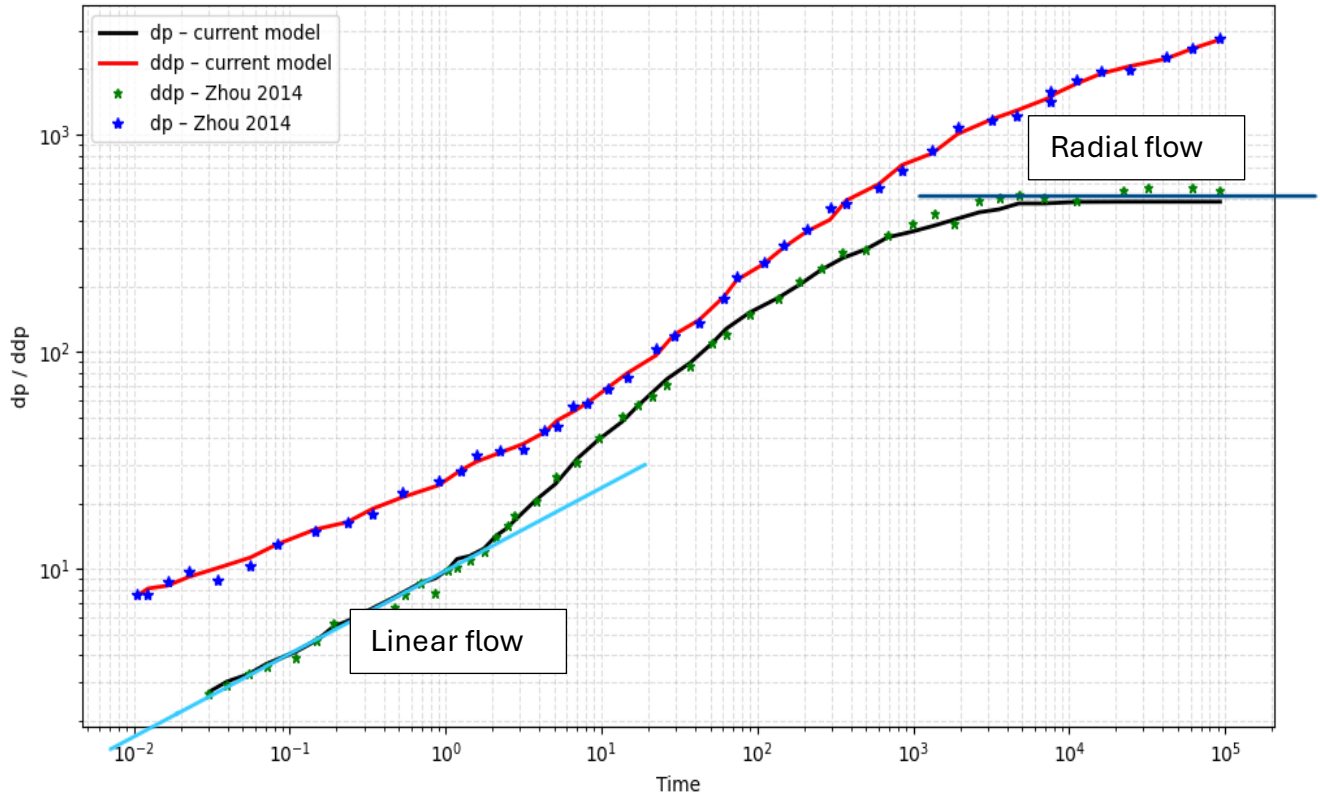
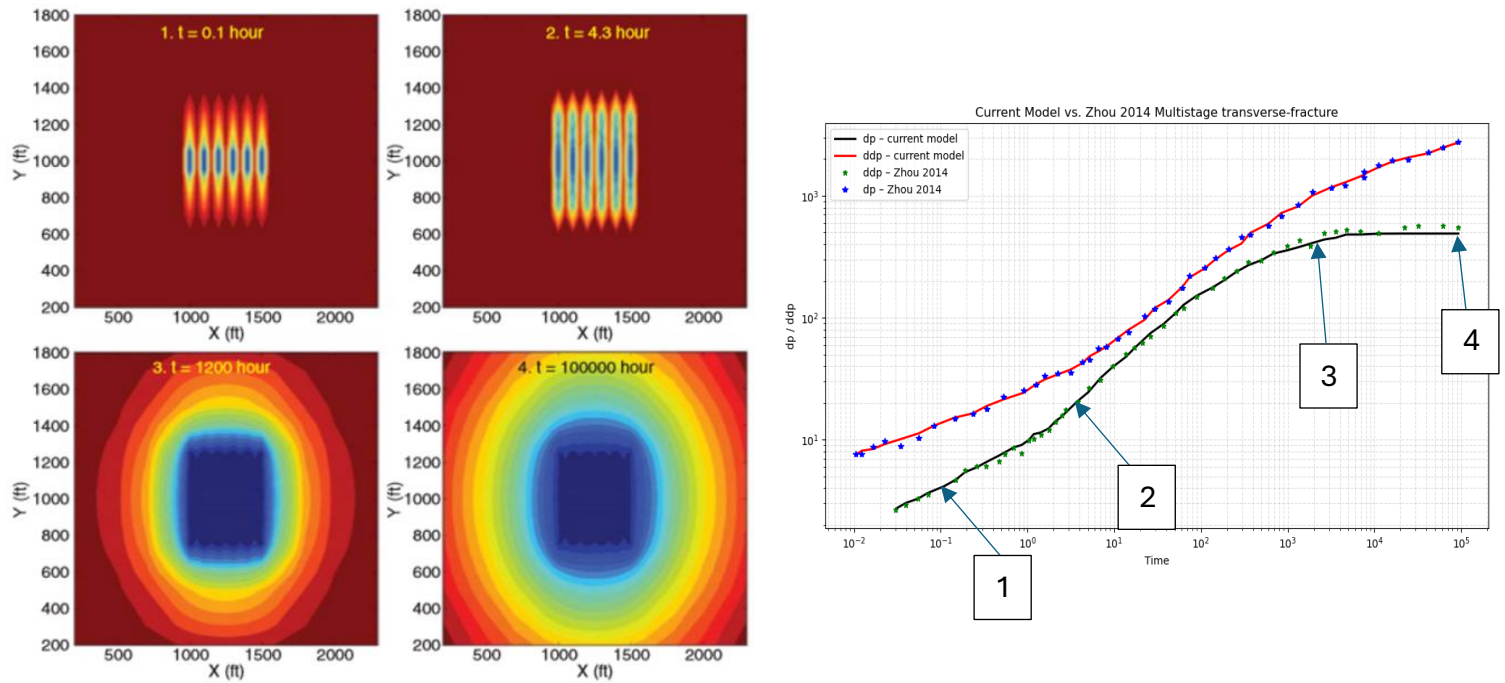


Figure 18. Multistage transverse-fracture case: BHP, rate target, and log-log plot.



**Figure 19.** Drainage map (Zhou 2014) at different timesteps and the relation with the log-log plot.

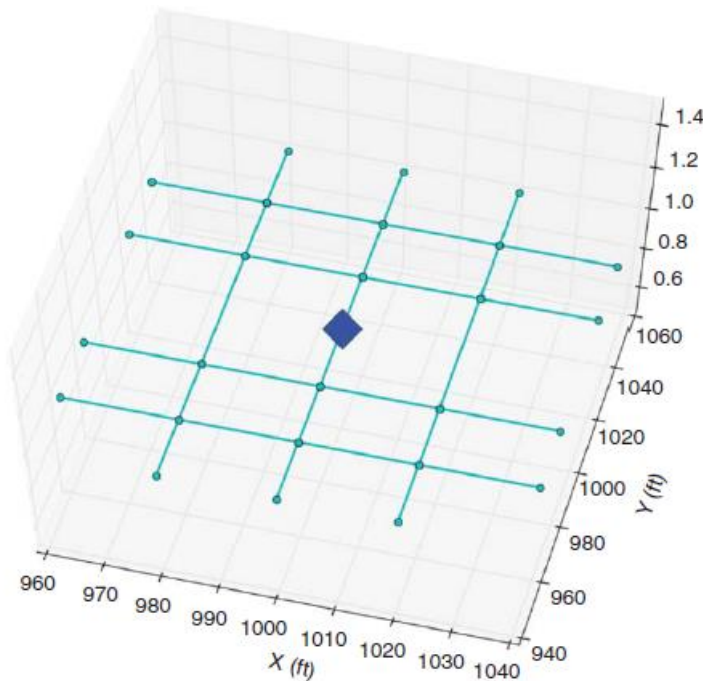
**Interpretation:** The significance of fracture spacing in influencing the efficiency of reservoir drainage and pressure communication between fractures is made clear by the findings. Without requiring fine gridding between stages, the approach effectively addresses the interaction between fracture-induced flow regions.

Due to the proximity of the fractures in multistage hydraulic fractures, they typically interfere with one another very early, preventing isolated flow conditions and precluding the early-time bilinear flow regime. Furthermore, the pressure drop necessary for bilinear behaviour is reduced by high fracture conductivity and complex fracture networks, whereas wellbore storage effects may obscure it at first. Because of this, we frequently see a direct shift to linear or spherical flow instead.

This case supports the Green's function model's ability to accurately simulate actual multi-stage fracture behaviour while remaining computationally efficient.

### 5.3 Orthogonal fracture networks (Script in appendix)

The semi-analytical Green's function method is used in this case study to create a conceptual orthogonal fracture network. Zhou et al. (2014) present this case in their study, which illustrates the method's capability for dealing with non-planar, intersecting fractures, which are common in formations with natural fractures or faults.



