A DYNAMIC REPORTING GRID METHOD FOR INVESTIGATING SLIGHTLY COMPRESSIBLE FLUID FLOW IN LOW-PERMEABILITY MEDIA

by

Susan Mary Schrader

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Mathematics

MONTANA STATE UNIVERSITY
Bozeman, Montana

November 2025

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

NOMENCLATURE

Parameter	SI Units	Customary Units	Name
k	m^2	md (millidarcy)	Permeability
v	m/s	ft/sec	Flow velocity
μ	$Pa-s$	cP (centipoise)	Viscosity
p	Pa	psi	Pressure
q	m^3/s	bbl/day	Volumetric flow rate
A	m^2	ft^2	Cross sectional area to flow
ρ	kg/m^3	lbf/ft^3	Fluid density
φ	unitless		Porosity
c_o	$1/Pa$	$1/psi$	Oil compressibility
c_φ	$1/Pa$	$1/psi$	Formation compressibility
c_t	$1/Pa$	$1/psi$	Total compressibility
G	Pa/m	psi/ft	Threshold pressure gradient
$s(t)$	m	ft	Moving boundary parameter

Table 0.1: Parameter definitions and units

1

INTRODUCTION

Research Statement

Darcy's Law is a foundational equation for modeling fluid flow through porous rock and other media. It is a constitutive law that relates fluid velocity to the pressure gradient, scaled by the ratio of permeability (the rock's ability to transmit fluids) to viscosity (the fluid's resistance to flow). As a stand-alone equation, Darcy's Law is sufficient to model steady-state flow of incompressible fluids, but it can be extended to account for compressible fluids and time-dependent flow. This extension to a transient flow model is fundamental to both analytical and numerical approaches in fields such as reservoir engineering.

When rock permeability is low, as in unconventional reservoirs, laboratory studies indicate that Darcy's Law does not accurately predict the observed flow velocity [10, 27, 38], and nonlinear behavior is observed. Modified versions of Darcy's Law have been proposed for low-permeability samples, including formulations that introduce a threshold pressure gradient [14, 15, 38].

This dissertation develops transient flow equations for low-permeability media by replacing Darcy's Law with a modified version that incorporates a threshold pressure gradient. The equations are solved numerically using a dynamic reporting grid method, which numerically replicates threshold pressure gradient behavior. This method provides an adaptable framework for modeling threshold-driven flow and enables investigation of scaling effects and nonlinear terms.

Flow in Porous Media

Recent developments in new drilling and production techniques have led to a growing interest in modeling efforts that are aimed specifically for flow through low-permeability porous media. As shown in Figure 1.1, oil and gas reservoirs can be classified as conventional or tight. In this context, *tight* means low permeability, and, in contrast to the conventional ones, tight reservoirs are classified as a type of unconventional reservoir. The category of unconventional reservoirs also includes reservoirs with high viscosity. Figure 1.2 presents a classification based on the pairing of both the viscosity and permeability within the medium.

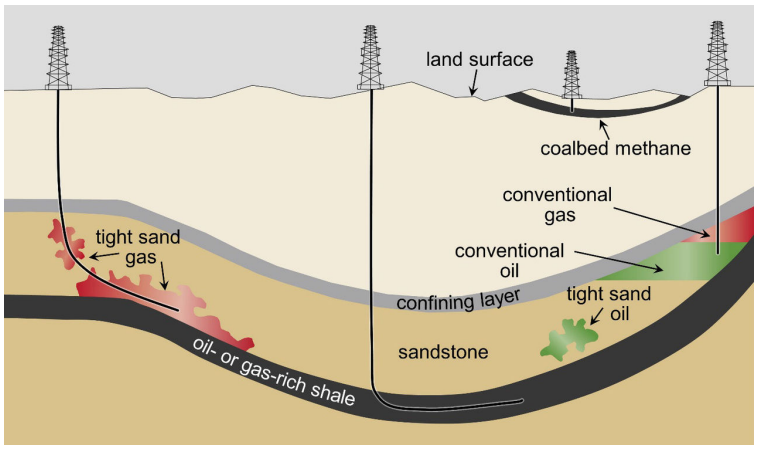

Figure 1.1: General oil and gas reservoir types including low permeability reservoirs (tight sand where oil and gas have migrated from source rock, and production directly from oil-or-gas rich source rock. [9].

Fundamentally, permeability is measured as an area, with SI units of square meters. The common measure for permeability is a darcy (or millidarcy), named after the engineer Henry Darcy. To give some perspective to the unit, dry beach sand has a permeability of around 1000 millidarcy [24]. In figure 1.2, both permeability and viscosity are used to classify unconventional reservoirs.

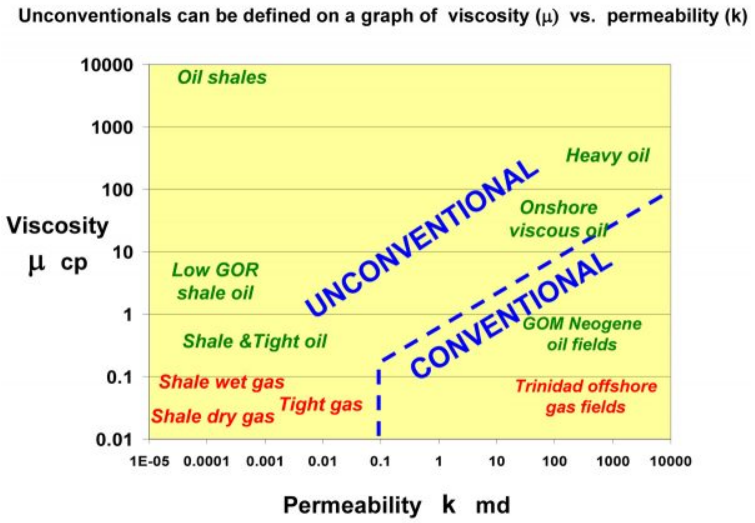

Figure 1.2: Graph exhibiting the parameter regimes that distinguish the unconventional reservoirs from those that are conventional . GOM here refers to Gulf of Mexico, and GOR is gas/oil ratio, with low GOR indicating an oil that carries little dissolved gas. [4].

The definition of low permeability can vary, as it can be defined from an operational perspective or based on observed flow behavior.

Wang et al. [45] observed that samples with permeability below approximately 50 md exhibit threshold pressure gradient behavior in laboratory experiments. This provides a range to define low permeability based on the flow behavior observed in the laboratory.

Other classifications are based on the implications for production methods. These include unconventional/tight oil(< 20 md) and oil shale (nanodarcy scale - no flow) [43, 46].

Darcy's Law

History of Darcy's Law

In 1856, Henri Darcy proposed an equation for the flow rate of water when passed through a sand filter. He developed this equation working with a column apparatus with which he measured the discharge (L^3/t) of the fluid exiting the sand pack. The relationship

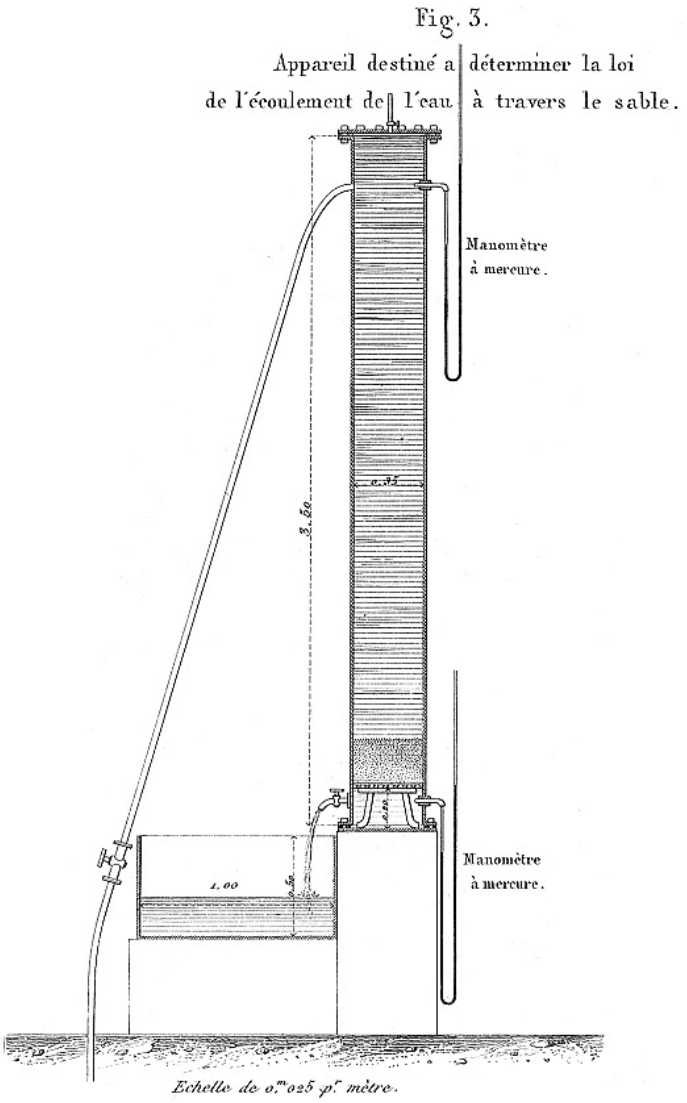

Figure 1.3: Column Apparatus used by Darcy and Bezin in 1856 [3].

observed was expressed as

$$Q = \frac{AK\left((h_1 + z_1) - (h_2 + z_2)\right)}{L} \tag{1.1}$$

Here Q is the volumetric discharge of the fluid, K is the hydraulic conductivity (a function of permeability, density, and viscosity), z is the elevation, h is the pressure head, and L is the length of the flow path. The subscripts 1 and 2 are at the upper and lower manometer locations shown in Figure 1.3.

Assumptions for Using Darcy's Law

Darcy's law as typically stated requires several assumptions in order to be sufficient to model fluid flow. These include two key conditions, one on the fluid and one on the pressure gradient. The fluid is required to be incompressible, having a nearly constant density despite varying pressures, and the flow is required to be steady-state in that the pressure in the model does not change with respect to time. Both of these conditions are reasonably met in many situations, such as when doing permeability measurements in a laboratory using a liquid. The flow becomes steady-state once the mass flow rate entering is equal to the mass flow rate exiting the system, and liquids at surface conditions can be modeled as incompressible without loss of accuracy. Additional requirements are that the flow is laminar and that the permeability and viscosity are constant with respect to position and pressure.

Statements of Darcy's Law

For a rectangular model, two versions of Darcy's Law for 1D flow in a Cartesian coordinate system are given in equations (1.2) and (1.3)

$$v = -\frac{k}{\mu}\frac{dp}{dx} \tag{1.2}$$

$$q = -\frac{kA}{\mu}\frac{dp}{dx}, \tag{1.3}$$

where

v is the velocity of the fluid in m/s (q/A where A is the area perpendicular to the flow and q is the volumetric flow rate)

k is the permeability of the porous medium in m^2

μ is viscosity of the fluid in $Pa \cdot s$

p is the pressure in Pa.

7

A CONVENTIONAL TRANSIENT FLOW MODEL

Incompressibility and steady-state flow, two of the assumptions required to apply Darcy's Law, can be met in some applications, such as the measurement of permeability in the laboratory setting using liquids. For the subsurface flow of hydrocarbons, Darcy's Law must be extended to form time-dependent flow models for compressible fluids. A process to achieve this is to combine Darcy's Law, the Conservation of Mass Principle, and compressibility models for the fluid and the rock. The resulting partial differential equation has analytical solutions, including those developed by Van Everdingen and Hurst [44] and Katz [16]. The analytic solutions have many applications and are used here to verify the numerical solutions.

Building the Model for Slightly Compressible Fluids

The first step of the process is to combine conservation of mass in (2.1) for 1D Cartesian flow, with Darcy's Law (1.2) to allow for time dependence.

$$-\frac{\partial}{\partial x}(v\rho) = \frac{\partial}{\partial t}(\varphi\rho) \tag{2.1}$$

In (2.1), the porosity (φ) is the ratio of the pore or void space volume of a porous solid to the total volume of that solid. Using the 1D Cartesian form of both equations and substituting the velocity from (1.2) into the velocity term in (2.1), we have (2.3).

$$-\frac{\partial}{\partial x}\left(-\rho\frac{k}{\mu}\frac{\partial p}{\partial x}\right) = \frac{\partial}{\partial t}(\varphi\rho) \tag{2.2}$$

$$\frac{\partial}{\partial x}\left(\rho\frac{\partial p}{\partial x}\right) = \frac{\mu}{k}\frac{\partial}{\partial t}(\varphi\rho) \tag{2.3}$$

There are two important assumptions to note in this case; both the viscosity of the fluid and the permeability of the media are treated as constants. In reality, the viscosity of fluid is a function of pressure, but for liquids, a constant viscosity is a reasonable assumption. Note that for gas flow, viscosity is a strong function of pressure and cannot be assumed constant. The permeability is unlikely to be a constant; however, we can treat permeability as an average value for a large region, or as a constant in a small homogeneous sample.

The next step is to express the density of the fluid in terms of the pressure. There are three categories to describe fluids in this context, based on a material property called the isothermal compressibility, c, generally defined below.

$$c = \frac{1}{\rho}\frac{d\rho}{dp} \tag{2.4}$$

Incompressible fluids fall into the first category and are assumed to have a constant density with respect to pressure and thus a compressibility of zero. This category describes liquids at standard conditions, and it can also be used to describe water at higher pressures when it does not contain a significant amount of dissolved gas. The second category is slightly compressible fluids, assumed to have a small and constant compressibility. This category describes liquids such as oil, which contain a significant amount of dissolved gases at high pressures. The third category is compressible fluids, and it is appropriate for gases. In this case compressibility is a function of both position and pressure.

For slightly compressible fluids (oil in this context), we assume a constant and small compressibility (c_o). Separating variables and integrating (2.4) from the pressure at standard conditions (p_{sc}) to the current pressure, and the density at standard conditions to the current density (ρ_{sc}), gives a relationship between density and pressure

$$\int_{p_{sc}}^{p} c_o dp \;\; = \int_{\rho_{sc}}^{\rho} \frac{d\rho}{\rho}$$

that when assuming a constant compressibility reduces to:

$$\rho = \rho_{\text{sc}} e^{c_o(p-p_{sc})} \tag{2.5}$$

Combining this model with equation (2.3) leads to an equation for the transient flow of slightly compressible fluids in porous media. In the first step, the derivative on the right side is expanded.

$$\begin{aligned} \frac{\partial}{\partial x}\left(\rho\frac{\partial p}{\partial x}\right) &= \frac{\mu}{k}\frac{\partial}{\partial t}\left(\varphi\rho\right) \\ \frac{\partial}{\partial x}\left(\rho\frac{\partial p}{\partial x}\right) &= \frac{\mu}{k}\left(\varphi\frac{\partial \rho}{\partial t} + \rho\frac{\partial \varphi}{\partial t}\right) \end{aligned}$$

In the next step, we introduce another important compressibility, the formation compressibility. This term describes the way the pore volume changes with respect to pressure. From the general definition of compressibility in (2.4), the pore volume compressibility (c_ϕ) can be defined as below. [42].

$$c_\phi = \frac{1}{\varphi}\frac{\partial \varphi}{\partial p} \tag{2.6}$$

Using this definition, we can rewrite the second term of the right side of the equation above in order to have derivatives of pressure with respect to time.

$$\frac{\partial \varphi}{\partial t} = \frac{\partial \varphi}{\partial p}\frac{\partial p}{\partial t} = c_\phi \varphi \frac{\partial p}{\partial t} \tag{2.7}$$