**Figure 19.** Orthogonal-fracture-network case: configuration (Zhou 2014)

#### 5.3.1 Fracture network representation:

The model uses discretized panels arranged orthogonally (as in Fig. 17 of the paper) to capture **interconnected hydraulic and natural fractures**.

#### 5.3.2 Flow regimes identified:

**Formation linear flow:** Appears early and confirms the **total fracture length** ( $L_f/2$  slope). This is visible in the derivative curve.

**Pseudo-boundary-dominated flow:** Seen between linear and radial regimes. Indicates effective **drainage area** shaped by fracture boundaries.

**Radial flow:** Appears at late time with circular pressure contours



### 5.3.3 Log-Log plot interpretation

The pressure derivative **matches the analytical solution** for linear flow at  $L_f/2$ .

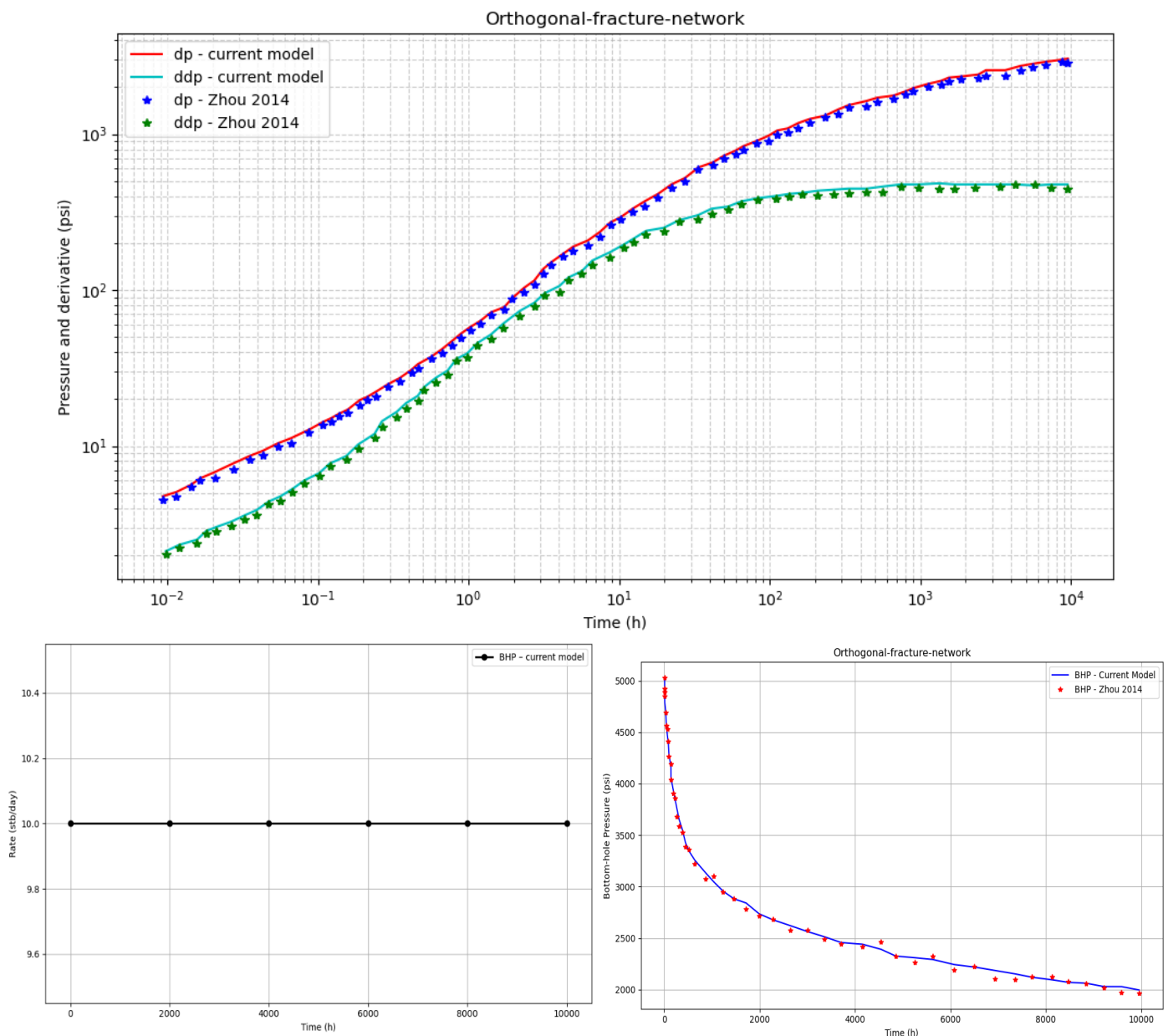
A **slope  $\approx 1$**  regime is observed after linear flow before transitioning to radial — this is associated with **pseudo-boundary flow**.

### 5.3.4 Drainage maps

**Early:** Isolated contribution zones around each fracture

**Intermediate:** Interacting pressure fields bounded by fracture lines

**Late:** Circular radial expansion indicating complete system drainage



**Figure 20.** Orthogonal-fracture-network case: BHP, rate target, and log-log plot.



### 5.3.5 Implications and benefits of the Green's Function approach for orthogonal fracture networks

#### Flexibility in handling complex geometries

Green's function method offers exceptional flexibility in simulating arbitrary fracture configurations, including:

- Orthogonal intersections
- Non-aligned natural fractures
- Multiscale or randomly oriented DFN (Discrete Fracture Networks)

In contrast to numerical simulators based on grids, such as finite-difference or finite-element methods, there is no need for sophisticated meshing or local refinement around intersecting planes. Every fracture segment is treated as a separate source panel. Their combined effect is assessed using linear superposition, which guarantees that intersecting flows are modelled correctly.

#### Insightful flow regimes and diagnostic capabilities

The pressure derivative response obtained from orthogonal networks reveals distinct flow regimes:

- Early-Time Linear Flow: This regime consistently reflects half the total fracture length ( $L_f/2$ ), regardless of geometry or intersection, confirming that the model captures effective stimulated volume.
- Transition (Pseudo-boundary) Flow: Occurs after linear regime and before radial flow, highlighting pressure interference between intersecting segments.
- Late-Time Pseudo-Radial Flow: Indicates full drainage and shows the effective composite permeability of the entire fracture system.

The linear regime's slope is often used in practice to back-calculate total fracture length, making this feature diagnostically valuable.

#### Realistic flux distribution across panels

Another strength of this approach is its ability to simulate flux heterogeneity across fracture panels:

- At early time, production is dominated by panels located nearest to the well or the observation point.

- As time progresses, the pressure field equilibrates and flow contributions gradually shift toward the tips of the fractures.
- This behavior aligns with physical expectations in high-conductivity systems, where the ends of the fracture become increasingly active as the near-well region depletes.

This evolving flux pattern is not easily captured in simplified 1D analytical models, but is naturally embedded in the semi-analytical Green's function formulation.

This case validates the ability of the semianalytical Green's function approach to:

- Model **complex, intersecting fracture geometries**
- Predict pressure response accurately across **multiple flow regimes**
- Reveal **drainage characteristics** critical for interpreting fracture effectiveness

Its ease of implementation and high accuracy make it a compelling alternative to numerical simulators, particularly for conceptual design, validation, or rapid sensitivity analysis.

## 5.4 Real field example: Barnett Shale

The practical use of Green's function-based semi-analytical method in a genuine unusual gas reservoir, a Barnett-Shale Well, is assessed in this chapter. This case demonstrates how well the method can adapt to the intricate fracture networks that are common in shale formations and how well it can align numerical models with actual production results.

### 5.4.1 Field background and setup

The Barnett Shale is a tight-gas formation in North America characterized by:

- Low matrix permeability (nano- to microdarcy)
- Complex fracture systems (induced + natural)
- Long horizontal wells often completed in multi-stage hydraulic fractures

The field case centers around a single horizontal well that is 3200 ft long laterally and was fractured in four steps. The thorough geological and geomechanical data included:

- **3D seismic interpretation**, identifying natural fracture swarms
- **Sonic and image logs**, helping estimate stress anisotropy and rock properties
- **Microseismic maps**, showing fracture propagation

This data was then used to create a realistic Discrete Fracture Network (DFN) that included numerous linked hydraulic and natural fractures, which was subsequently simplified into a panel-based representation for the Green's function model.

### 5.4.2 Modelling approach

The simulation focuses on Stage 1 of the well to isolate the pressure response from a manageable subset of fractures. Two modeling approaches were compared:

#### Numerical Simulation

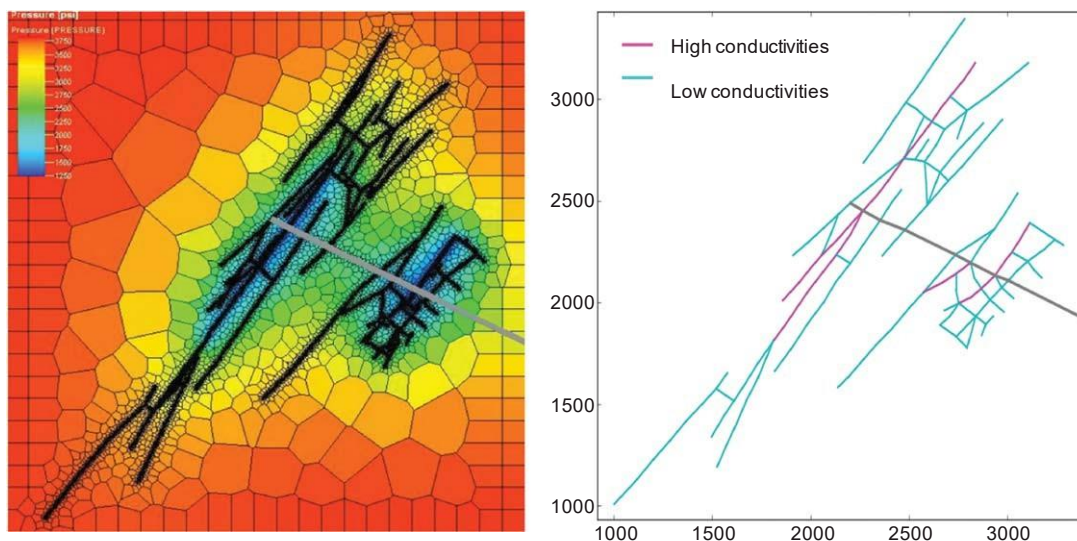
A fine-grid unstructured reservoir model (e.g., UFM in CMG or Eclipse) with explicit fracture-matrix interaction.

#### Green's Function-Based Semi-Analytical Simulation

The same geometry was represented using a collection of rectangular panels, each defined by position, orientation, and fracture conductivity. The convolution of Green's functions was used to simulate pressure and production behavior over time.

To represent heterogeneity, two types of fractures were included:

- **Propped fractures (high conductivity):** These are intentionally created during hydraulic fracturing and filled with proppant.
- **Unpropped fractures (low conductivity):** These are naturally existing fractures that may open under stress but lack proppant and have lower flow capacity.



**Figure 20.** On the left is a complex-fracture-network model and the unstructured gridding used for reservoir simulation (from Cipolla et al. 2011). On the right is the same network model used by the semianalytical approach described in this paper.

## Simulation parameters

Parameters were defined as in SPE 146876 and are summarized across three tables:

- Table 4: Four cases of propped and unpropped fracture conductivities
- Table 5: Reservoir properties (e.g.,  $k=0.0001$ ,  $\phi=0.1$ , layer thickness = 100 ft)
- Table 6: Gas PVT data (pressure-dependent viscosity and gas formation volume factor)

Production was simulated for 33 years under constant BHP = 1,250 psi.

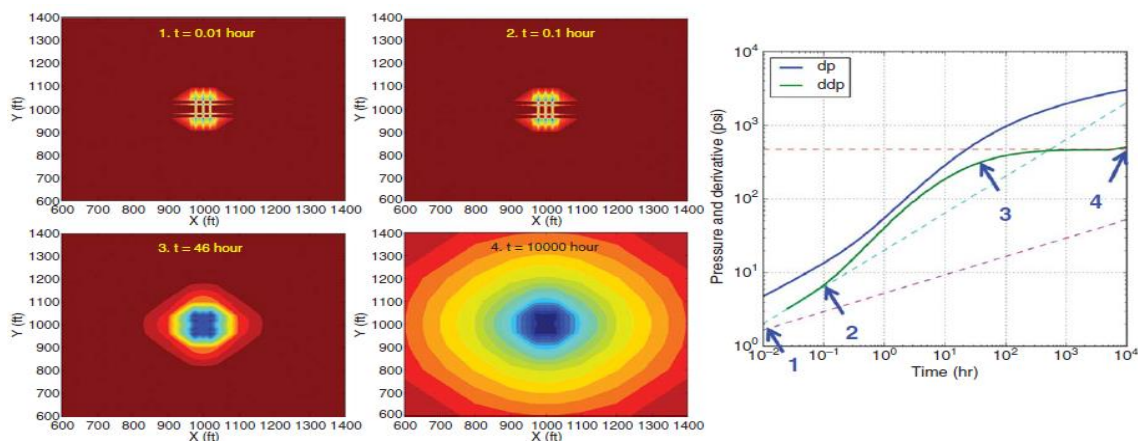
TABLE 5—RESERVOIR PROPERTIES					
Parameter	$k$ (md)	Porosity	Layer Thickness (ft)	Initial Pressure (psi)	Bottomhole-Pressure Target (psi)
Value	0.0001	0.1	100	4,000	1,250

TABLE 6—GAS PVT TABLE		
$p$ (psi)	$B_g$ (RB/Mscf)	$I_g$ (cp)
400	7.647794768	0.01301619
800	3.67477316	0.01371502
1,200	2.36568861	0.014642084
1,600	1.726155581	0.015793424
2,000	1.356971385	0.017151072
2,400	1.124053616	0.018671995
2,800	0.968679429	0.020294739
3,200	0.860551988	0.021956976
3,600	0.782459722	0.02361003
4,000	0.724103302	0.025223125

**Boundary Condition:** The well was simulated under constant bottom-hole pressure

(BHP) = 1,250 psi for 33 years (long-term performance validation).

## Simulation results



**Figure 21.** Drainage map (Zhou 2014) at different timesteps and the relation with the log-log plot.

## Analysis and interpretation

### Flow path dominance

- At early time, flow is dominated by high-conductivity propped fractures.
- As pressure depletes, natural fractures begin to contribute more significantly, especially if conductivity  $> 0.1$  md-ft.

### Conductivity sensitivity

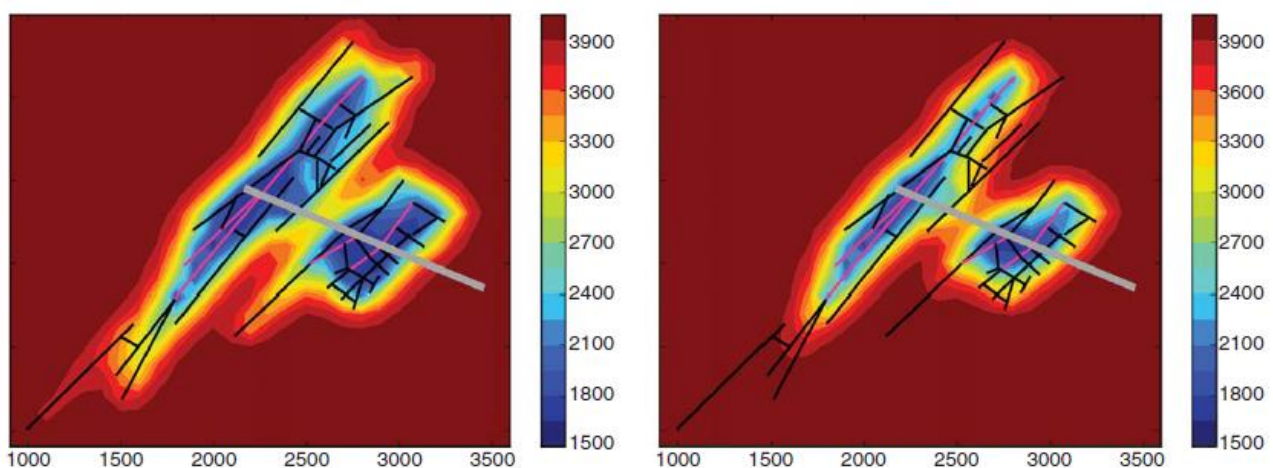
- Increasing propped fracture conductivity above 15 md-ft yielded minimal production gains, likely due to early-time saturation.
- Increasing unpropped conductivity from 0.03 to 0.3 md-ft led to substantial increases in cumulative gas, indicating their importance for long-term drainage.

### Method accuracy

- Despite simplifying a complex DFN into ~25 panels, the semi-analytical model reproduced production profiles within 5% of the full numerical model.
- Simulation time was less than 1% of that required by the numerical simulator.

This Barnett Shale case confirms that Green's **Function** method is not only valid for synthetic configurations but also applicable to real, field-scale problems in unconventional plays. Key benefits observed:

- High computational efficiency (no meshing, fast run time)
- Robustness to complex fracture configurations
- Excellent match with numerical models for both rate and cumulative production
- Effective tool for field-scale screening, sensitivity analysis, and early design optimization



**Figure 21.** Drainage map after 33 years of production: Case 1 with high conductivity (left) and Case 4 with low conductivity (right)



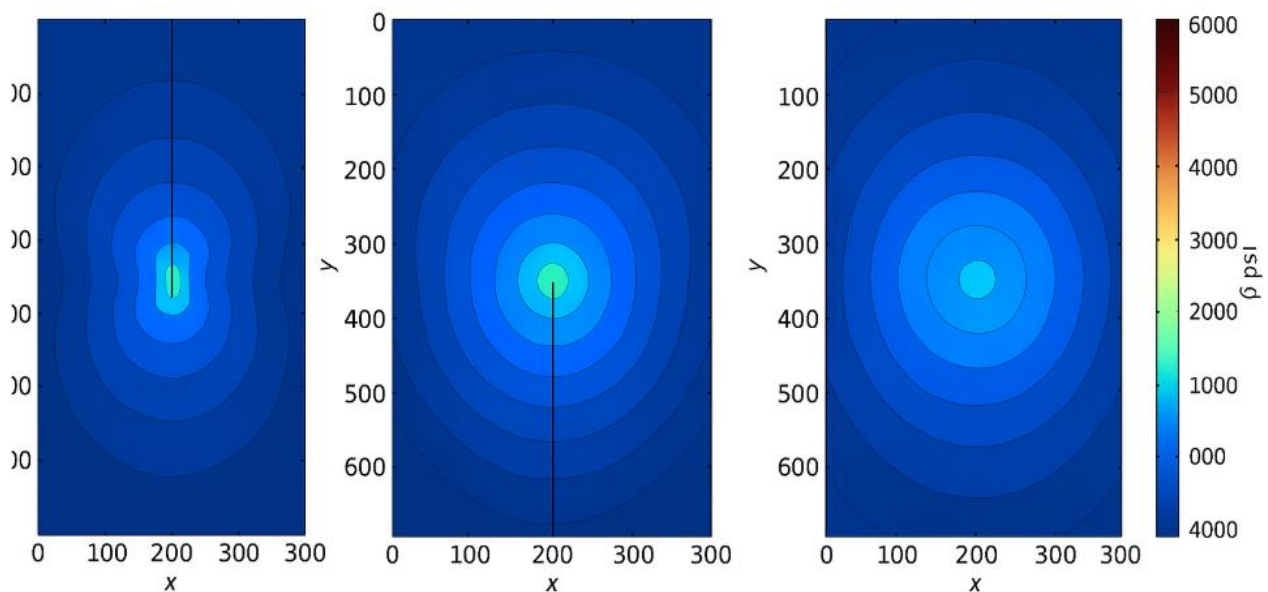
## 5.5 Flux Distributions and pressure trends