Substituting this and the fluid model in (2.5) and taking the derivatives gives:

$$
\begin{aligned}
\tfrac{\partial}{\partial x}\left(\rho\tfrac{\partial p}{\partial x}\right) &= \tfrac{\mu}{k}\left(\varphi\tfrac{\partial \rho}{\partial t} + \rho c_\phi \varphi \tfrac{\partial p}{\partial t}\right) \\
\tfrac{\partial}{\partial x}\left(\rho_{\mathrm{sc}} e^{c_o(p-p_{sc})}\tfrac{\partial p}{\partial x}\right) &= \tfrac{\mu}{k}\left(\varphi\tfrac{\partial}{\partial t}\left(\rho_{\mathrm{sc}} e^{c_o(p-p_{sc})}\right) + \rho_{\mathrm{sc}} e^{c_o(p-p_{sc})} c_\phi \varphi \tfrac{\partial p}{\partial t}\right) \\
\rho_{\mathrm{sc}}\left(e^{c_o(p-p_{sc})}\tfrac{\partial^2 p}{\partial x^2} + c_o e^{c(p-p_{sc})}\left(\tfrac{\partial p}{\partial x}\right)^2\right) &= \tfrac{\mu}{k}\left(\varphi c_o \rho_{\mathrm{sc}} e^{c_o(p-p_{sc})}\tfrac{\partial p}{\partial t} + \rho_{\mathrm{sc}} e^{c_o(p-p_{sc})} c_\phi \varphi \tfrac{\partial p}{\partial t}\right) \\
\rho_{\mathrm{sc}} e^{c_o(p-p_{sc})}\left(\tfrac{\partial^2 p}{\partial x^2} + c_o\left(\tfrac{\partial p}{\partial x}\right)^2\right) &= \tfrac{\mu\varphi}{k}\rho_{\mathrm{sc}} e^{c_o(p-p_{sc})}\left(c_o\tfrac{\partial p}{\partial t} + c_\phi\tfrac{\partial p}{\partial t}\right) \\
\tfrac{\partial^2 p}{\partial x^2} + c_o\left(\tfrac{\partial p}{\partial x}\right)^2 &= \tfrac{\mu\varphi}{k}\left(c_o\tfrac{\partial p}{\partial t} + c_\phi\tfrac{\partial p}{\partial t}\right)
\end{aligned}
$$

The resulting equation can be simplified by defining a total compressibility $c_t = c_o + c_\phi$. Both terms are considered constant for this work, c_o, due to the assumption of a slightly compressible fluid, and c_ϕ, due to porosity ϕ being assumed constant.

$$
\frac{\partial^2 p}{\partial x^2} + c_o\left(\frac{\partial p}{\partial x}\right)^2 = \varphi c_t \frac{\mu}{k}\frac{\partial p}{\partial t} \tag{2.8}
$$

This equation is nonlinear due to the quadratic gradient term, $c_o(dp/dx)^2$. In the case of a slightly compressible fluid, where the compressibility is small and constant, the quadratic gradient term is often neglected to obtain the linear equation shown below, see [16, 42].

$$
\frac{\partial^2 p}{\partial x^2} = \varphi c_t \frac{\mu}{k}\frac{\partial p}{\partial t} \tag{2.9}
$$

The quadratic gradient term may need to be included when a high pressure gradient exists, as in hydraulic fracturing where fluid is injected at high pressure. [49].

While the focus of this work is on Cartesian coordinates for numerical approximation, the analytic solutions of the equations are more developed for radial coordinates, as they are regularly used for solving flow problems at a single well. For a radial coordinate system, the equations are built in the same way, resulting in the equations below.(2.10)

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} + c_o\left(\frac{\partial p}{\partial r}\right)^2 = \varphi c_t \frac{\mu}{k}\frac{\partial p}{\partial t} \tag{2.10}$$

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} = \varphi c_t \frac{\mu}{k}\frac{\partial p}{\partial t} \tag{2.11}$$

Full Equation Statements

For equations (2.9) the initial condition will always be taken as the pressure in the system before the pressure gradient is applied.

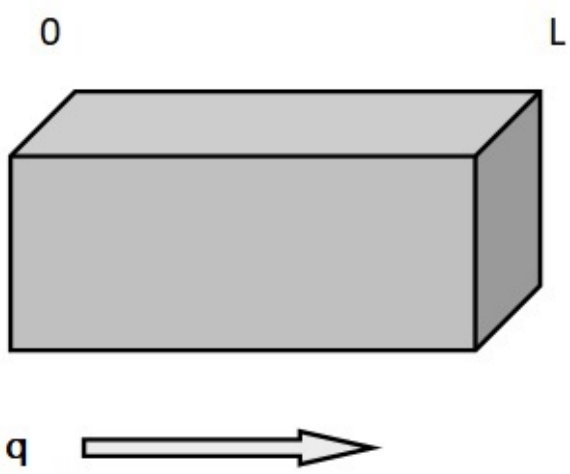

Figure 2.1: Rectangular flow model with a flow direction of left to right indicated. The pressure in this case is monotonically increasing from right to left

The boundary conditions are given at the inlet and outlet for the rectangular model, where $x = 0$ and $x = L$ as in Figure 2.1. Options for the boundary condition are a fixed pressure or a fixed flow rate at the inlet and outlet. The flow rate q is acceptable as a boundary condition as Darcy's Law (1.3) gives it as a constant times the pressure gradient

dp/dx at the boundary location. Pressure boundary conditions are used in this work.

$$p(x,0) = p_{\text{init}} \tag{2.12}$$

$$p(0,t) = p_{\text{left}} \tag{2.13}$$

$$p(L,t) = p_{\text{right}} \tag{2.14}$$

Analytic Solutions

Limited analytic solutions have been developed for the flow model in equation (2.9) in Cartesian coordinates for a rectangular domain, as shown in Figure 2.1. However, numerous analytic solutions exist for the radial flow model in equation (2.20) with appropriate boundary and initial conditions. This is likely due to the application of radial models in research areas such as pressure transient analysis, where the pressure change with respect to time is compared with the model predictions to estimate parameters such as permeability.

Analytic Solution of the Rectangular Flow Model

Katz [16] presented a solution for the equation(2.9) without the quadratic gradient term, based on a solution of the heat equation for temperature in a semi-infinite solid during the given by Churchill [6]. The solution utilizes a pressure boundary condition on the left at $x = 0$, and a semi-infinite condition on the right where as x goes to infinity, the pressure goes to the initial pressure.

The solution $p(x,t)$ gives the pressure at any point x and time t when the pressure at the face $x = 0$ is suddenly dropped from the initial pressure $p(x,0)$ to the left boundary pressure $p(0,t)$. This produces flow moving from right to left as seen in Figure 2.2.

$$\frac{\partial^2 p}{\partial x^2} = \varphi c_t \frac{\mu}{k} \frac{\partial p}{\partial t} \tag{2.9}$$

$$p(x,0) = p_{\text{init}} \tag{2.15}$$

$$p(0,t) = p_{\text{left}} \tag{2.16}$$

$$\lim_{x\to\infty} p(x,t) = p(x,0) \tag{2.17}$$

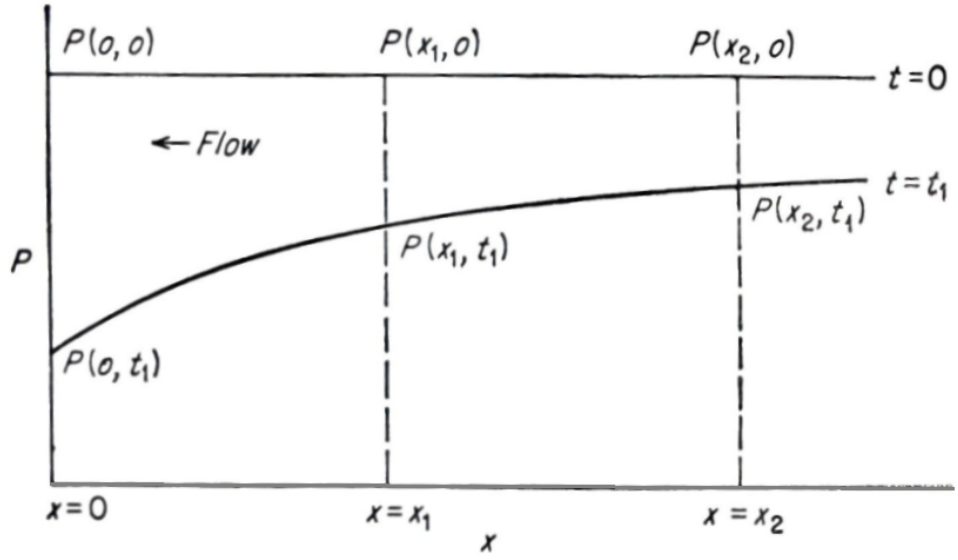

Figure 2.2: An image of the domain at initial pressure and at later time t_1 from Katz [16] showing flow from right to left.

To apply the heat equation solution, Katz defined a dimensionless time for linear oil flow in US customary field units.

$$t_D = \frac{2.634 \times 10^{-4} kt}{\varphi \mu c_t x^2} \tag{2.18}$$

$$\frac{p(x,t) - p_{\text{left}}}{p_{\text{init}} - p_{\text{left}}} = -erfc\frac{1}{2{t_D}^{1/2}} \tag{2.19}$$

This analytic solution is used to verify the code that produces the numerical solution of the conventional flow equation during the transient flow period by using a semi-infinite domain.

Analytic Solutions of the Radial Flow Model

Analytic solutions were developed for the radial flow model given in equation (2.20) by van Everdingen and Hurst [44] without the inclusion of the quadratic gradient term. Solutions for the radial flow model with the inclusion of the quadratic gradient term were developed by Chakrabarty, et al. and Odeh and Babu [5, 34]. An analytic solution is shown for the model below with the boundary conditions of a constant flow rate ($q(r_w, t)$) at the wellbore and a constant flow rate of zero at the outer boundary (corresponding to a closed outer boundary).

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} = \varphi c_t \frac{\mu}{k}\frac{\partial p}{\partial t} \tag{2.20}$$

$$p(r, 0) = p_{\text{init}} \tag{2.21}$$

$$q(r_w, t) = q_{\text{wellbore}} \tag{2.22}$$

$$0 = q_{\text{outer boundary}} \tag{2.23}$$

This solution was developed by expressing equation (2.20) in terms of the following dimensionless parameters.

$$t_{Dw} \equiv \frac{kt}{\varphi \mu c_t r_w^2} \tag{2.24}$$

$$r_{eD} \equiv \frac{r_e}{r_w} \tag{2.25}$$

$$p_D \equiv \frac{2\pi kh(p_{\text{init}} - p)}{q_{\text{wellbore}} B\mu} \tag{2.26}$$

where:

- r_e is the drainage area radius
- r_w is the wellbore radius
- h is the formation thickness
- B is the formation volume factor, the ratio of volume of a fluid in the reservoir to the volume that the same mass of fluid occupies at standard conditions

In terms of these dimensionless parameters, an analytic solution was developed by van Everdingen and Hurst [44] for $r = r_w$ as that is the only location where the pressure can be measured.

$$p_D(r_w, t) = \frac{2\left(t_{Dw} + 0.25\right)}{r_{eD}^2} - \frac{3r_{eD}^4 - 4r_{eD}^4 \ln r_{eD} - 2r_{eD}^2 - 1}{4(r_{eD}^2 - 1)^2} + 2\sum_{n=1}^{\infty} \frac{e^{-a_n^2 t_{Dw}} J_1^2(a_n r_{eD})}{a_n^2 \left(J_1^2(a_n r_{eD}) - J_1^2(a_n)\right)} \tag{2.27}$$

where a_n are the roots of the equation:

$$J_1\left(a_n r_{eD}\right) Y_1(a_n) - J_1(a_n) Y_1(a_n r_{eD}) = 0 \tag{2.28}$$

Here J_1 is the first-order Bessel function of the first kind and Y_1 is the first-order Bessel function of the second kind. Since the analytic solution in (2.27) is difficult to evaluate, van Everdingen and Hurst presented tabulated solutions [44]. Approximations to this solution were developed by Lee [20] using the same dimensionless parameters. The first approximation is valid in the infinite acting period before the pressure disturbance reaches the drainage area boundary. This ends when dimensionless time equals one-fourth of the square of the

dimensionless radius [20, 42]. The second approximation is when the pressure disturbance reaches the outer boundary of the drainage area and the well is said to be finite-acting

$$\begin{aligned} t_{Dw} &< \frac{1}{4} r_{eD}^2, \\ p_D(r_w, t_{Dw}(t)) &= 0.5(\ln t_{Dw} + 0.80907). \end{aligned} \tag{2.29}$$

$$\begin{aligned} t_{Dw} &> \frac{1}{4} r_{eD}^2, \\ p_D(r_w, t_{Dw}(t)) &= \frac{2t_{Dw}}{r_{eD}^2} + \ln r_{eD} - \frac{3}{4}. \end{aligned} \tag{2.30}$$

Most of the analytic solutions for the flow equation, including the ones above, assume that the quadratic term is negligible. To assume that it is negligible requires either a small fluid compressibility c_o or a small pressure gradient $(\frac{\partial p}{\partial r})^2$. The assumption of a slightly compressible fluid, requiring a small and constant compressibility, allows for a small fluid compressibility, but there are some special cases where it is believed that the pressure gradient is large enough to cause sufficient error such as in some types of pressure transient analysis and in high drawdown flow, where a large pressure gradient is needed to produce flow [5, 34].

Chakrabarty et al. [5] developed analytic solutions for the radial and Cartesian flow models with the quadratic gradient term for equations (2.8) and (2.10), with their associated boundary conditions (2.12),(2.13),(2.14),(2.21),(2.22), and (2.23).

The dimensionless parameters used to solve this model were the same as those used above (2.18),(2.24),(2.25), and (2.26); however, an additional dimensionless parameter, α, is needed for the nonlinear term.

$$\alpha = -\frac{q_{\text{wellbore}} \mu c_o}{2\pi k h} \tag{2.31}$$

With q assumed constant as a boundary condition, and the parameters μ, k, and h continuing to be assumed to be constant, α is directly proportional to c_o. Comparisons of the solutions with and without the quadratic pressure gradient terms are given as functions of α. In the

following plot of the solutions, dimensionless pressure is plotted versus dimensionless time. While pressure decreases at the wellbore as a function of time, dimensionless pressure is defined so that it increases as wellbore pressure decreases. This graph indicates an increasing difference between the two models as α increases.

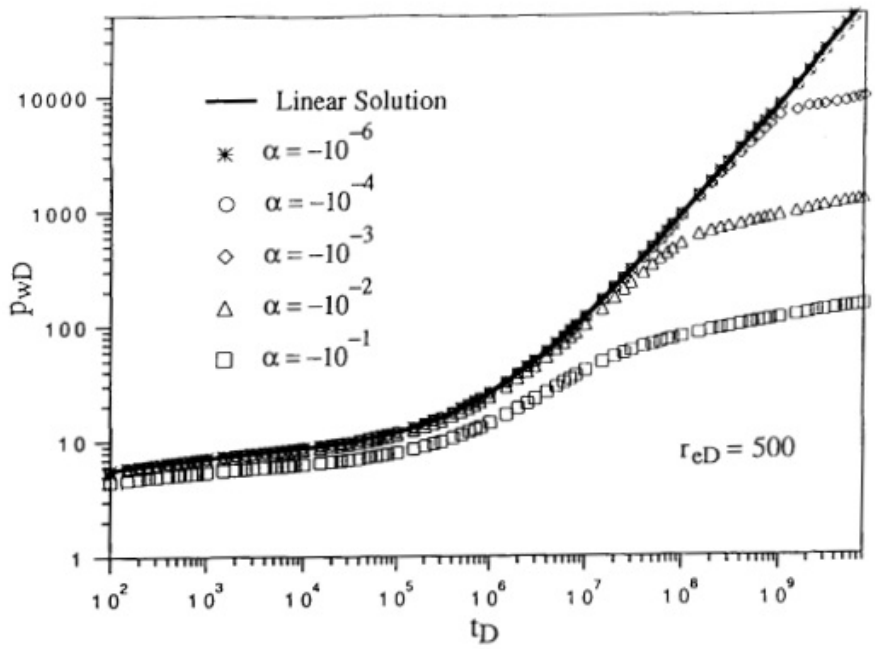

Figure 2.3: A comparison of analytic solutions for the radial flow model with and without the quadratic gradient term, showing the effect of larger values of α [5].