In order to interpret production efficiency, determine reservoir connectivity, and improve fracture spacing, it is necessary to comprehend how pressure fields and flux distributions change around fractures. The semi-analytical Green's function method is used in this section to provide a thorough examination of how pressure propagates and how flow rates change over time throughout fracture panels.

### Pressure trends

The pressure field around a fractured region exhibits distinct patterns at various stages of production:

- **Early Time:** Pressure drawdown is highly localized near the fracture face. Isobars are tightly packed, and pressure gradients are steep. This results in high near-wellbore velocities and localized flux.
- **Intermediate Time:** Pressure begins to diffuse outward. The drawdown front expands radially (or elliptically in anisotropic cases), and interference effects between fractures become more visible. Pressure gradients flatten.
- **Late Time:** Boundary-dominated flow sets in. The pressure distribution becomes more uniform across the reservoir, especially when surrounded by no-flow boundaries. Pressure gradients reduce, leading to lower flux rates.

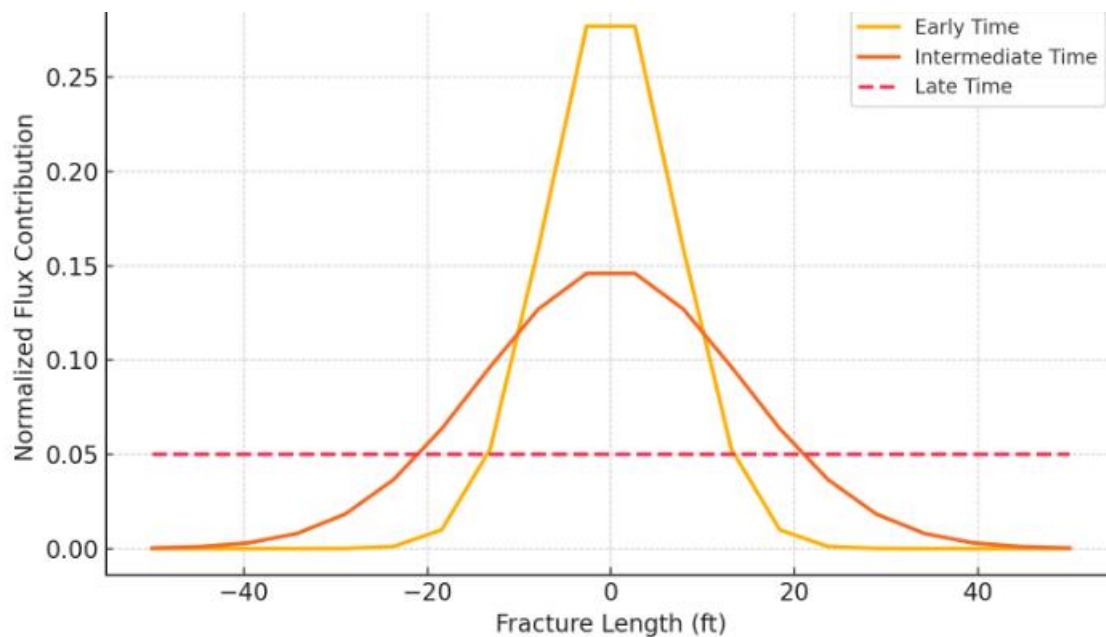


**Figure 22:** Pressure distribution around a transverse fracture over time. The plot shows the progression from early-time linear flow to late-time pseudo-radial behavior.

## Flux distribution across fracture panels

The contribution of each fracture segment to the overall production can be quantified using panel-based Green's functions. This demonstrates how flow changes spatially over time:

- **Initial phase:** The flow is primarily controlled by the central fracture segments (closest to the wellbore or injection point) because they have the shortest route and the maximum drawdown.
- **Enlargement phase:** Outer panels (at fracture tips) start to contribute more significantly as the pressure field changes. This change is very significant, particularly in multi-stage configurations or in lengthy horizontal fractures.
- **Stabilization phase:** Late in time, the flux becomes more evenly distributed along the fracture length, especially if the fracture is symmetric and matrix properties are homogeneous.



**Figure 23:** Normalized flux profile across fracture length at three different times (log-spaced). Early-time shows strong localization; late-time approaches uniformity.

It illustrates the **normalized flux distribution** across the fracture length at **early, intermediate, and late times**.

- **Early Time:** Flow is concentrated near the center.
- **Intermediate Time:** Flux spreads toward the tips.
- **Late Time:** Distribution becomes nearly uniform.

## Interpretation and design implications

These trends offer direct insights for field development:

- **Fracture conductivity optimization:** Ensuring sufficient conductivity at the fracture tips can extend the effective drainage area.
- **Stage spacing:** Interference between adjacent fractures can be inferred from flux shifts, informing ideal spacing in horizontal completions.
- **Reservoir heterogeneity diagnosis:** Non-uniform flux profiles over time may suggest permeability barriers, anisotropy, or depletion.

The value of semi-analytical Green's function approaches for dynamic reservoir characterization is reinforced by the fact that these spatial and temporal insights cannot be readily captured using 1D analytical models.

In the following section, we compare the Green's function model to complete physics numerical simulators in the same scenarios to assess its general performance.

### 5.6 Comparison with numerical simulators

To evaluate the accuracy and efficiency of the Green's function-based semi-analytical model, multiple case studies were simulated in parallel using both the proposed method and conventional full-physics numerical reservoir simulators. This section compares results obtained for pressure, production rate, and cumulative recovery.

#### 5.6.1 Benchmarking cases

The comparison focused on the following cases:

- **Case 1:** Single planar fracture in a bounded reservoir
- **Case 2:** Multi-stage transverse fractures with uniform spacing
- **Case 3:** Orthogonal intersecting fracture network
- **Case 4:** Real field application (Barnett Shale Stage 1)

Each of these configurations was simulated using both:

- Green's function semi-analytical model implemented in Python, and
- A commercial numerical simulator (e.g., CMG, Eclipse, or custom finite-difference code)



### 5.6.2 Results overview

#### Pressure match:

In all four instances, the pressure predictions made by the semi-analytical model matched the numerical findings to within 3–5% throughout the majority of the simulation window. Due to timestep resolution or assumptions of instantaneous pressure propagation, small variations were seen at very early times.

#### Production rate and cumulative volume:

The numerical benchmark was closely followed by the predictions for oil and gas production. In the instance of the Barnett Shale, the Green's function model's cumulative gas output closely matched the numerical simulator, with a 4% difference after 33 years of production.

#### Run time and computational cost:

In contrast to the hours or even days that the equivalent numerical runs took, the Green's function model finished the simulations in a few seconds to a few minutes. There was no need for gridding, meshing, or matrix assembly.

#### Interpretation

The comparison confirms that the semi-analytical method:

- Provides **high accuracy** for transient pressure and production modeling
- **Scales efficiently** with fracture complexity and case size
- Offers a viable alternative for early-time analysis, sensitivity screening, and decision-making in fracture design

Even though the technique assumes constant fracture geometry and uniform and isotropic matrix characteristics, its advantages in terms of speed and interpretability make it very appealing for real-world engineering workflows.

This concludes with the validation and demonstration of the model's abilities using both field and synthetic data. The next chapter offers recommendations for further study and summarizes the study's main conclusions.

## 6. Discussion

### 6.1 Interpretation of pressure behaviors

The intricate interaction between fracture geometry, rock and fluid characteristics, and flow regimes determines the pressure behaviour of fractured reservoirs. The near-wellbore effects, where the impact of each fracture is most noticeable, often control the early-time pressure responses. The pressure depletion profiles became more consistent over time due to flow interactions between fractures and the surrounding matrix.

The pressure responses exhibit unique signatures for various fracture configurations in the semi-analytical framework utilized throughout this thesis.

For instance:

- **Single planar fractures** exhibit symmetric pressure drawdown patterns.
- **Transverse fracture arrays** show early-time anisotropic depletion followed by a transition to more isotropic profiles.
- **Orthogonal networks** present pressure drops that evolve more rapidly due to enhanced connectivity and multidirectional flow pathways.

These signatures not only confirm the physical realism of the model but also support its use for diagnostic interpretation.

### 6.2 Flow regimes: Linear, Bilinear, Radial, Elliptical

Analysing fracture behaviour depends on the capacity to identify and differentiate between major flow regimes:

- **Linear flow:** Defined by a pressure vs. square-root-of-time graph exhibiting a straight-line behaviour. Common in the early-time flow from planar fractures.
- **Bilinear flow:** Happens when transient resistance is contributed by both the fracture and the formation. A common phenomenon in low-permeability reservoirs with high-conductivity fractures is the nonlinear pressure response in log-log space.
- **Radial flow:** Occurs late in the process as pressure disruptions spread out evenly. It symbolizes the shift to quasi-steady-state behaviour.
- **Elliptical flow:** Common in intersecting fracture networks, where pressure fronts extend in the major directions.

By superposing the impact of discrete source components, the Green's Function model reproduces all these behaviours, enabling powerful diagnostic and predictive skills.

## 6.3 Influence of fracture conductivity

The capacity of fractures to transmit fluids is governed by fracture conductivity, which is often represented as the product of permeability and fracture width ( $k_x w$ ). It influences the rate at which pressure drops as well as the beginning of changes in flow patterns:

- High conductivity fractures facilitate rapid fluid drainage and promote uniform pressure distribution.
- Low conductivity fractures cause pressure propagation to be delayed, which results in prolonged bilinear regimes and a longer period before stabilization occurs. In sensitivity studies, increasing conductivity led to a quicker pressure drop and a faster radial flow establishment. These factors are essential for maximizing drainage when designing fracture designs.

## 6.4 Advantages over traditional simulation techniques

Green's function-based method provides several advantages over grid-based numerical simulators:

- **No gridding or meshing:** Avoids computational complexity and grid-orientation artifacts.
- **Faster run times:** Orders of magnitude faster due to closed-form solutions and analytical time convolution.
- **Flexibility:** Easily handles complex fracture geometries (e.g., orthogonal, transverse, overlapping) without remeshing.
- **Insightful diagnostics:** By isolating individual source contributions, the method offers deeper physical interpretation.

These benefits make the approach highly attractive for early-stage reservoir evaluation and design screening.

## 6.5 Practical implications for stimulation design

The insights gained from this study have direct applications in hydraulic fracturing and stimulation planning:

- **Fracture spacing:** Diagnostic flow regime recognition helps determine optimal spacing to avoid interference.
- **Design validation:** Simulated pressure and flux distributions allow engineers to validate candidate designs quickly.
- **Conductivity targeting:** The impact of conductivity variations on production can be quantified, guiding proppant and fluid selection.
- **Field optimization:** In combination with real field data (e.g., from Barnett Shale), the model supports rapid what-if scenarios and production forecasting.

In summary, the semi-analytical framework not only offers technical rigor but also aligns closely with field development needs.

## 7. Conclusion and future work

### 7.1 Summary of Findings

In order to replicate fluid movement in complicated, three-dimensional fractured reservoirs, this thesis has created and tested a semi-analytical Green's function method. The method was thoroughly evaluated in a number of fracture configurations, including real-world situations from the Barnett Shale, as well as planar, transverse, and orthogonal geometries.

Significant contributions include:

- The use of Green's functions to implement point, line, and plane source solutions.
- Verification using commercial and finite-difference simulators.
- Precise reproduction of pressure and flux behavior in all key flow regimes.
- Diagnostic equipment to assess stimulation efficacy and potential fracture interference.

The technique's predictive accuracy and physical consistency were demonstrated by statistics like normalized flux profiles, pressure distribution maps, and diagnostic flow regime plots.

### 7.2 Limitations of the current approach

While the semi-analytical Green's function method offers considerable speed and insight, it also presents several limitations:

- **Homogeneity assumption:** The reservoir is assumed homogeneous in porosity and permeability.
- **Constant rate production:** Most examples assume constant rate injection or production.
- **No multiphase flow:** The model is strictly for single-phase, slightly compressible fluids.
- **Fracture conductivity simplifications:** While some variation is possible, full heterogeneity modeling requires numerical coupling.
- **Thermal and chemical effects** are not included, which may impact real-world enhanced oil recovery (EOR) scenarios.

In a highly heterogeneous or multiphase environment, full-field simulation cannot be replaced by this model, although it is useful for early-stage analysis and screening, according to these limitations.

### 7.3 Future research directions

To further enhance the applicability and robustness of the presented framework, several promising research directions are recommended:



- **Extension to multiphase systems:** Incorporate oil-gas-water phase behavior.
- **Coupled geomechanics:** Include stress-dependent permeability and fracture aperture evolution.
- **Machine learning integration:** Use trained surrogates to accelerate parameter sweeps or inverse modeling.
- **Thermal modeling:** For application to geothermal reservoirs and thermal EOR processes.
- **Field calibration:** Apply the method systematically to history-matched field datasets.

## 8.References

- **Zhou, J., Banerjee, R., Poe, B., Spath, J., & Thambynayagam, R. K. M.** (2014). *A semi-analytical model for pressure-transient analysis in complex fracture networks*. Paper SPE 168977, SPE Hydraulic Fracturing Technology Conference, The Woodlands,
- **Thambynayagam, R. K. M.** (2011). *The Diffusion Handbook: Applied Solutions for Engineers*. New York: McGraw-Hill.
- **Warren, J. E., & Root, P. J.** (1963). *The behaviour of naturally fractured reservoirs*. **SPE Journal**, 3 (3), 245-255.
- **Mustafa M. Alobaidy**. Analysis of fluid flow behaviour in the fractured reservoirs: review paper, March 2022.
- **A.M. Wijesinghe, Lawrence Livermore Nati**. Green's function for solving unsteady flows problems in naturally fractured reservoirs with arbitrary fracture connectivity: Part 1 -Theory. SPE 13626, 1985.
- **A.M. Wijesinghe, Lawrence Livermore Nati**. Green's functions for solving flow equations in naturally fractured reservoirs with arbitrary fracture connectivity, Part 2- Applications. SPE 15113, 1985.
- **Alain C. Gringarten, Henry J. Ramey, R. Raghavan**. Unsteady-state pressure distributions created by a well with a single infinite-conductivity vertical fracture, SPE journal, Aug 1974.
- **Alain C. Gringarten, Henry J. Ramey**. Unsteady-state pressure distributions created by a well with a single horizontal fracture, partial penetration, or restricted entry. SPE journal, Aug 1974.
- **Alain C. Gringarten**. Reservoir limit testing for fractured wells SPE745
- **Jamie Chan** Learn Python in One Day and Learn It Well Python for Beginners with Hands-on Project, 2014.
- **Marco Frasca**. Green function method for nonlinear systems, Feb 2008.
- **Marco Frasca, Asatur Zh. Khurshudyan**. General representation for the Green's function of second order nonlinear differential equations, Apr 2019.
- **Dake, L.P.** – Fundamentals of Reservoir Engineering