18

LOW PERMEABILITY FLOW BEHAVIOR

Models based on Darcy's Law have proven instrumental in advancing the understanding of conventional oil and gas reservoirs, and they serve as a foundation for laboratory methods used to determine permeability. In these experiments, equation (1.3) is commonly applied to calculate permeability. The laboratory setup typically involves an incompressible fluid—such as water—flowing through a rock core, with measurements taken once a steady-state flow is established. Given known values for the pressure gradient and measured flow rate, along with the core sample's length, cross-sectional area, and the fluid's viscosity, the equation can be rearranged to solve for permeability. However, several studies have noted that even under these controlled conditions, Darcy's Law fails to yield accurate permeability values for low-permeability rocks [14, 21, 38]. This limitation suggests potential challenges in applying Darcy-based models to unconventional reservoirs where low permeability is a defining characteristic.

According to Darcy's Law, even a minimal pressure gradient should generate a measurable flow rate. However, empirical evidence shows that flow often does not commence until a critical threshold pressure gradient is exceeded. Before the threshold pressure gradient is reached, the relationship between pressure gradient and flow rate may exhibit nonlinear behavior [2, 21]. This initial phase of nonlinear response is often referred to as *pre-Darcy* flow [38].

Multiple methods of modeling the pre-Darcy flow have been proposed [41]. In the threshold pressure gradient model used in this work, no pre-Darcy flow occurs, with the flow rate being linearly related to the pressure gradient once the threshold pressure gradient has been achieved. Other models of pre-Darcy flows give a smooth, nonlinear relation between flow and pressure gradient, becoming linear after the threshold pressure gradient is reached. Both types of models are represented in Figure 3.1 [45].

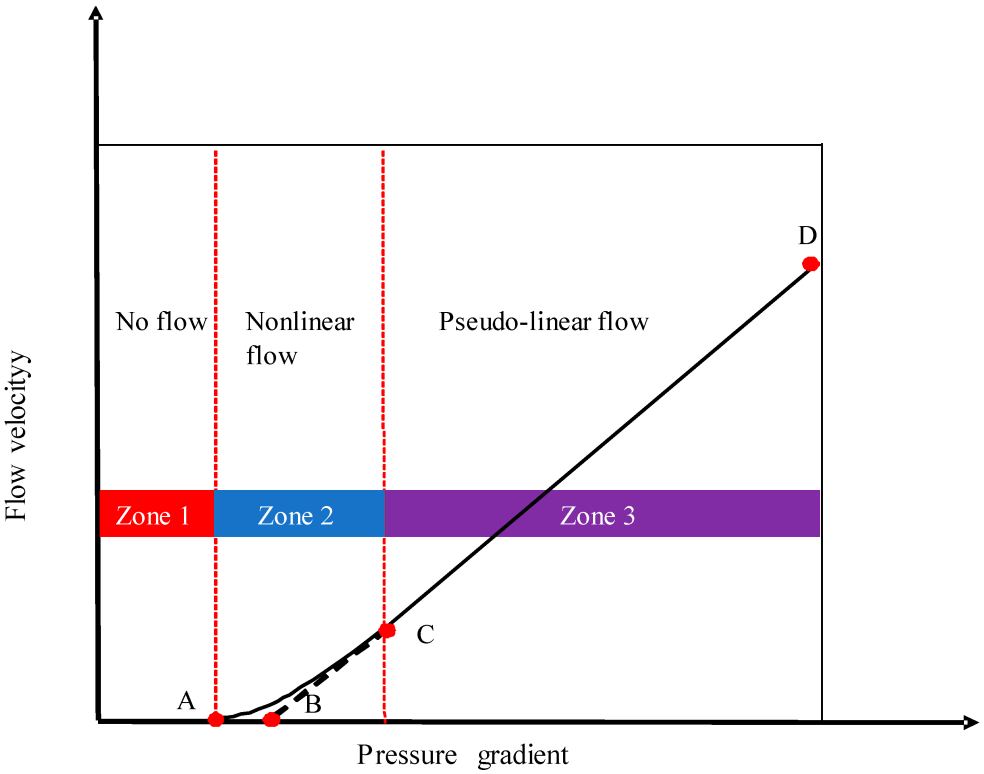

Figure 3.1: Pre-Darcy flow behavior, modeled as either a smooth transition to linear flow or a threshold pressure [45].

Threshold Pressure Gradient Model

To model the flow threshold pressure gradient, with no flow in the pre-Darcy period, the following piecewise expression for the flow velocity (3.1) has been proposed by Prada and other authors [14, 38] to replace Darcy's Law, where G represents a threshold pressure gradeint (taken as a positive number).

$$v = \begin{cases} 0, & \frac{dp}{dx} < G \\ \frac{-k}{\mu}\left(\frac{dp}{dx} - G\right), & \frac{dp}{dx} \geq G \end{cases} \tag{3.1}$$

Prada [38] performed a number of experiments to measure the threshold pressure gradient on different core samples with the tabulated results show in Figures 6.10, 6.11

and 6.12. For the core with the lowest permeability to viscosity ratio, a shaly sand, they found a threshold pressure gradient of 0.813 psi/cm. These data for the shaly sand are used in numerical approximations in Ch. 6.

Huang [14], following work by Jiang [15], also proposed a model for fluid velocity (3.2) based on two parameters, A and B, shown in Figure 3.1. Note that when $B = 0$, equation (3.2) reduces to Darcy's Law.

$$v = -\frac{k}{\mu}\frac{\partial p}{\partial x}\left(1 - \frac{B}{\frac{\partial p}{\partial x} + A - B}\right) \tag{3.2}$$

Mechanisms Behind Pre-Darcy Flow

Multiple factors are believed to cause pre-Darcy flow behavior in low-permeability porous rock. In porous materials, fluid flows through the connected pores, so the permeability is partially dependent on the connections between pores, the pore-throat size in these connections, and the length of the path that fluid follows to move linearly in the direction of decreasing pressure. So while permeability requires porosity, there are not always good

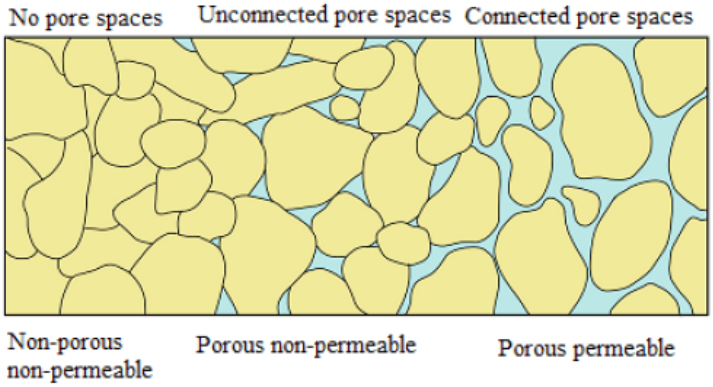

Figure 3.2: An illustration of the relationship between porosity and permeability, showing porous but non-permeable rock [1] This middle section illustrates what could be considered unconventional reservoir rock, capable of being produced with newer techniques.

correlations between the two parameters [31].

One key factor in low permeability materials is pore throat size. In unconventional low permeability reservoirs such as the Eagle Ford and Bakken, nanopores with pore throat radii ranging from 2 nm to 15 nm make up a significant amount of the rock's natural porosity [40]. Small pore throat sizes reduce the rock's ability to transmit fluids by effectively raising fluid viscosity, among other effects. The Kozeny Equation (3.3) can be used to relate pore throat size (r) in nanometers (nm) to permeability in millidarcies (md) and porosity [40]. This allows estimates of pore throat size for understanding the viscous effects.

$$r = 31.45\sqrt{\frac{8k}{\phi}} \tag{3.3}$$

Narrow pore throats can increase effective viscosity because of the no-slip condition: "When fluid is in contact with a solid body, the velocity of the fluid at the point of contact is the same as the velocity of the solid body at the same point.[8]" This is illustrated in Figure 3.3. The fluid in contact with the boundary is stationary, and the fluid near the boundary

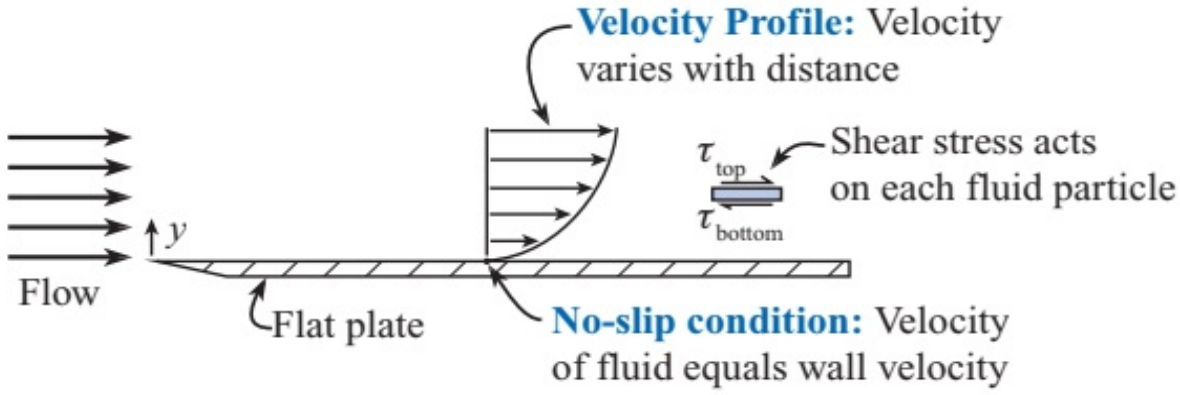

Figure 3.3: Flowing fluid interacts with a solid boundary illustrating the no-slip condition [8].

moves at a slower velocity than that of the bulk fluid. When the fluid is flowing between two solids that are close together, the effect of the boundary on the flow rate is pronounced.

The flowing fluid can be classified as boundary fluid with a high resistance to flow caused by the no-slip condition, or bulk fluid, farther from the stationary boundary. When the pores are small, the boundary fluid can contribute to pre-Darcy flow behavior. With the small

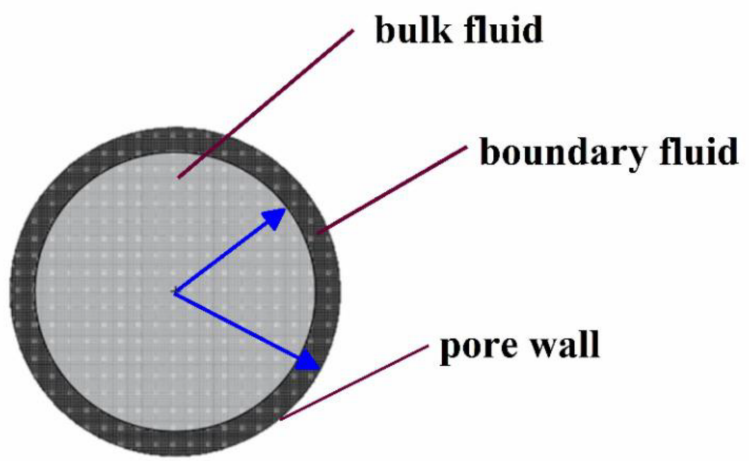

Figure 3.4: Illustration of bulk and boundary fluid in a pore throat [48]

.

diameter of the pore throats, capillary action can also be a contributing factor to pre-Darcy flow [41]. Capillary action is the tendency of a fluid to move into small openings, independent of gravity or a pressure gradient [8]. To overcome this action, a larger pressure gradient may be required [41].

While other factors such as interactions between the fluid and the surface, and the length of the fluid path [41], viscous and capillary effects play a significant role.

AN UNCONVENTIONAL TRANSIENT FLOW EQUATION

The Development of the Unconventional Flow Equation

To develop the partial differential equations for transient, slightly compressible flow in a low-permeability rock, a similar procedure will be used as when creating the conventional flow model in Chapter 3. In this case, the model is built by using the threshold gradient equation of (3.1) in place of Darcy's Law and restricting the problem to where the threshold pressure gradient has been met. A crucial assumption of the model is that where the threshold pressure gradient (G) is not met, the flow velocity remains zero. Where it is met, the flow velocity obeys the following relationship.

$$v = \frac{-k}{\mu}\left(\frac{dp}{dx} - G\right), \quad \text{where } \frac{dp}{dx} \geq G \tag{4.1}$$

As in the development of equation (2.9), we substitute the velocity defined above for the velocity in the conservation of mass equation, which holds for x such that $\frac{dp}{dx} \geq G$.

$$\frac{\partial}{\partial x}\left(\frac{k}{\mu}\left(\frac{dp}{dx} - G\right)\rho\right) = \varphi\frac{\partial \rho}{\partial t} + \rho c_\phi \varphi \frac{\partial p}{\partial t}$$

Applying the definition of density of a slightly compressible fluid from (2.5):

$$\begin{aligned}
\frac{\partial}{\partial x}\left(\frac{k}{\mu}\rho_{\text{sc}}e^{c_o(p-p_{sc})}\left(\frac{dp}{dx}-G\right)\right) &= \varphi\frac{\partial}{\partial t}\left(\rho_{\text{sc}}e^{c_o(p-p_{sc})}\right)+\rho_{\text{sc}}e^{c_o(p-p_{sc})}c_\phi\varphi\frac{\partial p}{\partial t}\\
\frac{\partial}{\partial x}\left(\rho_{\text{sc}}e^{c_o(p-p_{sc})}\left(\frac{dp}{dx}-G\right)\right) &= \frac{\mu}{k}\left(\varphi\frac{\partial}{\partial t}\left(\rho_{\text{sc}}e^{c_o(p-p_{sc})}\right)+\rho_{\text{sc}}e^{c_o(p-p_{sc})}c_\phi\varphi\frac{\partial p}{\partial t}\right)\\
\frac{\partial}{\partial x}\left(\rho_{\text{sc}}e^{c_o(p-p_{sc})}\frac{dp}{dx}\right)-\frac{\partial}{\partial x}\left(\rho_{\text{sc}}e^{c_o(p-p_{sc})}G\right) &= \rho_{\text{sc}}e^{c_o(p-p_{sc})}\frac{\mu}{k}\left(\varphi c_o\frac{\partial p}{\partial t}+c_\phi\varphi\frac{\partial p}{\partial t}\right)\\
\rho_{\text{sc}}e^{c_o(p-p_{sc})}\left(\frac{\partial^2 p}{\partial x^2}+c_o\left(\frac{\partial p}{\partial x}\right)^2-c_o\,G\frac{\partial p}{\partial x}\right) &= \rho_{\text{sc}}e^{c_o(p-p_{sc})}\frac{\mu}{k}\left(\varphi c_t\frac{\partial p}{\partial t}\right)\\
\frac{\partial^2 p}{\partial x^2}+c_o\left(\frac{\partial p}{\partial x}\right)^2-c_o\,G\frac{\partial p}{\partial x} &= \frac{\mu}{k}\left(\varphi c_t\frac{\partial p}{\partial t}\right)
\end{aligned}$$

The resulting equation differs from the final form of the conventional flow equation (2.9) with the appearance of the "small term" ($c_oG\frac{\partial p}{\partial x}$), and the quadratic gradient term (QGT). The small term is assumed to be negligible [26, 42]. Equation (4.3) results from this assumption. The assumption that the QGT can be neglected for slightly compressible fluids is used to linearize equation (2.8) and facilitate the derivation of analytic solutions. This relies on the fluid model having a small compressibility c_o, along with the assumption of a small pressure gradient. However, it has been noted that in cases with higher compressibility or steeper pressure gradients, this assumption may introduce significant error. Such concerns are particularly relevant in low-permeability systems, where larger pressure gradients are required to sustain flow. Therefore, retaining the QGT term may be advisable in these scenarios [26]. Assuming both the QGT and the small term are insignificant gives (4.4).

$$\frac{\partial^2 p}{\partial x^2}+\underbrace{c_o\left(\frac{\partial p}{\partial x}\right)^2}_{\text{QGT}}-\underbrace{c_o\,G\,\frac{\partial p}{\partial x}}_{\text{Small Term}}=\frac{\mu\varphi c_t}{k}\frac{\partial p}{\partial t},\quad \forall\ x\ \ni\ \frac{dp}{dx}\geq G \tag{4.2}$$

$$\frac{\partial^2 p}{\partial x^2} + c_o \left(\frac{\partial p}{\partial x}\right)^2 = \frac{\mu \varphi c_t}{k} \frac{\partial p}{\partial t} \tag{4.3}$$

$$\frac{\partial^2 p}{\partial x^2} = \frac{\mu \varphi c_t}{k} \frac{\partial p}{\partial t} \tag{4.4}$$

In all three cases, the boundary conditions are formatted to include the threshold gradient. Figure 4.1 shows how initially, the model is at the initial pressure $p(x, 0)$. When flow is initiated, the pressure on the right is dropped from the initial pressure to the right boundary pressure $p(L, t)$, initiating fluid flow from left to right. However, the moving boundary on the left, indicated by the vertical line in the second image for $t > 0$, moves left proportionally to a rate at which the threshold pressure gradient is achieved. The boundary moves left according to a function $L - s(t)$ where $s(t)$ runs from 0 to L. Here, the function $s(t)$ defines the length of the domain over which the model equation holds at time t.