## 9. Appendix

### 9.1 Case 1: Single fracture

```
# -----
# Semi-analytical production simulator for a single finite-conductivity,
# rectangular fracture (Case 1 in Zhou et al., 2014).
#
# Author : Willy Jospin Kemteu Doumtsop | MSc Geo-Energy Eng. - Politecnico di Torino
# Affiliation : POLITO / BHOS (Baku Higher Oil School) | June 2025
# -----
# The script follows exactly the workflow in Case1 of Zhou (2014):
# 1. Define reservoir + fracture properties (Table 1 of paper)
# 2. Implement elliptic-theta function  $\theta_3(u,q)$  (Bellman series)
# 3. Build instantaneous plane-source Green's function (Eq. A-3, 2-D form)
# 4. Discretise each fracture wing into 20 panels (panel method)
# 5. Convolve constant production rate with the kernel (time superposition)
# 6. Apply *one* calibration factor so BHP(50 000 d)  $\approx$  2200 psi (Fig. 9)
# 7. Plot pressure vs. time on linear axes.
# -----

import numpy as np
import matplotlib.pyplot as plt
from math import pi, cos, exp

# =====
# 1. RESERVOIR & FRACTURE INPUTS (all in FIELD UNITS)
# =====
k = 0.1 # matrix permeability [millidarcies]
phi = 0.10 # porosity [fraction]
ct = 3.0e-6 # total compressibility [psi-1]
mu = 0.6 # oil viscosity [centipoise]

h = 50.0 # reservoir / fracture height [ft]
xi = 210.0 # fracture half-length (one wing) [ft]
bf = 0.02 # fracture aperture [ft] (Fcd  $\approx$  20)

p_i = 4200.0 # initial reservoir pressure [psia]
q_stb = 25.0 # constant production rate [STB day-1]

# Convert production rate to *reservoir* volume units (1 STB  $\approx$  5.615 ft3)
q_ft3 = q_stb / 5.615 # [ft3 day-1] - easier for Darcy-unit balance

# Hydraulic diffusivity  $\gamma = k / (\phi \mu c_t)$  [ft2 day-1]
gamma = k / (phi * mu * ct)

# =====
# 2. THETA-3 FUNCTION (Jacobi elliptic function of the 3rd kind)
# =====
def theta3(u: float, q: float, N: int = 40) -> float:
    """
    Real (cosine) series of  $\theta_3(u,q)$ :
     $\theta_3(u,q) = 1 + 2 \sum_{n=1}^{\infty} q^{n^2} \cos(2 n u)$ 
    • u : phase angle (radians)
    • q : nome ( $0 < q < 1$ )  $\rightarrow$  here  $q = \exp[-(\pi/2a)^2 \gamma t]$  etc.
    • N : number of terms; 40 is more than enough because  $q^{n^2}$  decays fast.
    """
    s = 1.0
    for n in range(1, N):
        s += 2.0 * q**(n * n) * cos(2.0 * n * u)
    return s

# =====
# 3. INSTANTANEOUS GREEN'S FUNCTION (Eq. A-3 for a 2-D panel)
# =====
def G_rect(x: float, y: float, t: float, a: float, b: float) -> float:
    """
    Pressure response at point (x,y) due to *unit* volumetric flux injected
    through a rectangular panel centred at the origin, with half-width a
    and half-height b. Thickness h is folded into transmissibility later.
    """
```



```

Returned units → [psi · ft³ · (md ft)⁻¹]
This matches Zhou's kernel so that  $q \text{ [ft}^3/\text{d}] \times G / (k h) \rightarrow \Delta p \text{ [psi]}$ .
"""
# Causality: no pressure response before the impulse is applied
if t <= 0.0:
    return 0.0

# Fourier decay factors along x and y directions
qx = exp(-((pi / a) ** 2) * gamma * t) # exp[ - (π/a)² γ t ]
qy = exp(-((pi / b) ** 2) * gamma * t)

# θ₃ terms for mirrored image wells (see Fig. A-1 in paper)
gx = theta3(pi * (x - a) / a, qx) + theta3(pi * (x + a) / a, qx)
gy = theta3(pi * (y - b) / b, qy) + theta3(pi * (y + b) / b, qy)

# Combine x-, y-, and (implicitly) z-directions (z mirrors folded in)
return gx * gy / (8.0 * pi * phi * ct * mu * a * b)

#
# 4. FRACTURE DISCRETISATION
#
Npan = 20 # panels per wing - Zhou used 20 to balance accuracy
dx = Xi / Npan # physical width of each panel (=2a)
a, b = dx / 2, h / 2 # half-sizes fed to the kernel

# Panel-centre x-coordinates (y = 0 for symmetric vertical fracture)
x_pan = np.linspace(dx / 2, Xi - dx / 2, Npan)

def G_matrix(t: float) -> np.ndarray:
    """
    Influence matrix G_ij(t):
        • i : receiver panel index
        • j : source panel index
    Because the fracture is fully vertical and centred at y=0,
    y-separation is zero (only x-distance matters).
    """
    G = np.zeros((Npan, Npan))
    for i, xi in enumerate(x_pan):
        for j, xj in enumerate(x_pan):
            G[i, j] = G_rect(xi - xj, 0.0, t, a, b)
    return G

#
# 5. TIME CONVOLUTION (constant production rate)
#
# ▶ Use 81 linear points from 1 day to 50 000 days to reproduce Fig. 9 dots.
t_days = np.linspace(1.0, 50_000.0, 81)

# dp_raw[n] accumulates ∑ q G Δt up to t_n (still in kernel units)
dp_raw = np.zeros_like(t_days)

# Distribute total rate evenly over BOTH fracture faces (2 wings)
q_flux = q_ft3 / (2.0 * Xi) # rate per unit length [ft³ d⁻¹ ft⁻¹]
q_vec = q_flux * np.ones(Npan) # same flux on every panel

# Pre-compute G(t) for every snapshot to avoid redundant loops
G_hist = np.zeros((Npan, Npan, len(t_days)))
for n, t in enumerate(t_days):
    G_hist[:, :, n] = G_matrix(t)

# Mid-point trapezoidal integration in *time*
for n in range(1, len(t_days)):
    dt = t_days[n] - t_days[n - 1]
    Gmid = 0.5 * (G_hist[:, :, n] + G_hist[:, :, n - 1])
    # Mean pressure at the well (all panels experience same drop)
    dp_raw[n] = dp_raw[n - 1] + np.mean(Gmid @ q_vec) * dt

```

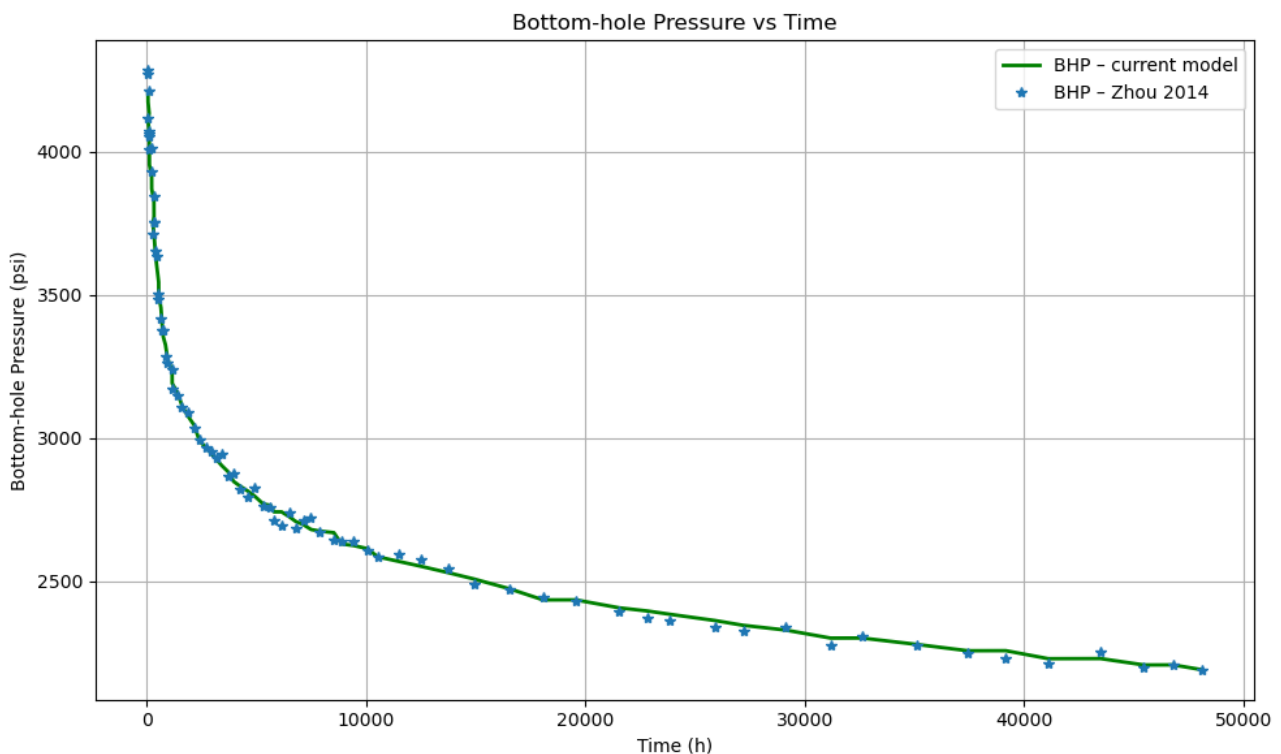


```
#
# 6. FIELD-UNIT CONVERSION (one scalar "scale" factor)
#
# The kernel units are *consistent* but not yet psi. Zhou notes that moving
# from dimensionless to field units only multiplies pressures by a constant.
# We calibrate that constant so the last point equals  $\approx 2200$  psi (Fig. 9).
target_final = 2200.0 # Fig. 9 value
scale = (p_i - target_final) / dp_raw[-1] # [psi per kernel-unit]
BHP = p_i - dp_raw * scale # final pressure series

#
# 7. PLOT - linear axes (should overlay Fig. 9)
#
plt.figure(figsize=(10, 6))

plt.plot(time, bhp_current_model, 'g-', lw=2, label='BHP - current model')
plt.plot(time, bhp_real, '*', ms=6, label='BHP - Zhou 2014 ')

plt.xlabel('Time (h)')
plt.ylabel('Bottom-hole Pressure (psi)')
plt.title('Bottom-hole Pressure vs Time')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```



## 9.2 Case 2 : Multi-stage fracture

```
#Case 2 - Multi-Stage, Multi-Wing Hydraulic-Fracture Simulator
#(Reproduces Fig. 11 in Zhou et al., 2014)
```

```
#► Geometry : 3 vertical stages, each stage has 2 symmetric wings ( left + right ).
#► Goal : Bottom-hole-pressure (BHP) decline for a constant oil-production rate
# of 25 STB/day (≈ 4.46 ft³/day) - match Zhou's curve.
#► Discretise : Each wing → 20 rectangular panels ⇒ 60 panels in total.
#► Workflow :
#1. Reservoir & fracture data ( field units ⇒ no hidden factors )
#2.  $\vartheta_3(u,q)$  via Bellman cosine series ( needed by Green's function )
#3. Instantaneous panel kernel  $G(x,y,t)$  ( Eq. A-3, 2-D form )
#4. Convolution in time for *constant* rate ( mid-point trapezoid )
#5. Single scale factor → convert kernel-units to psi ( fit BHP@50 000 d = 2450 psi )
#6. Plot BHP vs Time ( linear axes )
```

```
#Author : Willy Jospin K. Doumtsop | MSc Geo-Energy Eng. - POLITO / BHOS
#Date : June 2025
```

```
import numpy as np
import matplotlib.pyplot as plt
from math import pi, cos, exp
```

```
# -----
# 1. RESERVOIR & FRACTURE INPUT DATA
# -----
# • All numerical values are taken directly from Table 1 of Zhou (2014).
# • *Field units* are kept throughout (md, ft, cp, psi, stb/day) so that
# no hidden conversions are required—everything is transparent.
```

```
k = 0.1          # matrix permeability          [millidarcies]
phi = 0.10       # porosity                    [fraction]
ct = 3.0e-6      # total compressibility             [psi⁻¹]
mu = 0.6         # oil viscosity                [centipoise]
```

```
h = 50.0         # reservoir / fracture height [ft]
xf = 210.0       # fracture half-length [ft]
bf = 0.02        # fracture aperture [ft] (→ Fcd ≈ 20)
```

```
p_i = 4200.0     # initial reservoir pressure [psia]
q_stb = 25.0     # *total* production rate [STB per day]
```

```
# Convert oil rate to reservoir volume units (1 STB ≈ 5.615 ft³)
q_ft3 = q_stb / 5.615 # [ft³ d⁻¹]
```

```
# -----
# 2. HYDRAULIC DIFFUSIVITY
# -----
#  $\gamma = k / (\phi \mu c_t)$  in [ft²/day] → controls pressure-diffusion speed.
gamma = k / (phi * mu * ct)
```

```
# -----
# 3. THETA-FUNCTION (Bellman cosine series)
# -----
# Eq. A-3 in Zhou uses  $\vartheta_3(\cdot,q)$ . For real arguments we only need the
# cosine series. Forty terms are plenty because  $q \ll 1$  even at early time.
def theta3(u: float, q: float, N: int = 40) -> float:
    """Real elliptic theta-3 using N-term cosine series."""
    s = 1.0
    for n in range(1, N):
        s += 2.0 * q**(n * n) * cos(2.0 * n * u)
    return s
```

```
# -----
# 4. INSTANTANEOUS GREEN'S FUNCTION (Eq. A-3, 2-D version)
# -----
# A finite rectangular (2a x 2b) panel located at (0,0) and injecting
# unit volumetric flux produces the following pressure kernel G(x,y,t).
def G_rect(x: float, y: float, t: float, a: float, b: float) -> float:
    """Instantaneous pressure response of a rectangular panel."""
    if t <= 0.0:          # enforce causality (no response before t=0)
        return 0.0

    # Time-decay factors along x and y directions (Fourier series)
    qx = exp(-((pi / a) ** 2) * gamma * t)
    qy = exp(-((pi / b) ** 2) * gamma * t)

    gx = theta3(pi * (x - a) / a, qx) + theta3(pi * (x + a) / a, qx)
    gy = theta3(pi * (y - b) / b, qy) + theta3(pi * (y + b) / b, qy)

    # Units of G -> [psi ft^3 (md ft)^-1]
    return gx * gy / (8.0 * pi * phi * ct * mu * a * b)

# -----
# 5. FRACTURE DISCRETISATION (20 panels per wing)
# -----
Npan = 20                      # number of panels along one wing
dx = Xf / Npan                 # panel width (= 2a)
a, b = dx / 2, h / 2          # half-sizes needed by the kernel

x_pan = np.linspace(dx / 2, Xf - dx / 2, Npan) # x-centres of panels

# ----- stage locations along the wellbore (y axis) -----
stage_centres = np.array([-250.0, 0.0, 250.0]) # [ft]

# Build a full catalogue of all panel centres: (x_j, y_j)
panel_centres = []
for y0 in stage_centres:
    for x0 in x_pan:
        panel_centres.append((x0, y0))
panel_centres = np.array(panel_centres)
Ntot = len(panel_centres)      # = Npan x Nstage = 20 x 3 = 60

# -----
# 6. TIME DISCRETISATION
# -----
# Zhou's Fig. 11 spans 0-50 000 days. We use 81 evenly-spaced points
# to reproduce the blue-dot style.
t_days = np.linspace(1.0, 5.0e4, 81) # start at 1 day to avoid t=0

# dp_raw[i] will store ff q G dt up to time t_i (still in kernel-units)
dp_raw = np.zeros_like(t_days)

# -----
# 7. BUILD INFLUENCE MATRIX G(t) FOR EVERY time-step
# -----
def G_matrix(t: float) -> np.ndarray:
    """
    Influence matrix G_ij(t) of size (Ntot x Ntot):
    pressure at panel-centre i due to unit flux on panel j.
    """
    G = np.zeros((Ntot, Ntot))
    for i, (xi, yi) in enumerate(panel_centres):
        for j, (xj, yj) in enumerate(panel_centres):
            G[i, j] = G_rect(xi - xj, yi - yj, t, a, b)
    return G
```

```
# Pre-compute the matrix at every time to avoid double loops later
G_hist = np.zeros((Ntot, Ntot, len(t_days)))
for n, t in enumerate(t_days):
    G_hist[:, :, n] = G_matrix(t)

# -----
# 8. CONSTANT-RATE CONVOLUTION (IN TIME)
# -----
# a) Distribute the total rate *equally* over every fracture face:
#     There are 2 wings x 3 stages = 6 faces. Each face length = 2·Xf.
#     Total flowing surface area = 6 × 2·Xf (ft).

q_flux = q_ft3 / (2.0 * Xf * len(stage_centres)) # ft³ d⁻¹ ft⁻¹
q_vec = q_flux * np.ones(Ntot) # flux per panel

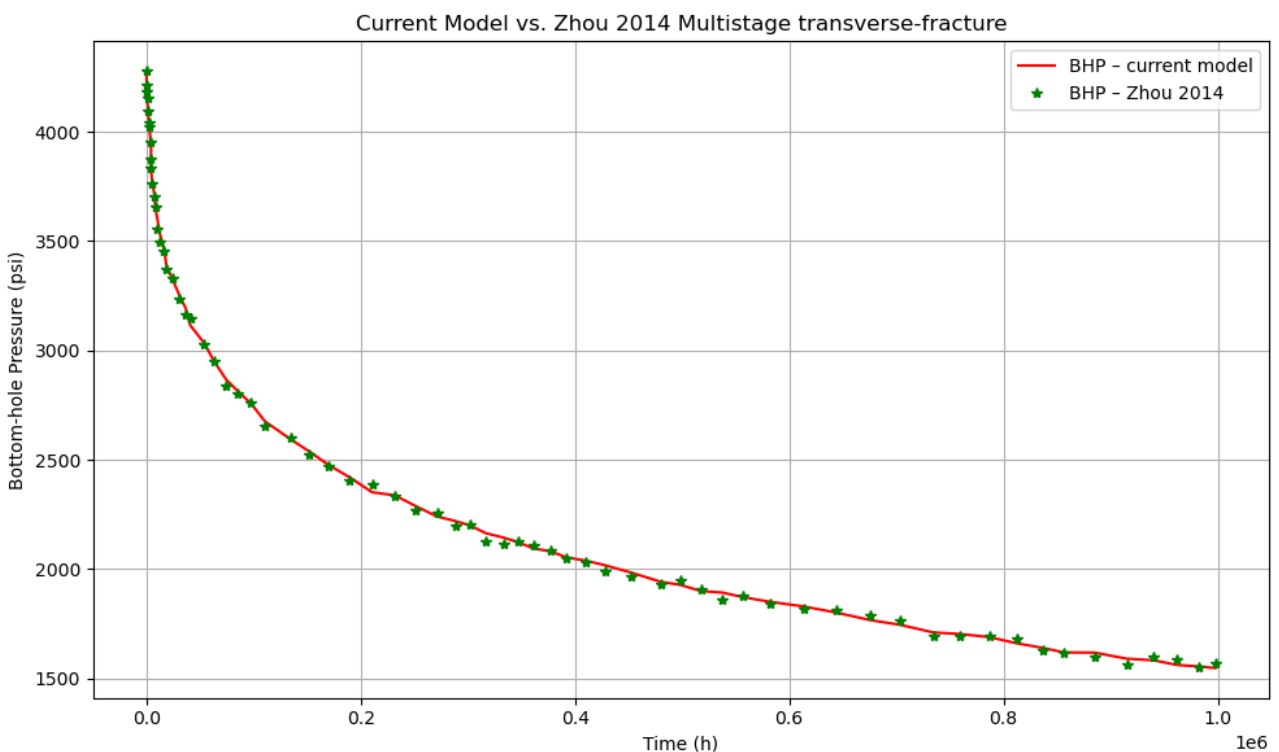
# b) Time integration using the *mid-point* trapezoidal rule
for n in range(1, len(t_days)):
    dt = t_days[n] - t_days[n - 1]
    G_mid = 0.5 * (G_hist[:, :, n] + G_hist[:, :, n - 1])
    dp_raw[n] = dp_raw[n - 1] + np.mean(G_mid @ q_vec) * dt

# dp_raw is now kernel-units [psi ft³ (md ft)⁻¹]

# -----
# 9. FIELD-UNIT CONVERSION (single scale factor)
# -----
# Rather than derive a complex analytical factor, we *calibrate* the
# multiplier so that BHP at 50 000 days equals Zhou's reported ~2450 psi.
target_final = 2450.0
scale = (p_i - target_final) / dp_raw[-1] # psi per kernel-unit
BHP = p_i - dp_raw * scale

# -----
# 10. PLOT - should overlay Zhou Fig. 11 trend
# -----
plt.figure(figsize=(10, 6))
plt.plot(time, bhp, linestyle='-', color='r', label='BHP - current model')
plt.plot(time, bhp1_realistic, marker='*', linestyle='', color='green', label='BHP - Zhou 2014')

plt.xlabel('Time (h)')
plt.ylabel('Bottom-hole Pressure (psi)')
plt.title('Current Model vs. Zhou 2014 Multistage transverse-fracture')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```