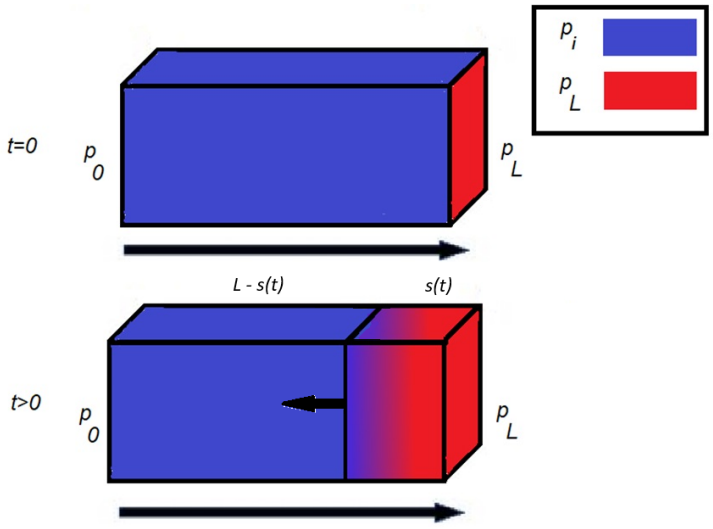

Figure 4.1: Illustration of a moving boundary condition

Including the moving boundary, the initial and boundary conditions, which are the same for all three cases of equation (4.2) are given as follows.

Initial conditions:

$$\begin{aligned} p(x,0) &= p_{\text{init}} \\ s(0) &= 0 \end{aligned} \tag{4.5}$$

Right boundary condition:

$$p(L,t) = p_{\text{right}} \tag{4.6}$$

Moving left boundary conditions:

$$\begin{aligned} p(L-s(t),t) &= p(0,t) = p_{\text{left}} \\ \tfrac{\partial p}{\partial x}\big|_{L-s(t),t} &= -G \end{aligned} \tag{4.7}$$

Analytic Solution of the Unconventional Flow Equation

An analytic solution for the case with the moving boundary condition produced by the threshold pressure gradient is given by Liu, et al. [28]. When constructing the solution, the authors define the dimensionless variables given in Table 4.1, with v_w in the work defined by the authors as the flow velocity in (3.1) and set as a constant in the lab experiments, and all definitions holding only when $\frac{dp}{dx} \geq G$. In Liu's work, with the flow moving from right to left, the function $s(t)$ is used instead of $L - s(t)$ to describe the position of the moving boundary.

With these parameters, the equation and boundary conditions are as follows, where the dimensionless pressure U is a function of X and T.

$$\frac{\partial^2 U}{\partial X^2} = \frac{\partial U}{\partial T}, \quad 0 \leq X \leq \delta(T) \tag{4.8}$$

$$\delta(0) = 0 \tag{4.9}$$

Variable	Dimensionless quantity
$T = \frac{kt}{\mu\phi c_t L^2}$	time
$X = \frac{x}{L}$	position
$\delta = \frac{s}{L}$	moving boundary position
$U = \frac{k(p_{init}-p)}{v_w L\mu}$	pressure
$U_L = \frac{k(p_{init}-p_L)}{v_w L\mu}$	pressure
$\Lambda = \frac{kG}{v_w\mu}$	threshold pressure gradient

Table 4.1: Notation from Liu et al. [28].

$$U(X,0) = 0 \tag{4.10}$$

$$U(0,T) = U_L \tag{4.11}$$

$$U(\delta(T),T) = 0 \tag{4.12}$$

$$\left.\frac{\partial U}{\partial X}\right|_{X=\delta(T)} = -\Lambda \tag{4.13}$$

With these equations, the authors are able to derive an expression for the velocity of the moving boundary condition as proportional to the second derivative of dimensionless pressure U with respect to dimensionless position X.

$$\frac{d\delta(T)}{dT} = \frac{1}{\Lambda}\left.\frac{\partial^2 U}{\partial X^2}\right|_{X=\delta(T)} \tag{4.14}$$

Liu, et.al.[28] note that equation (4.14) is different from what is seen with the heat equation with a moving boundary, in the sense that the velocity of the moving boundary is being related to the second derivative instead of the first, as with a classic Stefan problem. The Stefan problem describes a class of problems with a moving boundary caused by a

liquid-solid phase change, such as melting or freezing [18]. An example is heating a solid slab of metal initially at a constant temperature by raising the temperature at one end to above the melting point. The interface between the solid and the liquid part of the slab moves in the direction away from the heated end [13]. In the case of the moving boundary being the interface between the melted and solid metal, the velocity of the moving boundary is proportional to the first derivative of the dependent variable, in this case, temperature (T) instead of pressure (U).

While the formulation of the moving boundary is different in this case, the approach to solving the Stefan problem in [13] motivates Liu's work in [28]. The core idea is to look for a similarity transform to convert partial differential equations to ordinary differential equations, and by creating similarity variables, both solve the resulting ordinary differential equations with the results given in Figure 4.2 for a range of values of Λ.

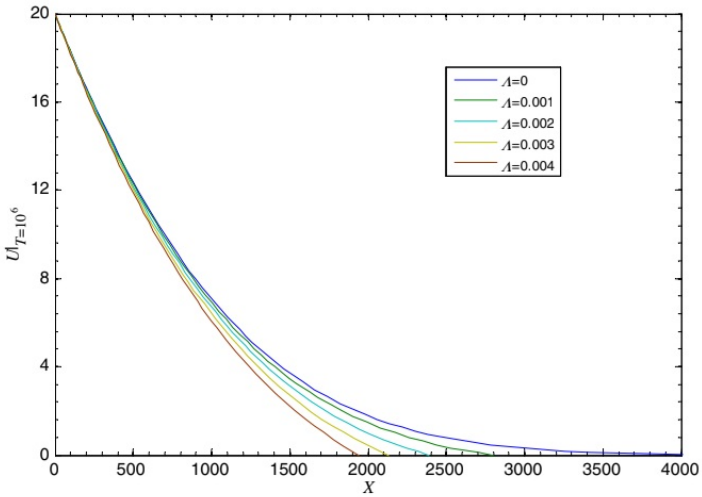

Figure 4.2: Analytic solutions of dimensionless pressure U vs dimensionless position X with various dimensionless threshold pressure gradients [28].

DIMENSIONAL ANALYSIS

To aid in the analysis of numerical models of the flow equations (2.9,4.3) it is beneficial to fully write these models in a non-dimensional form. While authors working on analytic solutions, such as Katz [16], Van Everdingen and Hurst [44], and Liu et. al. [28] have provided dimensionless parameters used in their work, this section develops a verified set of dimensionless parameters that are consistent between the conventional and the unconventional model.

This is done with the use of the Buckingham Π theorem and insights from the dimensionless parameters used to find analytic solutions as in equations (2.19) and (2.27). Verification of the dimensionless parameters developed can be found in Appendix B.

The Buckingham Π Theorem states that any group of n parameters can be arranged into $n-m$ dimensionless groups where m is the number of independent fundamental dimensions [8]. These groups are called Π groups and can be determined by finding an appropriate arrangement of the exponents $a_1, a_2, a_3, \ldots a_7$, in the expression below [29].

Dimensionless Form of the Conventional Flow Model

There are seven relevant parameters with units in the flow equation given in (2.9). The $n = 7$ parameters are given below. The terms in the second derivative are expressed here as Δp as the change in pressure over the length L. There is one unitless parameter ϕ which will be placed appropriately in the dimensionless form once the Π groups are formed. The parameters for the model considered in this work are Δp, x, μ, k, t, c_t, and L .

To find a set of exponents to define a Π group, each parameter is expressed in terms of the fundamental dimensions of force (F), length(L) and time (t). The FLT (Force, Length, and Time) system is chosen here instead of the alternative MLT system (Mass, Length and Time) as this simplifies expressing pressure, compressibility and viscosity in fundamental

dimensions. [39] For example Δp has the fundamental dimension F/L^2 or $F^1L^{-2}t^0$. We begin by defining Π as given below and requiring that it is a dimensionless parameter.

$$\Pi = \Delta p^{a_1} \;\; x^{a_2} \;\; \mu^{a_3} \;\; k^{a_4} \;\; t^{a_5} \;\; {c_t}^{a_6} \;\; L^{a_7} \tag{5.1}$$

$$\Pi = (F^1L^{-2}t^0)^{a_1}(F^0L^1t^0)^{a_2}(F^1L^{-2}t^1)^{a_3}(F^0L^2t^0)^{a_4}(F^0L^0t^1)^{a_5}(F^{-1}L^2t^0)^{a_6}(F^0L^1t^0)^{a_7} \tag{5.2}$$

When all like terms are collected, the exponent of each fundamental dimension must be zero, and this leads to a homogeneous system of linear equations in terms of the unknown exponents. Expressing the system in matrix form, the system matrix is built with each column representing a parameter in the order listed above, and each row representing the restriction that each exponent is zero. Once the matrix is built as below and the following system of equations is solved, a description of the elements of the null space of the matrix will define our choices for the dimensionless quantities for the problem.

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & -1 & 0 \\ -2 & 1 & -2 & 2 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

We consider the underdetermined system of linear equations and since there are an infinite number of solutions to the system, there is some freedom to choose appropriate values of four of the seven exponents.

$$a_1 + a_3 - a_6 = 0 \tag{5.3}$$

$$-2a_1 + a_2 - 2a_3 + 2a_4 + 2a_6 + a_7 = 0 \tag{5.4}$$

$$a_3 + a_5 = 0 \tag{5.5}$$

Choosing the basic variables to be a_1, a_2, a_3, the free variables are then a_4, a_5, a_6 and a_7. Hence, one can express the algebraic equations above in reduced echelon form in terms of the free variables as described in [30].

$$a_1 = a_5 + a_6 \tag{5.6}$$

$$a_2 = -2a_4 - a_7 \tag{5.7}$$

$$a_3 = -a_5 \tag{5.8}$$

The general solution of the matrix equation above can be described by the following

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = a_4 \begin{bmatrix} 0 \\ -2 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + a_5 \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + a_6 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + a_7 \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

for any choice of $a_4, a_5, a_6, a_7 \in \mathbb{R}$.

This expression allows us to choose the appropriate scaling for the time, space, and

pressure variables. The dimensionless parameters used by Van Everdingen and Hurst in equations (2.24), (2.25), (2.26) can provide hints for finding a useful solution set [44].

The dimensionless terms for the linear model are calculated below. The porosity term, being unitless, can be included where needed so that the flow equation can be written in dimensionless form. The first target is a dimensionless time. To find an appropriate formula for this parameter, the time exponent a_5 is set to 1, and the position and pressure exponents are set to 0. Solutions are found for the other four exponents using integer values. This results in the dimensionless parameter Π_1 given below.

$$a_1 = 0, a_2 = 0, a_3 = -1, a_4 = 1, a_5 = 1, a_6 = -1, a_7 = -2$$
$$\Pi_1 = \quad \Delta p^0 \quad x^0 \quad \mu^{-1} \quad k^1 \quad t^1 \quad {c_t}^{-1} \quad L^{-2}$$
$$\Pi_1 = \frac{kt}{\mu c_t L^2} = \frac{kt}{\varphi \mu c_t L^2}$$
$$t_D = \frac{t}{\frac{\varphi \mu c_t L^2}{k}}$$

For dimensionless position x_D, the exponent for x was set to 1, and the Δp and t exponents were set to 0. This resulted in the dimensionless parameter Π_2 given below.

$$a_1 = 0, a_2 = 1, a_3 = 0, a_4 = 0, a_5 = 0, a_6 = 0, a_7 = -1$$
$$\Pi_2 = \quad \Delta p^0 \quad x^1 \quad \mu^0 \quad k^0 \quad t^0 \quad {c_t}^0 \quad L^{-1}$$
$$\Pi_2 = \frac{x}{L}$$
$$x_D = \frac{x}{L}$$

To obtain a dimensionless pressure, the Δp exponent is set to 1, and the time t and position x exponents are set to 0. Solutions were found for the other exponents using integer

values. This resulted in Π_3

$$a_1 = 1, a_2 = 0, a_3 = 0, a_4 = 1, a_5 = 0, a_6 = 1, a_7 = -2$$
$$\Pi_3 = \quad \Delta p^1 \quad x^0 \quad \mu^0 \quad k^1 \quad t^0 \quad {c_t}^1 \quad L^{-2}$$
$$\Pi_3 = \frac{kc_t \Delta p}{L^2} = \frac{kc_t(p_{init} - p)}{L^2}$$
$$p_D = \frac{kc_t(p_{init} - p)}{L^2}$$

These three dimensionless parameters are then used to write the dimensionless linear model.

$$t_D = \frac{t}{\frac{\varphi \mu c_t L^2}{k}} \tag{5.9}$$

$$x_D = \frac{x}{L} \tag{5.10}$$

$$p_D = \frac{kc_t(p_{init} - p)}{L^2} \tag{5.11}$$

are then used to write the full dimensionless form of (2.9), the conventional flow model without the application of the threshold pressure gradient.

$$\frac{\partial^2 p_D}{\partial {x_D}^2} = \frac{\partial p_D}{\partial t_D}, \quad 0 < x_D < 1 \tag{5.12}$$

$$p_D(0, t_D) = 0, \qquad p_D(1, t_D) = P_{\text{right}D}, \quad t_D > 0 \tag{5.13}$$

$$p_D(x_D, 0) = 0, \quad 0 < x_D < 1 \tag{5.14}$$

Dimensionless Form of the Unconventional Flow Model

To build this dimensionless model, the dimensionless parameters in (5.9 - 5.11) for the conventional model are used as a starting point, with additional parameters introduced to address the new terms that are introduced in equations (4.3-4.4), with initial and boundary

conditions (4.5, 4.6, and 4.7).

These additional terms are: a time-dependent parameter, s (L), representing the location of the moving boundary at time t; and G $(P/L = F/L^3)$, a threshold pressure gradient. This indicates that there are two additional dimensionless Π groups. The oil compressibility c_o $(1/P = L^2/F)$, which is included in the total compressibility: $c_t = c_o + c_\phi$, appears as a coefficient in cases where the quadratic pressure gradient is included, such as in (4.3). Separating the oil compressibility out adds another relevant parameter, so to fully write the unconventional equations in (4.3) and (4.4), along with the boundary and initial conditions requires a third dimensionless Π group.

The required dimensionless parameters to non-dimensionalize the unconventional flow equations are dimensionless s_D, marking where the threshold pressure gradient is met, G_D, the dimensionless threshold pressure gradient, and c_{oD}, dimensionless oil compressibility.

The approach to finding dimensional parameters, or Π groups, involves solving an underdetermined system, allowing for the tailoring of the dimensionless parameters to different situations. For example, Liu et al. [28] developed dimensionless parameters for an analytic solution for a one-dimensional flow problem with a threshold pressure gradient using lab-scale parameters. They developed a dimensionless threshold pressure gradient involving a constant in Table 4.1 that does not apply to field-scale problems.

In this case, each dimensionless parameter should be expressed in terms of the relevant quantities in the equations and tailored to yield the fully dimensionless PDE. Candidates for the new dimensionless parameters are proposed and then verified by demonstrating that, together with the previously defined t_D, x_D, and p_D (see equations (5.9), (5.10), and (5.11)), they produce a fully dimensionless PDE.

The new dimensionless parameters were developed based on their unit types and those of the previously developed ones. The parameter s has the dimension of length, and a ratio

of lengths is proposed that ensures the movement of the boundary in the correct direction.

$$s_D = \frac{s}{L} \tag{5.15}$$

The parameter G has the unit type pressure divided by length, and the dimensionless form of this parameter is structured similarly to the dimensionless pressure definition.

$$G_D = \frac{kc_tG}{L} \tag{5.16}$$

The unit type for a compressibility term like c_o is a reciprocal of pressure. That led to the development of its dimensionless form by considering the reciprocal of dimensionless pressure.

$$c_{oD} = \frac{L^2c_o}{kc_t} \tag{5.17}$$

Verification that the six parameters defined above produce the full set of dimensionless equations is given in Appendix B.

The final model, with all three options for the PDE, is given below.

$$\begin{aligned}
&\tfrac{\partial^2 p_D}{\partial {x_D}^2} = \tfrac{\partial p_D}{\partial t_D} \\
&\tfrac{\partial^2 p_D}{\partial {x_D}^2} - c_{oD}\left(\tfrac{\partial p_D}{\partial x_D}\right)^2 = \tfrac{\partial p_D}{\partial t_D} \\
&\tfrac{\partial^2 p_D}{\partial {x_D}^2} - c_{oD}\left(\tfrac{\partial p_D}{\partial x_D}\right)^2 - G_D c_{oD}\tfrac{\partial p_D}{\partial x_D} = \tfrac{\partial p_D}{\partial t_D},
\end{aligned} \tag{5.18}$$

Initial conditions

$$\begin{aligned}
&p_D(x_D, 0) = 0 \quad 0 < x_D < 1 \\
&s_D(t_D = 0) = 0
\end{aligned} \tag{5.19}$$

Stationary right boundary condition

$$p_D(1, t_D) = P_{\text{right}D}, \quad t_D > 0 \tag{5.20}$$

Moving left boundary conditions

$$\begin{aligned} p_D(1 - s_D(t_D), t_D) &= 0, \quad t_D > 0 \\ \left.\frac{\partial p_D}{\partial x_D}\right|_{1-s_D(t_D)} &= -G_D, \quad t_D > 0 \end{aligned} \tag{5.21}$$

NUMERICAL APPROXIMATIONS FOR THE FLOW MODELS

For the model simulations presented in this work, a simplified finite volume method is proposed to solve the flow model in equations (5.12), and (5.18) with its boundary and initial conditions given in (5.19), (5.20), and (5.21). In order to solve the equations in (5.12) that include a moving boundary condition, we implement a working grid for computation and a dynamic reporting grid to capture boundary evolution. The reporting grid reflects pressure changes only when the pressure gradient in the working grid reaches a specified threshold, thereby modeling the boundary's progression in response to local flow dynamics.

All programs are written in MATLAB, with selected codes given in Appendix A. The progression used in developing the numerical approximations is as follows.

1. Develop both spatial and time discretizations for the fixed boundary case, and build the linear system. This is described below.

2. Use the resulting linear system to numerically solve (5.12). This is compared to the analytic solution (2.19) and used as a baseline case.

3. Implement the dynamic reporting grid method to apply the threshold pressure gradient, and compare these results to the analytic solution given in Fig. 4.2.

4. Modify the discretization to include the quadratic gradient term and apply to both the codes with or without the threshold pressure gradient.

Discretization of the Partial Differential Equations

The discretization below uses the dimensionless version of the flow equations given in (5.12) and (5.18), so that p will represent p_D, the dimensionless version of pressure, as with the other variables and parameters, and the full domain is between $x = 0$ and $x = 1$.

Spatial Discretization

Physically, the numerical models developed here represent fluid flow through connected pore spaces from high to low pressure. Although the flow is modeled as one-dimensional at the macroscopic scale, enabling the use of one-dimensional flow equations, it should reflect the three-dimensional nature of porous media, where fluid must traverse a complex network of interconnected pores. This suggests that spatial discretization be performed using a finite volume approach, a method widely employed in reservoir simulation [25].

The three-dimensional characteristics are evident in the model, which consists of cells labeled at their centers and boundary conditions assigned at the left edge of cell 1 and the right edge of cell N. The one-dimensional equation in this three-dimensional space has the average properties of the cells (porosity and permeability) and the resulting pressure is assigned to the center of each cell (Figure 6.1), such that $P_i^n \approx p(x_i, t_n)$ represents the average pressure in cell i at time t_n. The ith grid cell is defined as $\mathcal{C}_i = (x_{i-1/2}, x_{i+1/2})$. The cells, or cubes, have length Δx , where we fix $N \in \mathbb{Z}^+$ and let $\Delta x = 1/N$. We then define the grid points at the center of each cell as $x_i = \frac{(2i-1)\Delta x}{2}$ for $i = 1, \ldots N$ (Figure 6.2).

The fluid enters and exits each cell through the boundary between its neighboring cells. For cell i, its neighbors are cell $\mathcal{C}_{i-1}$ and cell $\mathcal{C}_{i+1}$, and the boundary between cell i and its neighbors are labeled $x_{i-1/2}$ and $x_{i+1/2}$. This provides a method for defining the locations of the boundary conditions, which are at $x_{1/2}$ and $x_{N+1/2}$, the left boundary of cell 1 and the right boundary of cell N. Following a finite volume approach [11, 22], the average pressure in a cell is defined as:

$$P_i(t) = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} p(x,t)\, dx \tag{6.1}$$

Taking the time derivative of both sides of (6.1) yields:

$$\frac{dP_i}{dt} = \frac{1}{\Delta x} \frac{d}{dt} \int_{x_{i-1/2}}^{x_{i+1/2}} p(x,t)\, dx \tag{6.2}$$

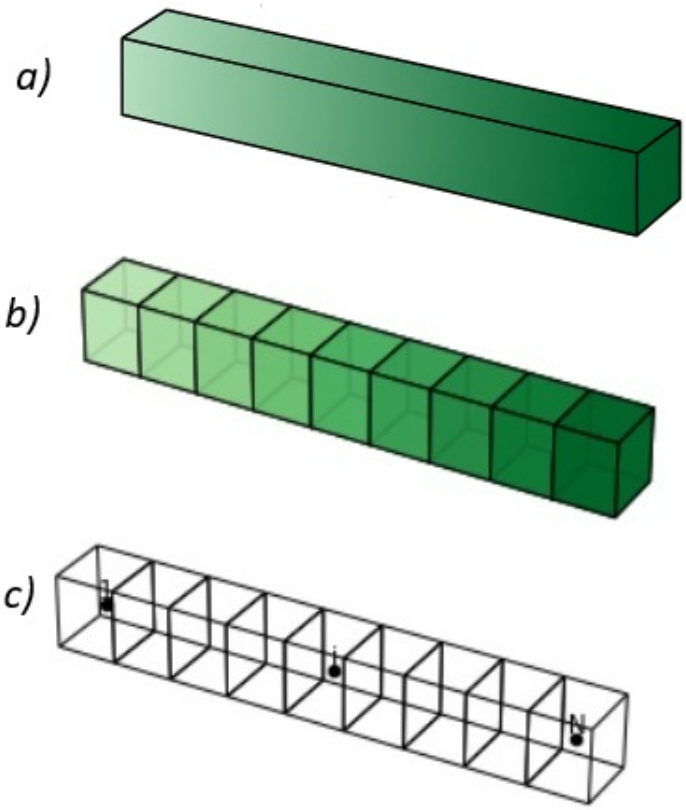

Figure 6.1: The progression from three-dimensional space with a continuous pressure gradient in the x direction (a) to discretized blocks (b) to a set of points located in the center of each gridblock, with each property of the block assigned to the center (c)

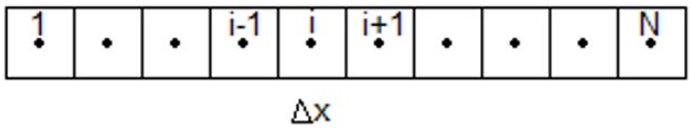

Figure 6.2: A two-dimensional rendering of the three-dimensional model with gridblocks labeled.

Next, Leibniz's rule is applied to interchange differentiation and integration:

$$\frac{dP_i}{dt} = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial p}{\partial t}(x,t)\, dx \tag{6.3}$$

By (5.12) $\frac{\partial p}{\partial t}$ is replaced by $\frac{\partial^2 p}{\partial x^2}$.

$$\frac{dP_i}{dt} = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 p}{\partial x^2}(x,t)\, dx \tag{6.4}$$

Applying the fundamental theorem of calculus to the integral on the right-hand side gives:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 p}{\partial x^2}\, dx = \left.\frac{\partial p}{\partial x}\right|_{x_{i+1/2}} - \left.\frac{\partial p}{\partial x}\right|_{x_{i-1/2}} \tag{6.5}$$

This results in a semi-discrete finite volume formulation [11], as the discretization is only spatial at this point.

$$\frac{dP_i}{dt} = \frac{1}{\Delta x} \left(\left.\frac{\partial p}{\partial x}\right|_{x_{i+1/2}} - \left.\frac{\partial p}{\partial x}\right|_{x_{i-1/2}} \right) \tag{6.6}$$

The derivative $\frac{\partial p}{\partial x}|_{x_{i+1/2}}$ is the pressure gradient at a cell face. To approximate its value, the central difference of the pressures at the centers of the neighboring cells is used, assuming a constant value of Δx.

$$\left.\frac{\partial p}{\partial x}\right|_{x_{i+1/2}} \approx \frac{P_{i+1} - P_i}{\Delta x} \tag{6.7}$$

$$\left.\frac{\partial p}{\partial x}\right|_{x_{i-1/2}} \approx \frac{P_i - P_{i-1}}{\Delta x} \tag{6.8}$$

Putting this together gives.

$$\frac{dP_i}{dt} = \frac{1}{\Delta x} \left(\frac{P_{i+1} - P_i}{\Delta x} - \frac{P_i - P_{i-1}}{\Delta x} \right) = \frac{1}{\Delta x} \left(\frac{P_{i+1} - 2P_i + P_{i-1}}{\Delta x} \right) \tag{6.9}$$

Time Discretization

The time discretization is defined by choosing a time step value Δt, and the resulting time grid is constructed as $t_n = n\Delta t$ for $n = 0, 1, 2, \ldots, n_f$. Time moves from $n = 0$, where the system is at initial pressure, to a final time $t_F = n_f \Delta t$. To discretize the time derivative at t_{n+1}, a backward Euler approach is used.

$$\frac{dP_i}{dt} \approx \frac{P_i^{n+1} - P_i^n}{\Delta t} \tag{6.10}$$

Combining the right sides of equations (6.9) and (6.10) produces the discretization of the PDE.

$$\frac{P_{i+1}^{n+1} - 2P_i^{n+1} + P_{i-1}^{n+1}}{\Delta x^2} \approx \frac{P_i^{n+1} - P_i^n}{\Delta t} \tag{6.11}$$

The resulting discretization yields the same algebraic system as a finite difference scheme using a central difference for the second spatial derivative and a backward (implicit) difference for the time derivative, as expected with a uniform cell size and a one-dimensional problem [33]. This implicit method, when applied to the parabolic equation in (5.12), is known to be unconditionally stable [32].

Boundary Conditions

In order to evaluate the boundary conditions, one has to treat the equation in (6.11) for the cases of $i = 0$ and $i = N$ separately. Since there is no cell that aligns with the left (or right) boundary, the boundary are located at $x = 1/2$ and $x = N + 1/2$ respectively. This requires a one-sided approximation of the second derivative.

A quadratic interpolant of the form ($P = ax^2 + bx + c$) is created through the boundary value (for the left side), $P(x_{1/2})$, the first interior value, $P(x_1)$, and the second interior value,$P(x_2)$ by describing the location of each in terms of x_1 and the length of a cell Δx. This allows the determination of a, b, and c, and then $2a$ is an estimate the second derivative.

For the left side of the model, that gives

$$\frac{P_2^{n+1} - 3P_1^{n+1} + 2P_{\text{left}}}{\frac{3}{4}\Delta x^2} \approx \frac{P_1^{n+1} - P_1^n}{\Delta t} \tag{6.12}$$

and a similar process for the right side gives:

$$\frac{2P_{\text{right}} - 3P_N^{n+1} + P_{N-1}^{n+1}}{\frac{3}{4}\Delta x^2} \approx \frac{P_N^{n+1} - P_N^n}{\Delta t} \tag{6.13}$$

Matrix Representation

For all cells, we have the system of equations:

$$\begin{aligned}
\frac{P_2^{n+1} - 3P_1^{n+1} + 2P_{\text{left}}}{\frac{3}{4}\Delta x^2} &= \frac{P_1^{n+1} - P_1^n}{\Delta t} && i = 1 \\
\frac{P_{i+1}^{n+1} - 2P_i^{n+1} + P_{i-1}^{n+1}}{\Delta x^2} &= \frac{P_i^{n+1} - P_i^n}{\Delta t} && i = 2, \ldots, N-1 \\
\frac{2P_{\text{right}} - 3P_N^{n+1} + P_{N-1}^{n+1}}{\frac{3}{4}\Delta x^2} &= \frac{P_N^{n+1} - P_N^n}{\Delta t} && i = N
\end{aligned}$$

By defining α as below, we can build a matrix system for the linear system, in this case shown as an example below, for a five-cell system. The resulting matrix is tridiagonal, symmetric, and has negative real eigenvalues. [30, 32] This system is algebraically equivalent to that of Section 2.8 of [32] with a slightly different matrix representation.

$$\alpha = \frac{\Delta x^2}{\Delta t} \tag{6.14}$$

$$\begin{bmatrix} -3-\frac{3}{4}\alpha & 1 & 0 & 0 & 0 \\ 1 & -2-\alpha & 1 & 0 & 0 \\ 0 & 1 & -2-\alpha & 1 & 0 \\ 0 & 0 & 1 & -2-\alpha & 1 \\ 0 & 0 & 0 & 1 & -3-\frac{3}{4}\alpha \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix}^{n+1} = \begin{bmatrix} -\frac{3}{4}\alpha P_1^n - 2P_{\text{left}} \\ -\alpha P_2^n \\ -\alpha P_3^n \\ -\alpha P_4^n \\ -\frac{3}{4}\alpha P_5^n - 2P_{\text{right}} \end{bmatrix} \tag{6.15}$$

Conventional Transient Flow Model

The models in this section approximate (5.12), with the fixed boundary conditions. They are used primarily as a baseline to compare with models that include a threshold pressure gradient or a quadratic gradient term.

Code 1: Baseline Case Without Moving Boundary Condition

For these codes, the linear systems in matrix form (represented by (6.15) for a 5-cell case) are solved by the built-in MATLAB solver (mldivide or backslash operator). The codes are designed so that one boundary condition for the pressure is set to match the initial condition, and the other is set to a lower value, allowing the flow to go either from right to left or left to right.

To verify that the linear system accurately represents the partial differential equation in (2.9), this code is run using the parameters from Katz given in Table 6.1. In this simulation, flow proceeds from right to left. The results are compared to the analytic solution provided by Katz [16], focusing on the early transient period—before the pressure disturbance from the lowered boundary condition propagates across the entire domain of 100 feet. The runtime is set to a final time corresponding to six minutes, consistent with [16], to ensure the system remains within this early transient regime. To investigate convergence given a

Table 6.1: The model parameter values used in the test cases of the code, first with the values from [16] and the second with field-scale values for a low permeability formation.