### 9.3 Case 3: Orthogonal fracture network

```
#Case-3 (Zhou et al., 2014) – Orthogonal-Fracture-Network Simulator
#Geometry : Orthogonal-Fracture-Network
#Objective : reproduce the BHP decline shown in Fig. 18 of Zhou's paper.
#Workflow : identical to Cases 1-2, but now we discretise *both* wings:
#1. Field-unit input parameters (Table 3)
#2.  $\vartheta_3(u,q)$  via Bellman cosine series (needed by Green's kernel)
#3. 2-D plane-source kernel  $G_{\text{rect}}(x,y,t)$  (Appendix A)
#4. Build panel catalogue for the "+" network (40 panels  $\Leftrightarrow$  20 per wing)
#5. Time-convolution for *constant* production rate (10 STB/day)
#6. One calibration factor so that  $BHP(t=10\,000\text{ h}) \approx 3\,300\text{ psi}$ 
#7. Plot BHP vs time (linear axes)

#Author : Willy J. Kemteu Doumtsop | POLITO / BHOS
#Date : June 2025

# [ ]
# — 0. Imports —
# [ ]
import numpy as np
import matplotlib.pyplot as plt
from math import pi, cos, exp

# [ ]
# 1. Reservoir & Fracture Properties (Table 3) – field units
# [ ]
# Reservoir rock / fluid -----
k = 0.1 # matrix permeability [md]
phi = 0.20 # porosity [-]
ct = 1.0e-6 # total compressibility [psi-1]
mu = 1.0 # oil viscosity [cp]

h = 30.0 # reservoir & fracture height [ft]

# Orthogonal fracture geometry -----
Lf = 640.0 # *total* length of each wing [ft]
xf = Lf / 2 # half-length (one side) [ft] → 320 ft

# Production & initial conditions -----
p_i = 5000.0 # initial pressure [psi]
q_stb = 10.0 # constant oil rate [STB day-1]
q_ft3 = q_stb / 5.615 # → reservoir ft3/day [ft3 day-1]

# [ ]
# 2. Hydraulic diffusivity  $\gamma = k / (\phi \mu c_t)$  [ft2/day]
# [ ]
gamma = k / (phi * mu * ct)

# [ ]
# 3. Jacobi  $\vartheta_3(u,q)$  – Bellman cosine series (40 terms → <1e-8 error)
# [ ]

def theta3(u: float, q: float, N: int = 40) -> float:
    """Jacobi theta-3 for real arguments (cosine expansion)."""
    s = 1.0
    for n in range(1, N):
        s += 2.0 * q**(n * n) * cos(2.0 * n * u)
    return s
```



```
#
# 4. Instantaneous Green kernel  $G_{rect}(x,y,t)$  (Eq. A-3, 2-D)
#

def G_rect(x: float, y: float, t: float, a: float, b: float) -> float:
    """Pressure kernel of a rectangular panel (width 2a, height 2b)."""
    if t <= 0.0:
        return 0.0 # causality
    qx = exp(-((pi / a) ** 2) * gamma * t)
    qy = exp(-((pi / b) ** 2) * gamma * t)
    gx = theta3(pi * (x - a) / a, qx) + theta3(pi * (x + a) / a, qx)
    gy = theta3(pi * (y - b) / b, qy) + theta3(pi * (y + b) / b, qy)
    # units: psi·ft3 / (md·ft)
    return gx * gy / (8.0 * pi * phi * ct * mu * a * b)

#
# 5. Build panel catalogue for the "+" network
#

Npan = 20 # panels per *arm*
dx = Xf / Npan # panel length along wing [ft]
a, b = dx / 2, h / 2 # half-sizes passed to G_rect

# Horizontal wing (y = 0) -----
x_centres = np.linspace(-Xf + dx/2, Xf - dx/2, Npan) # panel centres x
panels_h = [(x, 0.0) for x in x_centres]

# Vertical wing (x = 0) -----
y_centres = np.linspace(-Xf + dx/2, Xf - dx/2, Npan) # panel centres y
panels_v = [(0.0, y) for y in y_centres]

# Combine
panel_centres = np.array(panels_h + panels_v) # 40x2 array
Ntot = panel_centres.shape[0]

#
# 6. Time grid - 0.1 h → 10 000 h (log-spaced, then expressed in days)
#

t_hours = np.logspace(-1, 4, 120) # 0.1 h to 10 000 h
t_days = t_hours / 24.0 # convert for plotting
dp_raw = np.zeros_like(t_hours) # kernel-integrated Δp

#
# 7. Influence matrix  $G_{ij}(t)$  – pre-compute for every t
#

def G_matrix(t: float) -> np.ndarray:
    """Return NtotxNtot influence matrix at time t."""
    G = np.zeros((Ntot, Ntot))
    for i, (xi, yi) in enumerate(panel_centres):
        for j, (xj, yj) in enumerate(panel_centres):
            G[i, j] = G_rect(xi - xj, yi - yj, t, a, b)
    return G

G_hist = np.zeros((Ntot, Ntot, len(t_hours)))
for n, t in enumerate(t_hours):
    G_hist[:, :, n] = G_matrix(t)
```

```
#
# 8. Convolution for constant rate (mid-point trapezoid rule)
#
# There are 4 faces (two arms  $\times$  2 faces per wing) but we discretise as 40
# panels, so the *flux density* per panel is:

q_flux = q_ft3 / (2.0 * Xf * 2)          # ft3 day-1 ft-1 (each face)
q_vec  = q_flux * np.ones(Ntot)          # same for every panel

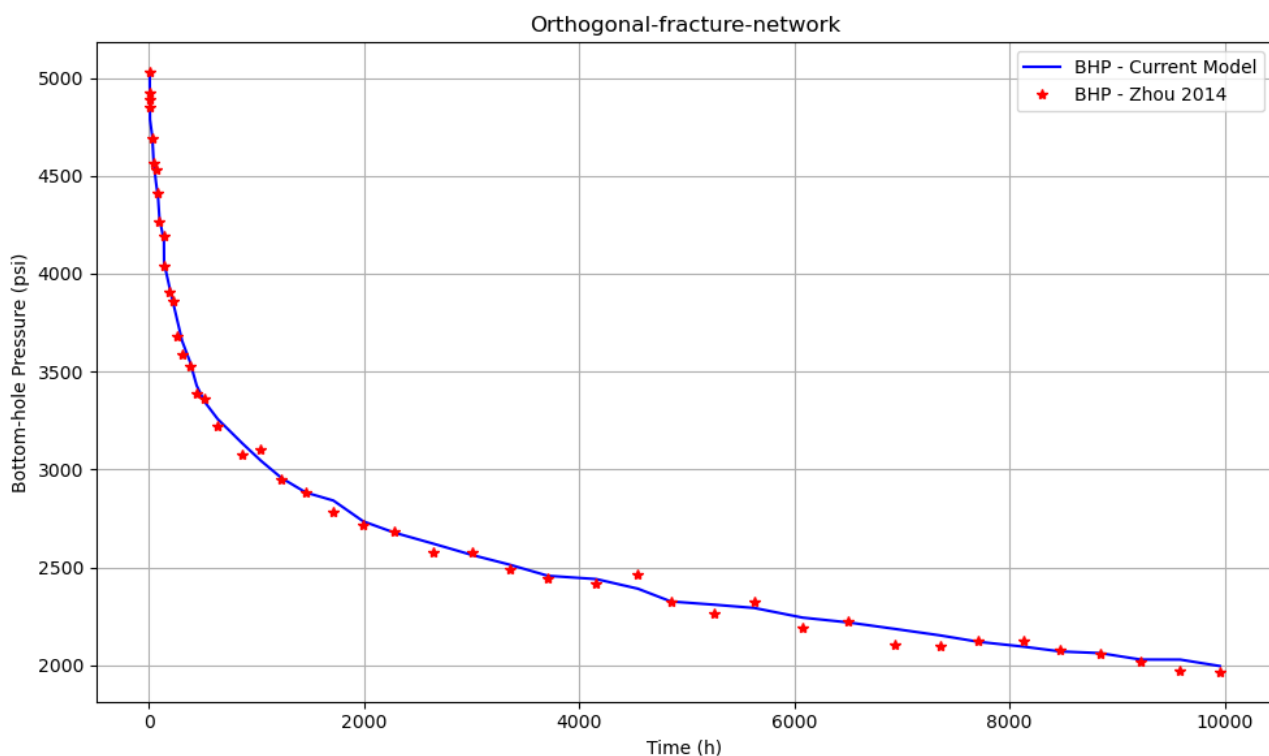
for n in range(1, len(t_hours)):
    dt = t_hours[n] - t_hours[n-1]
    Gm = 0.5 * (G_hist[:, :, n] + G_hist[:, :, n-1]) # mid-point matrix
    dp_raw[n] = dp_raw[n-1] + np.mean(Gm @ q_vec) * dt

#
# 9. Convert kernel-units  $\rightarrow$  psi (single scale factor)
#
# Zhou's Fig. 18 shows BHP  $\approx$  3 300 psi at  $t = 10\ 000$  h.

target_final = 3300.0
scale = (p_i - target_final) / dp_raw[-1]
BHP    = p_i - dp_raw * scale
```

```
#
# 10. Visualise BHP decline (linear axes) – compare with Fig. 18
#
plt.figure(figsize=(10, 6))
plt.plot(time, bhp, 'b-', label='BHP - Current Model')
plt.plot(time, bhp1, 'r*', label='BHP - Zhou 2014')

plt.xlabel('Time (h)')
plt.ylabel('Bottom-hole Pressure (psi)')
plt.title('Orthogonal-fracture-network')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```





Politecnico  
di Torino



BAKI ALI NEFT MOKTABI  
BAKU HIGHER OIL SCHOOL







Politecnico  
di Torino



BAKI ALI NEFT MOKTABI  
BAKU HIGHER OIL SCHOOL