Parameter Information and Values			
		Case 1	**Case 2**
Name	**unit**	**value**	**value**
initial pressure	psi	500	1500
left boundary pressure	psi	300	1000
right boundary pressure	psi	500	1500
total compressibility	1/psi	0.00061	0.0000005
porosity		0.12	0.13
viscosity	cP	0.2	2.3
permeability	md	15.6	0.2

small amount of time to work with (6 minutes), the length of the timestep (Δt) is first set to correspond to 6 minutes, and then reduced, while additional runs also involved decreasing Δx. The results are given in Table 6.2. In this table, Δt and Δx are given in the problem units, while α is as given in (6.14) The two error measurements are the maximum error and the L2 norm, given below. Figures 6.3, 6.4 and 6.5 illustrate Runs 1, 3 and 5.

$$\text{Max Error} = \max_i |p_{\text{exact},i} - p_{\text{numerical},i}| \tag{6.16}$$

$$\text{L2 Norm} = \left(\sum_{i=1}^{N} \left(p_{\text{exact},i} - p_{\text{numerical},i}\right)^2 \right)^{1/2} \tag{6.17}$$

The second run of the code used typical field scale numbers given in Table 6.1, but a lower permeability was chosen. The code displays both the final pressure gradient across the model and the pressure gradients at various timesteps throughout the simulation (Fig. 6.6). It

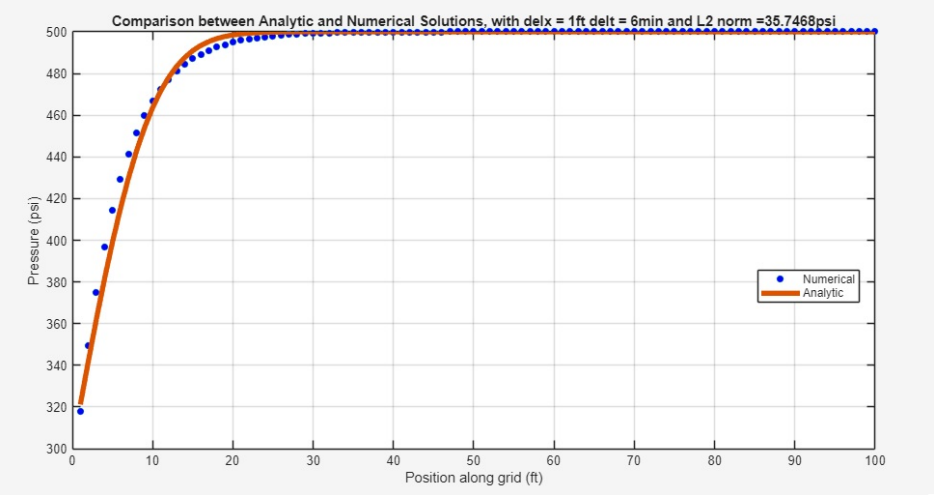

Figure 6.3: Results of Run 1, showing the poorest match with the largest of the tested timesteps and cell lengths.

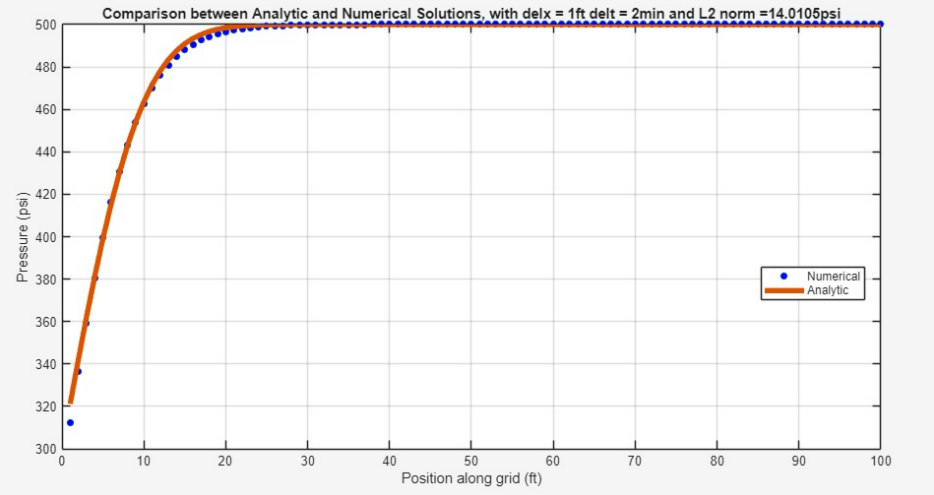

Figure 6.4: Results of Run 3, showing a better match when the timestep was shortened.

Run	Δt (min)	Δx (ft)	$\alpha = \frac{(\Delta x_D)^2}{\Delta t_D}$	Max error (psi)	L2 norm (psi)
1	6.0	1.0	0.036	15.23	35.75
2	3.0	1.0	0.071	7.98	15.67
3	2.0	1.0	0.11	9.03	14.01
4	2.0	0.5	0.027	4.79	17.59
5	1.0	0.33	0.024	3.31	10.62

Table 6.2: Convergence results for varying Δt and Δx

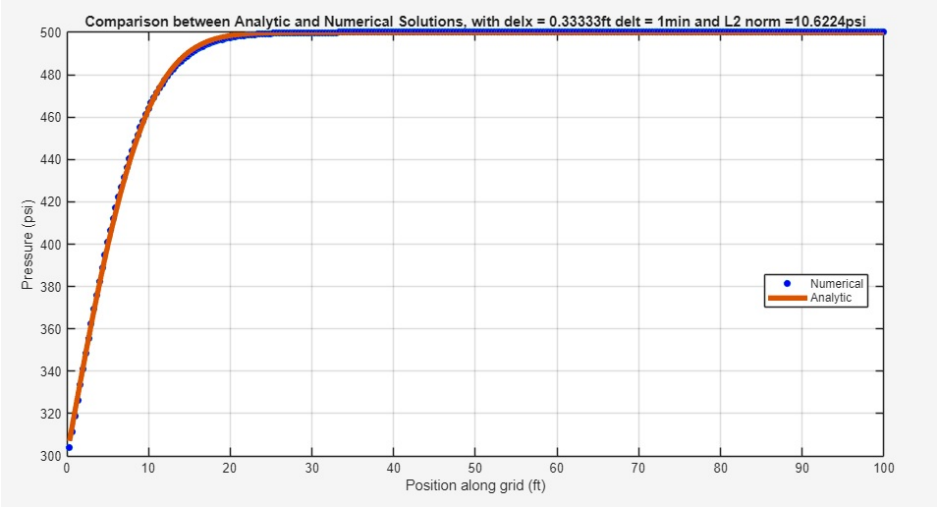

Figure 6.5: Results of Run 5, showing the strongest match of the five runs.

is the baseline code when comparing the results of the later codes with the moving boundary condition.

The Dynamic Reporting Grid Method for the Unconventional Flow Model with a Threshold Pressure Gradient

The dynamic reporting grid method developed here models fluid flow in low-permeability media by expanding the domain based on meeting the threshold pressure gradient criteria.

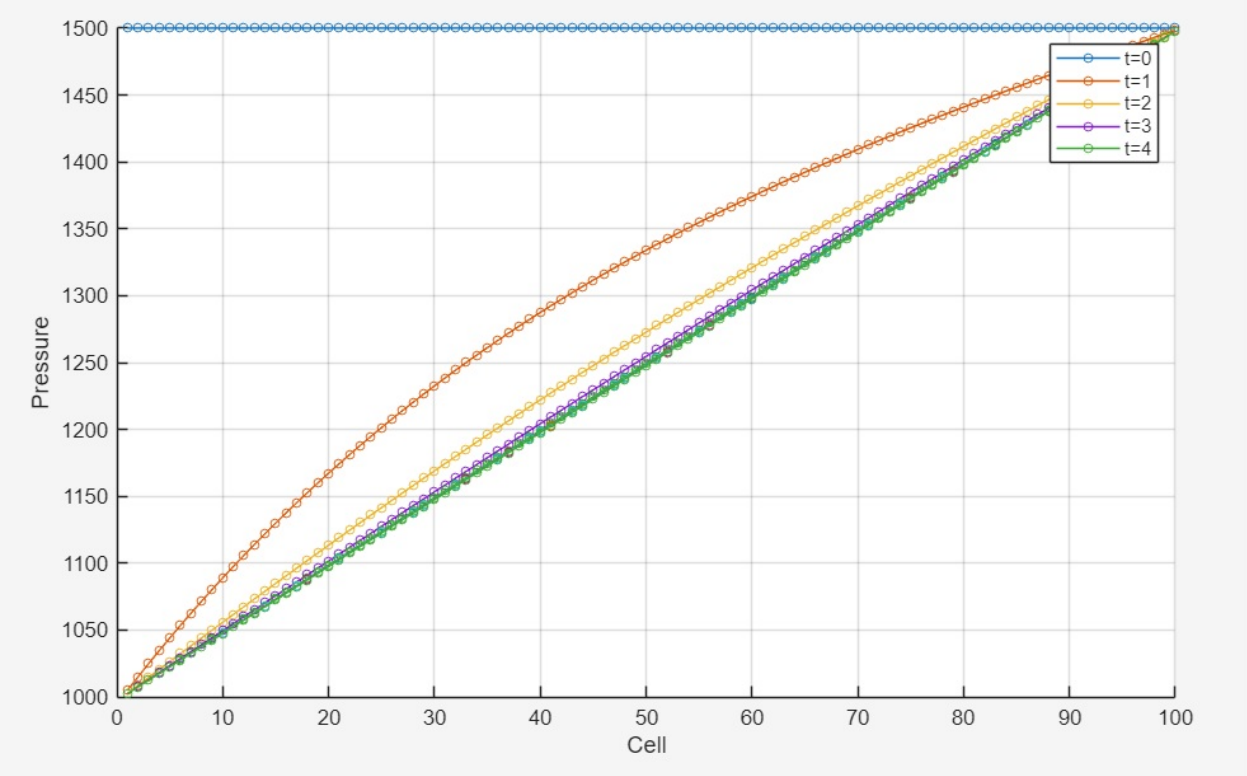

Figure 6.6: Flow in this example is right to left. The results at various timesteps are shown, with the straight line at t = 4 indicating that the problem has reached steady-state. These results correspond to the parameter values listed in Table 6.1. The total length of the domain is 1000 ft, with $\Delta x = 10$ ft, and the total simulation time is 365 days, with each timestep advancing the model by one month (30.4 days).

Unlike a fixed-domain approach, the PDE model uses a moving boundary formulation as in (5.18), (5.19), (5.20), and (5.21). The simulation method introduces a reporting grid that activates only when the pressure gradient exceeds the threshold pressure gradient in a working grid, allowing the interface to evolve by expanding the working grid. This design enables direct simulation without prescribing a location of the moving boundary as in the analytic solution given by Liu et al. [28]. Codes running this method will have the flow going from right to left.

The process, illustrated in Fig. 6.7 is:

- Set the desired model size with N gridblocks of length Δx and initialize each cell to the initial pressure (Reporting Grid). Note: P_i is used here instead of P_{init} to match with Figure 6.7).
- Create a second model with $2N$ gridblocks of length $\Delta x/2$ also initialized to initial pressure (Working grid).
- Using only the rightmost two cells in the working grid, set boundary conditions: $P_{x=0} = P_i$ on the left and $P_x = L$ to a lower pressure to initiate flow from left to right, then solve the system $A\mathbf{p} = \mathbf{d}$. to get $\mathbf{p} = [p_1^1, p_2^1]$, shown in Figure 6.7 as p_1 and p_2.
- Evaluate the pressure gradient between these two adjacent cells to determine if it exceeds the threshold pressure gradient G.
- If the gradient exceeds G, the cell is considered "reached" and two new adjacent cells are added to the working grid - expanding the model.
- If the gradient does not exceed G, advance the timestep on the two-cell model, solve the system and check again.

- Pressure continues to be updated cumulatively over all timesteps; new cells are tagged as reached when local gradients exceed G and remain active for the rest of the simulation.
- In the reporting grid, average pressures of each pair of cells in the working grid are used to update the corresponding reporting grid values, but only for cells where the TPG has been exceeded.
- Unreached cells retain their initial pressure P_i in the reporting grid, indicating no flow is moving in those cells. Continue until the final time is reached.

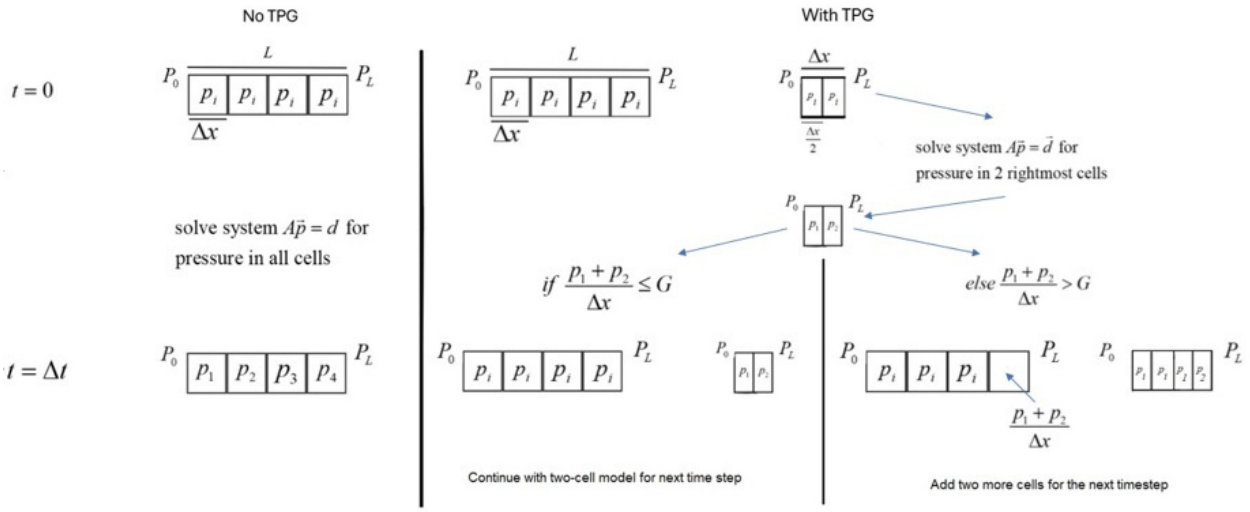

Figure 6.7: The first time iteration of a model, without the TPG applied and applying the TPG using the Dynamic Reporting Grid method

Code 2: Moving Boundary Condition Modeled with the Dynamic Reporting Grid Method

Code 2 uses the process described above and in Fig. 6.7 to add this implementation of the moving boundary to Code 1. This involved not only expanding the size of the spatial

domain over time but also dynamically updating the pressures in the reporting grid, and ensuring that the code did not exceed its total allotted time. In the flow equations (5.12) and (5.18), and the figures 6.7 and 4.1, the flow is in the conventional left-to-right direction. In the numerical simulations, the flow is reversed to right-to-left. This is due to the complexity of indexing left-to-right flow with the DRG method.

Example results of this code are presented for two cases, one on a field scale using a combination of synthetic data and values from Prada et al. [38], and one on a lab scale. Parameters for these example simulations are given in Table 6.3, and the graphical results are shown in Figs. 6.8 and 6.9, along with the results of Code 1 using the same parameters.

These graphs illustrate the expected behavior associated with a threshold pressure gradient. In models incorporating the TPG, a region on the right side maintains the initial pressure, indicating that the fluid remains immobile because the pressure gradient does not overcome the threshold value G. However, when the TPG has been achieved in a particular cell, the pressure in the effective cells does not remain constant. The pressure in those cells continues to decrease, resulting in the behavior in the affected cells approaching that of the pressure for the original model without the TPG, a model that exhibits a continuous decline in pressure across all cells.

The Prada Data Set The data set developed by Prada et al. [38] is a set of data from laboratory measurements, which include measurements of the TPG. This data set provides values for the lab scale model and some values for the field scale model. It is also used by researchers taking a more traditional approach to modeling the moving boundary condition [50]. In the experimental work, Prada [38] worked with seven core samples in three categories: sandstone, sandpack, and shaly sandstone. The first two categories, sandstone and sandpack, are not unconventional/low permeability based on the data provided, while the shaly sand core has a mobility value (mobility is defined as k/μ) of 56.6 md/cP, the value that is closest

Table 6.3: The model parameter values for an initial field scale and lab scale simulation with a threshold pressure gradient. The value of G and selected lab scale parameters are taken from [38]. Prada's data set reports the mobility, k/μ, so in both cases, the value of viscosity was selected to be 1 to isolate the permeability.

Parameter Information and Values			
		Field Scale	**Lab Scale**
		L=1000 ft	L=0.208 ft
		total time = 1 year	total time = 1 min.
Name	**unit**	**value**	**value**
initial pressure	psi	2500	12.8
left boundary pressure	psi	100	0
right boundary pressure	psi	2500	12.8
total compressibility	$\frac{1}{\text{psi}}$	0.000061	0.000061
porosity		0.12	0.18
viscosity	cP	1	1
permeability	md	56.6	56.6
threshold pressure gradient (G)	psi/ft	24.78	24.78

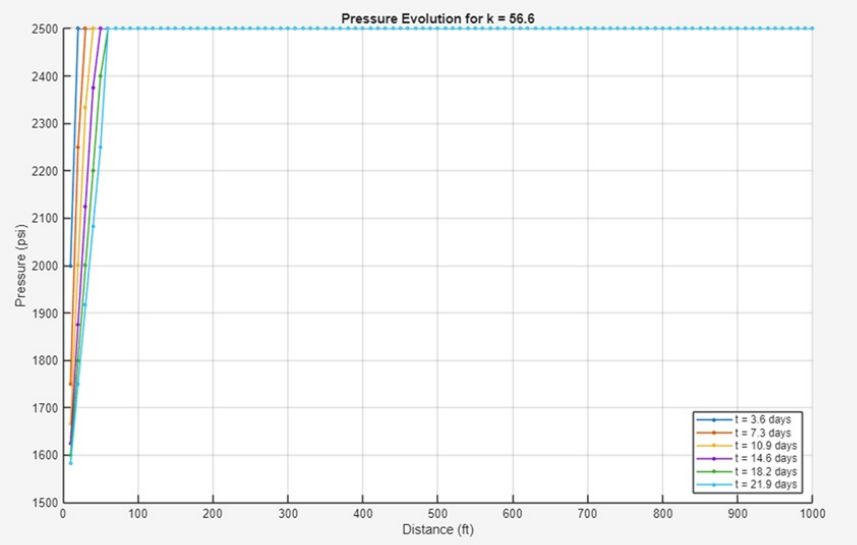

Figure 6.8: Results of the field scale test case for the first 6 timesteps, with a total length of 1000 ft in 100 cells with $\Delta x = 10$ ft. The movement along the top line indicates the increasing number of cells reached at each time step until the final time.

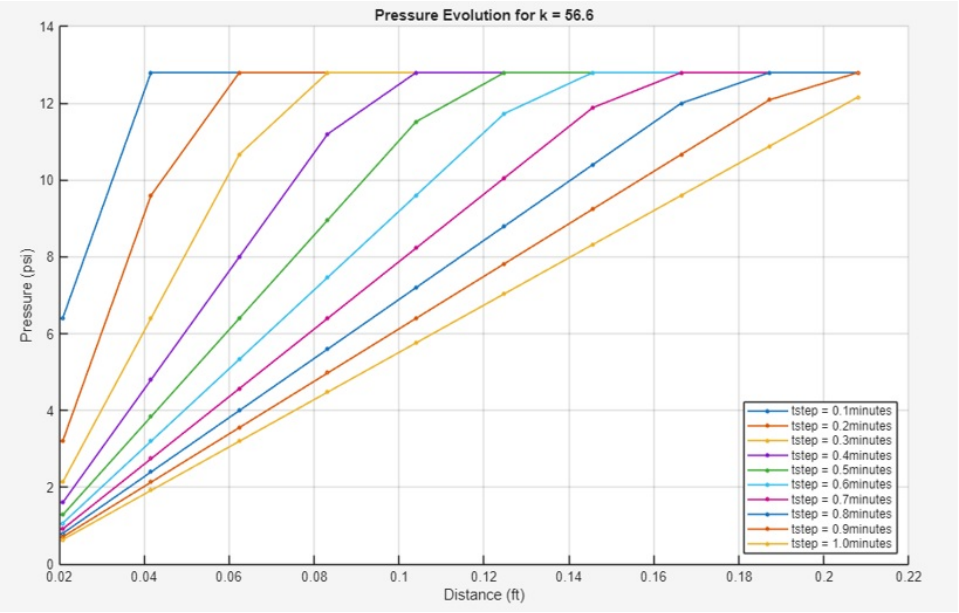

Figure 6.9: Results for the lab scale model at a total time of 1 minute with $\Delta t = 0.05$ minutes, and a total length of 0.208 ft in 10 cells.

to unconventional. As the data only give the mobility, when using the data for simulation, the viscosity will be set to one, and the mobility value will be used for permeability.

Tables 6.10, 6.11, and 6.12 give the data for the seven cores from Prada et al. [38]. The fourth table 6.13 contains an extension of that work from Yao, et al. [50] where they include the average flow velocity (here v is v_w) and dimensionless TPG. Their notation and formula for calculating the dimensionless TPG is given in equation (6.18).

$$\lambda_D = \frac{k}{\mu} \frac{\lambda}{v_w} \tag{6.18}$$

The first table, Figure 6.10 gives the porosity for the seven cores along with the other parameters used in its measurement. The second table in Figure 6.12 shows the results of an experiment where the flow rate across the core is set and the resulting pressure differential (here $\Delta p = \frac{p_0 - p_L}{L}$) is measured, or the pressure differential is set and the flow rate is measured. The third table in Figure 6.12 is the TPG and mobility for each core, with the TPG denoted here as $\frac{\Delta p}{\Delta L}_{cr}$.

Properties of the porous materials used in the flow experiments

Sample	Length (cm)	Cross-sectional area (cm^2)	Dry weight (g)	Saturated weight (g)	Porosity (fraction)	Pore volume (cc)
Brown sandstone #1	30.7	20.2	1178	1354	0.284	176
Brown sandstone #2	30.7	20.2	1148	1317	0.273	169
Brown sandstone #3	30.7	20.2	1215	1373	0.255	158
Brown sandstone #4	30.7	20.2	1190	1355	0.266	165
Sandpack (20–40 mesh)	55.48	11.4	N.A.	N.A	0.356	225
Sandpack (20–40 mesh)	55.48	11.4	N.A.	N.A	0.334	211
Sandpack (20–40 mesh)	55.48	11.4	N.A.	N.A	0.324	205
Shaly sandstone	6.35	11.4	190.83	203.87	0.18	13.04

Figure 6.10: Porosity and length of the seven cores (reproduced from [38])

A Fixed Domain Approach to Numerically Modeling the Threshold Pressure Gradient Problem Yao et al. [50] developed a numerical solution of the first equation in (5.18) with the moving boundary conditions by using a spatial coordinate transformation to create

Measured pressure difference vs. flow rate data

Brown-sandstone

Brown sandstone #1		Brown sandstone #2		Brown sandstone #3		Brown sandstone #4	
Δp (psi)	q (cc/min)	Δp (psi)	q (cc/min)	Δp (psi)	q (cc/min)	Δp (psi)	q (cc/min)
0.80	8.83	1.50	13.19	0.80	1.71	1.20	5.79
1.00	12.35	2.00	20.64	1.00	5.66	2.00	11.78
1.20	18.15	2.50	27.85	1.35	8.95	2.50	17.47
2.00	30.00	3.00	32.32	2.00	17.50	3.00	20.46

Sand packs

Sandpack #1		Sandpack #2		Sandpack #3	
Δp (psi)	q (cc/min)	Δp (psi)	q (cc/min)	Δp (psi)	q (cc/min)
0.50	3.81	0.60	2.47	0.50	4.57
0.80	13.34	1.00	12.34	1.00	11.59
1.20	25.25	1.20	15.12	1.40	21.91
1.80	34.30	1.50	20.34	1.80	29.92

Shaly sandstone

Δp (psi)	q (cc/min)
9.89	2
14.95	4
19.52	6

Figure 6.11: Pressure differential and associated flow rates for the seven cores (reproduced from [38])

Threshold pressure gradient vs. fluid mobility

Sample	$(\Delta p/\Delta L)_{cr}$ (psi/cm)	K/μ (md/cp)
Brown sandstone #1	8.46E-03	6.52E+03
Brown sandstone #2	1.40E-02	4.81E+03
Brown sandstone #3	2.03E-02	4.73E+03
Brown sandstone #4	1.70E-02	3.13E+03
Sandpack #1	4.60E-03	2.79E+04
Sandpack #2	7.89E-03	2.35E+04
Sandpack #3	5.84E-03	2.38E+04
Shaly sandstone	8.13E-01	5.66E+01

Figure 6.12: Threshold pressure gradient and mobility (reproduced from [38])

Sample	λ (psi/cm)	k/μ (md/cp)	Minimum flow rate q_1 (cm^3/min)	Maximum flow rate q_2 (cm^3/min)	Cross-sectional area A (cm^2)	Average flow velocity $v_w = (q_1 + q_2)/A/2$ (cm/min)	$\lambda_D = \lambda k/\mu/v_w$
Brown sandstone #1	0.00846	6520	8.83	30	20.2	0.961139	0.242
Brown sandstone #2	0.014	4810	13.19	32.32	20.2	1.126485	0.252
Brown sandstone #3	0.0203	4730	1.71	17.5	20.2	0.475495	0.852
Brown sandstone #4	0.017	3130	5.79	20.46	20.2	0.649752	0.345
Sandpack #1	0.0046	27900	3.81	34.3	11.4	1.671491	0.324
Sandpack #2	0.00789	23500	2.47	20.34	11.4	1.000439	0.782
Sandpack #3	0.00548	23800	4.57	29.92	11.4	1.512719	0.364
Shaly sandstone	0.813	56.6	2	6	11.4	0.350877	0.553

Figure 6.13: An extension of Prada's data to include average flow velocity and dimensionless TPG. (reproduced from [50])

a mathematical model with fixed boundary conditions. They verified their solution by comparing it to the solution given by Liu et al. [28] discussed in Section 4 while using the Prada et al. [38] data set, making their results useful to compare to the Code 3 results. Figure 6.14 gives the comparison of this numerical approach with Liu's analytic solution, showing a good match. Figure 6.15 shows the analytic results of dimensionless pressure (here P_D) plotted against dimensionless position x_D for three values of the dimensionless threshold pressure gradient, with $\lambda_D = 0.553$ (highlighted on the graph) corresponding to the shaly sandstone core. A version of Code 2 was developed to compare to the solution in Figure 6.15 for the shaly sand core (with dimensionless TPG of 0.553). There were a number of assumptions made for this work in cases where the data were not available in [28].

- There was no indication of the scale of the model. An assumption was made to use the length of the sandstone core.
- There was no initial or boundary pressure given. Boundary and initial pressures were set to 1000 psi and 0 psi, respectively, so that the resulting dimensionless pressures matched those values on the graph.
- The solution is plotted against dimensionless position, but as defined, that should range from 0 to 1. Instead, to match the x-axis, cell numbers were used.

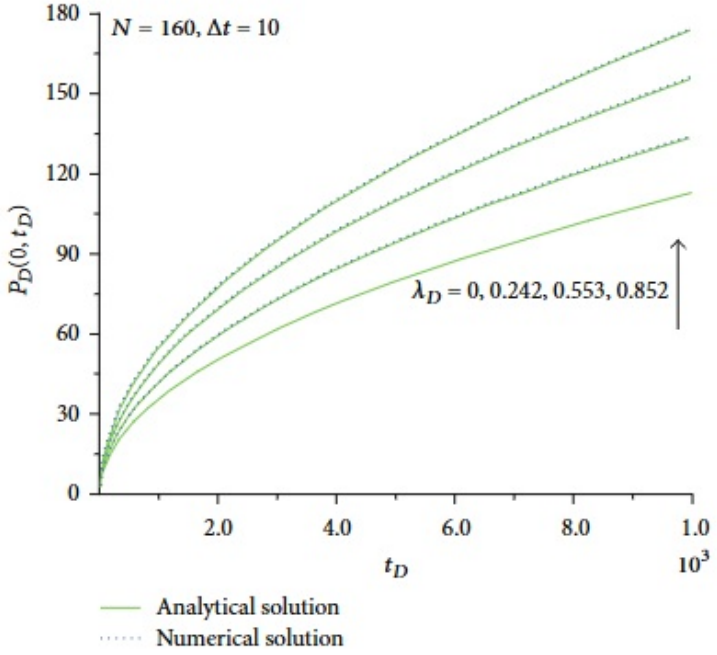

Figure 6.14: Comparison of Liu et al. analytic solution (solid lines) with Yao et al. numerical solution, where p_D is dimensionless pressure (also referred to as U), t_D is dimensionless time, and increasing values of dimensionless pressure gradient are represented by the different plots. (reproduced from [28]).

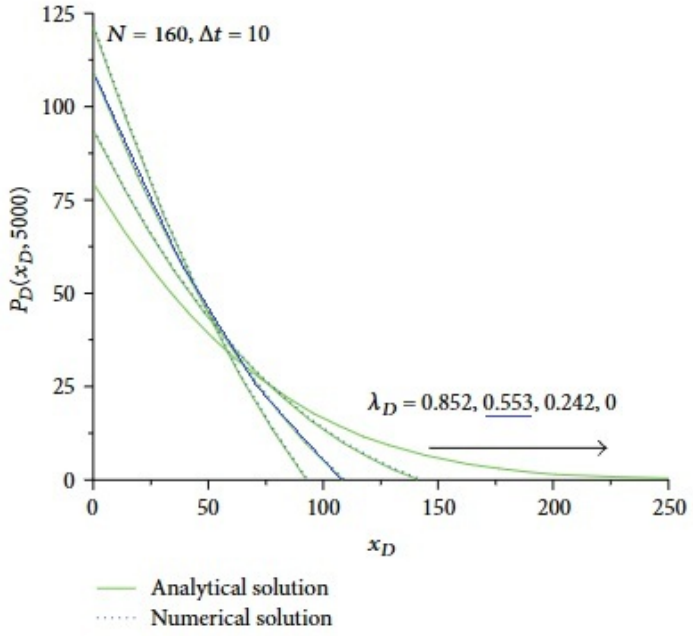

Figure 6.15: Numerical solutions at different dimensionless TPG values reproduced from [28]. The curve highlighted in blue uses input parameters matched in the simulation in Figure 6.16

.

- No compressibility values were given in Prada's data set, likely due to their working with an incompressible fluid. The same value used in other codes ($c_t = 0.0000611/psi$) was used.

Table 6.4: Parameters used in running Code 2 to compare with the analytic solution in Liu et al. [28]

Parameters used		
Comparison with Liu's Analytic Solution [28]		
initial pressure	1000	psi
left boundary pressure	1000	psi
right boundary pressure	0	psi
total compressibility	0.000061	1/psi
porosity	0.18	
viscosity	1	cP
permeability	56.6	md
threshold pressure gradient	24.78	psi/ft

The results of this simulation are given in Figure 6.16. The visual comparison with Figure 6.15 is encouraging, but the number of assumptions prevents a rigorous comparison.

Code 3 and 4: Including the Quadratic Gradient Term

Codes 3 and 4 extend Codes 1 and 2 by retaining the quadratic pressure gradient term into the partial differential equation defined in (5.12) and (5.18). These extensions are designed to explore the potential significance of the quadratic term in low-permeability regimes, as suggested by [26, 27], and to demonstrate how such a term can be integrated within the dynamic reporting grid framework.

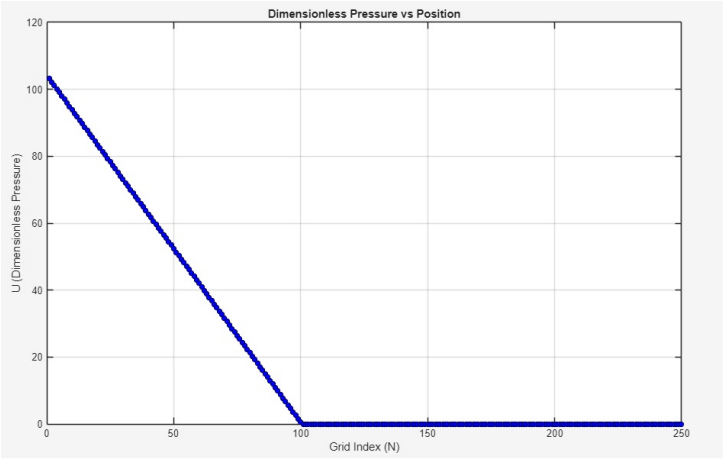

Figure 6.16: Dimensionless pressure at $\lambda_D = 0.553$ using the values in Table 6.4

In Codes 3 and 4, the quadratic pressure gradient term is treated as a nonlinear source term and added to the right-hand side vector d, preserving the structure of the matrix A. This avoids the need to linearize the term or modify the matrix coefficients, consistent with the iterative solution strategy described by Patankar [36].

The pressure gradient is approximated using a central finite difference for the fixed boundary condition (6.19) and a backwards finite difference for the dynamic reporting grid method with the moving boundary condition (6.20) (to avoid using a cell in the estimation that hasn't been reached), squared, and multiplied by the oil compressibility, then added to the existing term $d_i^n = -\alpha P_i^n$. In Code 3, this is done for $i = 2, \ldots, N-1$ and in Code 4 it is done for $i = 2, \ldots, N$.

$$d_i^n = -\alpha P_i^n + c_o \left(\frac{P_{i+1}^n - P_{i-1}^n}{2\Delta x} \right)^2 \tag{6.19}$$

$$d_i^n = -\alpha P_i^n + c_o \left(\frac{P_i^n - P_{i-1}^n}{\Delta x} \right)^2 \tag{6.20}$$

It is important to have a good value for oil compressibility c_o because the assumption that the quadratic gradient term is negligible is based on a small c_o and a small pressure

gradient. The value of c_ϕ, the formation compressibility, and their sum, c_t, the total compressibility, have to be considered as well, as c_t appears in the right-hand side of the equation with or without the quadratic gradient term.

In conventional modeling, a rule of thumb is that c_t can be assumed to be on the order of 10^{-6} psi^{-1}. However, Lan et al. [19] suggest that unconventional formations can have a formation compressibility in the range of 1×10^{-6} to 1×10^{-4} psi^{-1}, and Petrosky and Farshad's correlation [37] place the oil compressibility in the range of 2.5×10^{-5} to 3.5×10^{-5} psi^{-1}. Values in these ranges are used with Codes 3 and 4.

Using Code 3 with the parameters given in Table 6.5 generates Figure 6.19. In this simulation, the maximum difference in pressures between the two cases was 0.20735 psi. This small difference is consistently small throughout the problem domain, and is visualized in Figure 6.18 . This metric is used in further analysis in Chapter 7.

Table 6.5: Parameters used in example run of Codes 3 and 4

Parameters for Code 3 and 4: L = 100 ft, total time = 365		
initial pressure	2500	psi
left boundary pressure	100	psi
right boundary pressure	2500	psi
total compressibility	0.00008	1/psi
porosity	0.10	
viscosity	2	cP
permeability	2	md
oil compressibility	0.00003	1/psi

Code 4 also uses a right-hand side construction to incorporate the quadratic gradient

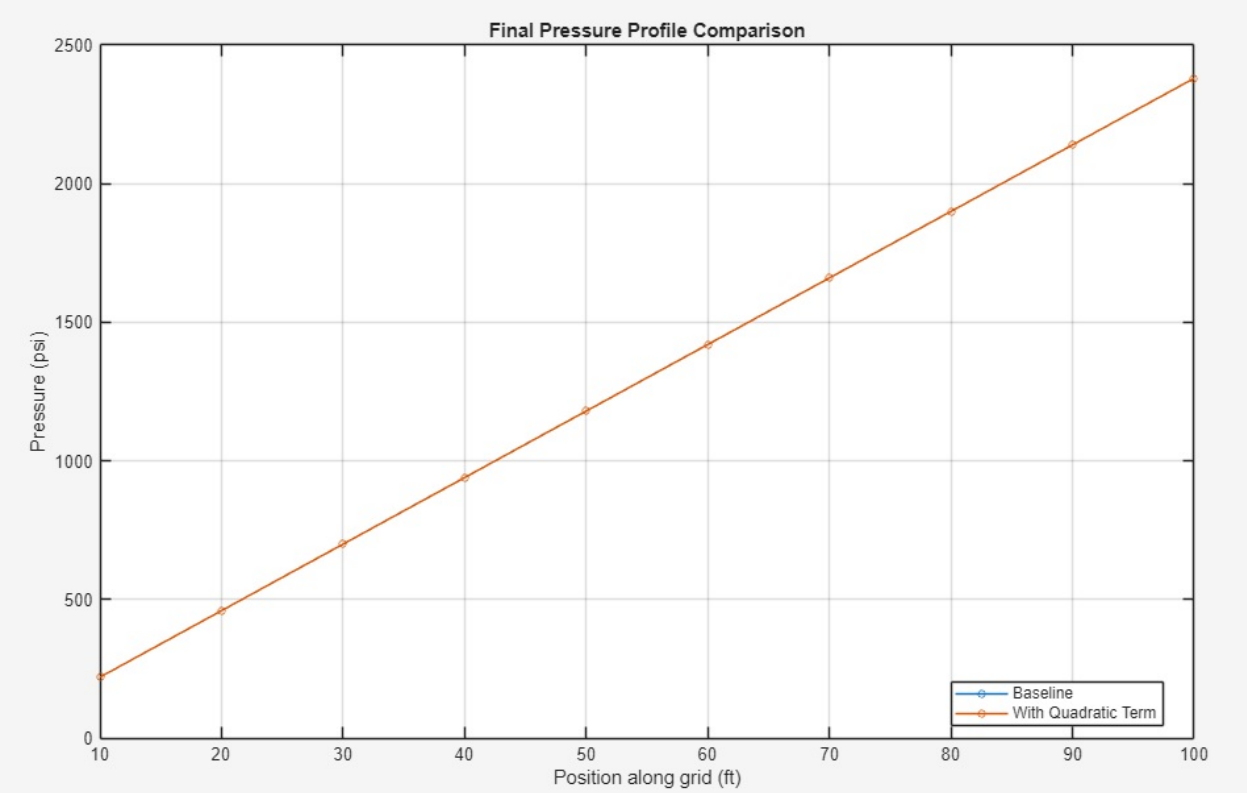

Figure 6.17: Comparison of results when the data in Table 6.5 are used in Code 3 with and without the inclusion of the quadratic gradient term

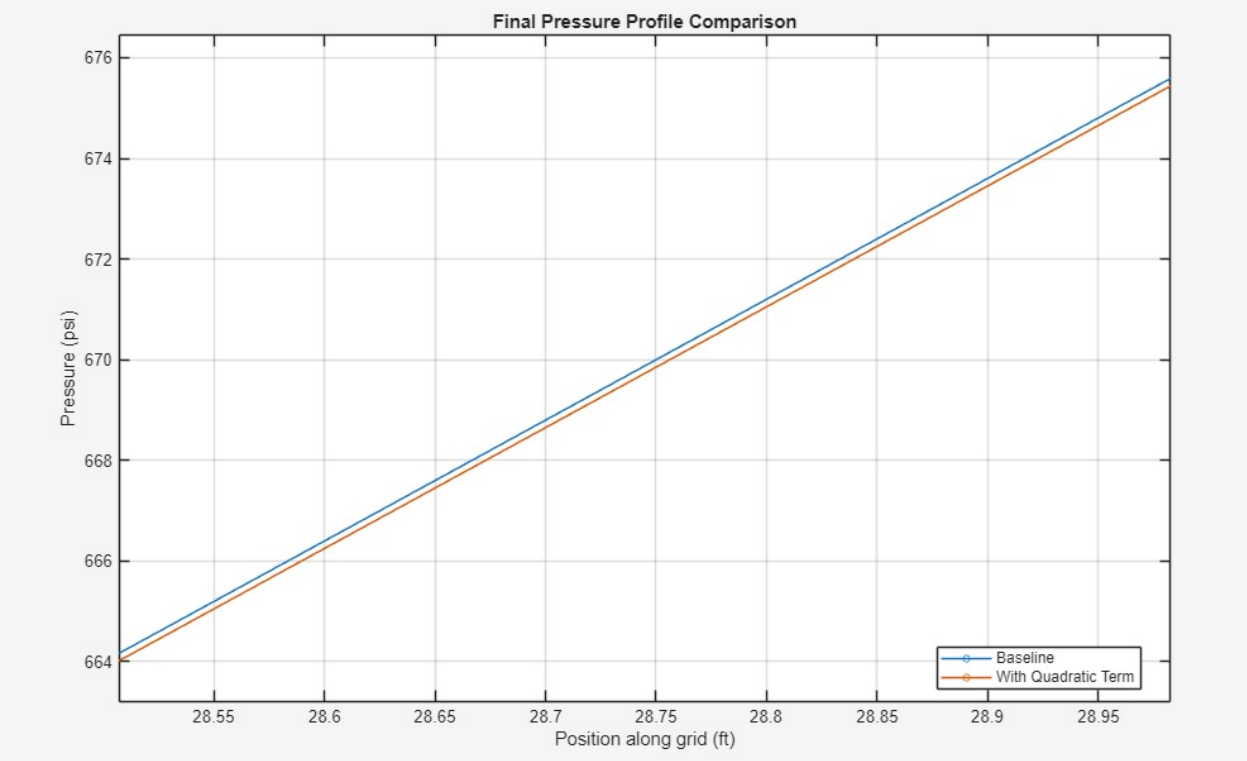

Figure 6.18: A close-up view of the Code 3 results showing the separation of the two models.

term, with a backward difference to discretize the pressure gradient (6.20), but it is then placed in the framework of the dynamic reporting grid (Figure 6.19). The maximum pressure difference in this case was 5.9023 psi, and it occurred at the second grid block, corresponding to 20 ft, as shown in Figure 6.20. However, close to the moving boundary, the effect of the quadratic gradient term was very small (Figure 6.21). This behavior is not surprising because the pressure gradient is very small near the moving boundary and larger near the fixed boundary. A similar pattern appears in analytic solutions of moving boundary problems with nonlinear terms where the lower bound solution of the partial differential equation with and without the nonlinear term closely matches near the moving boundary, indicating that the nonlinear contribution is negligible in that region [17].

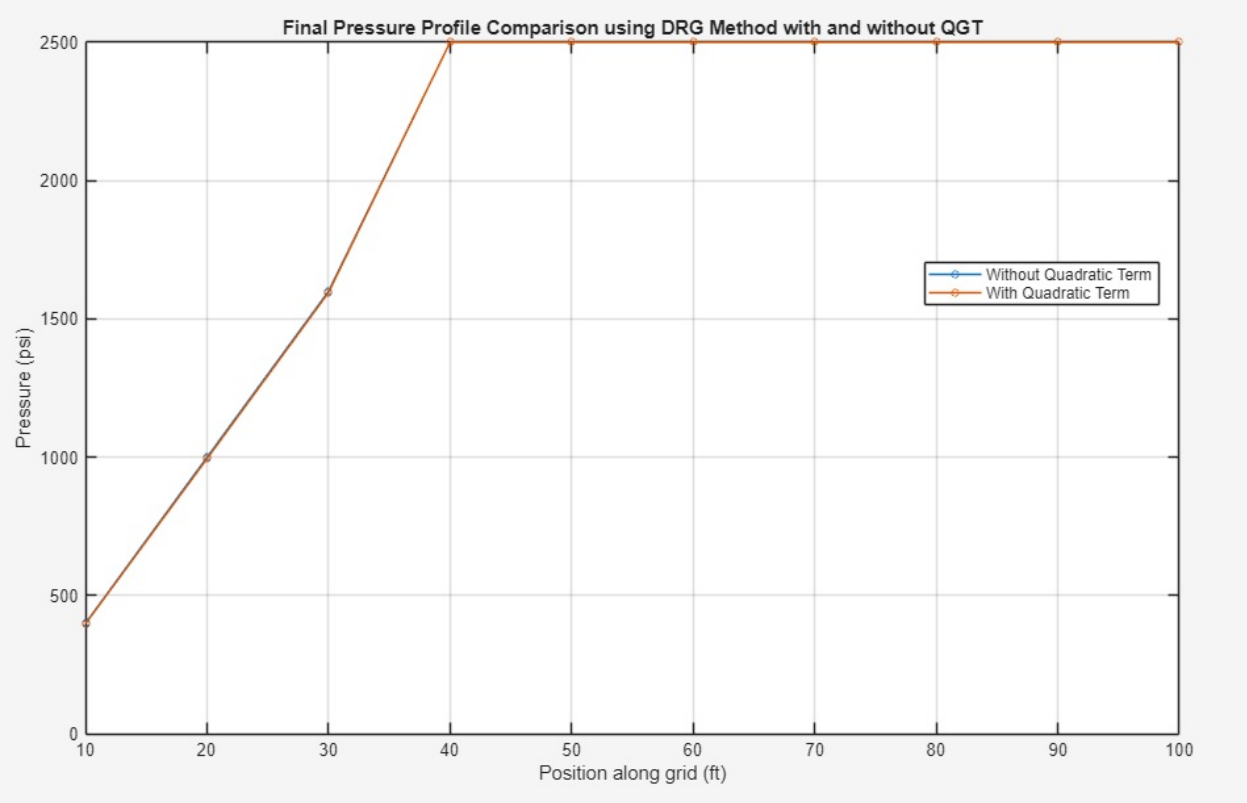

Figure 6.19: Full scale comparison of results when the data in Table 6.5 are used in Code 4 with and without the inclusion of the quadratic gradient term.

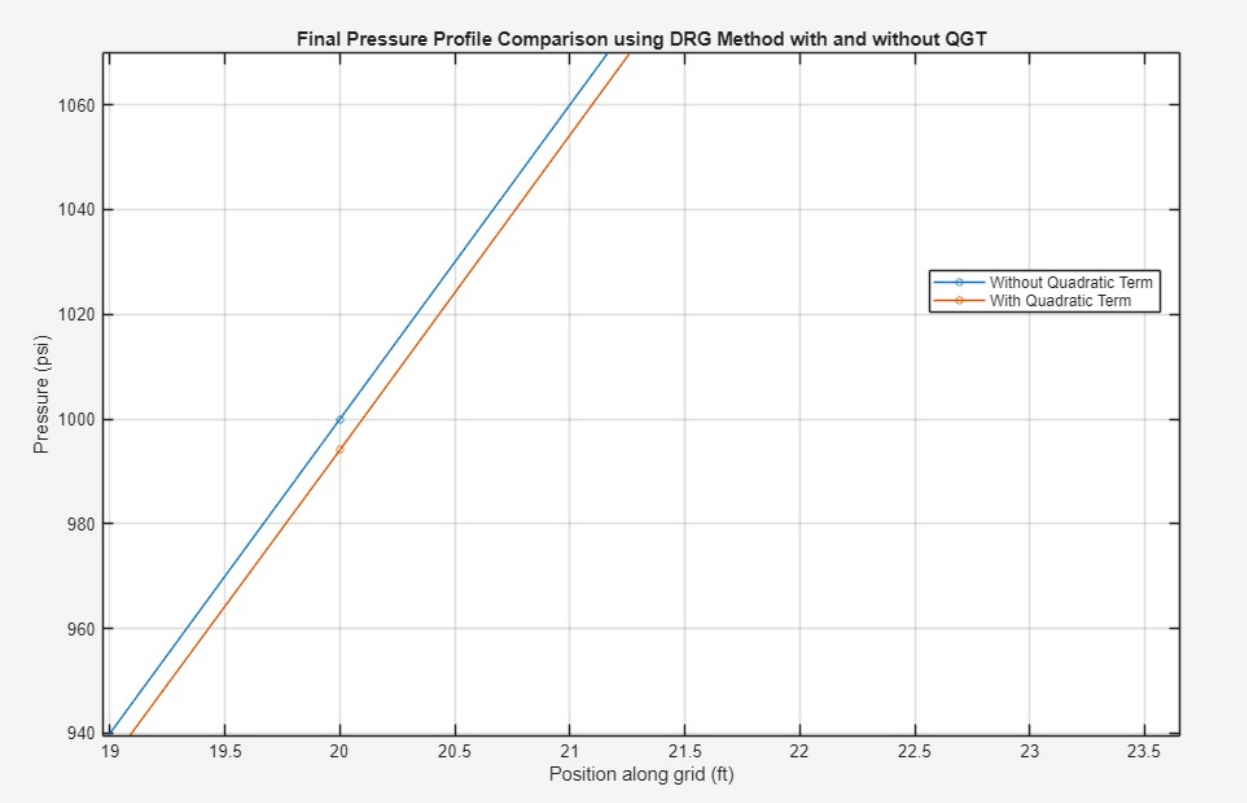

Figure 6.20: A closer view of the location of the maximum pressure difference.

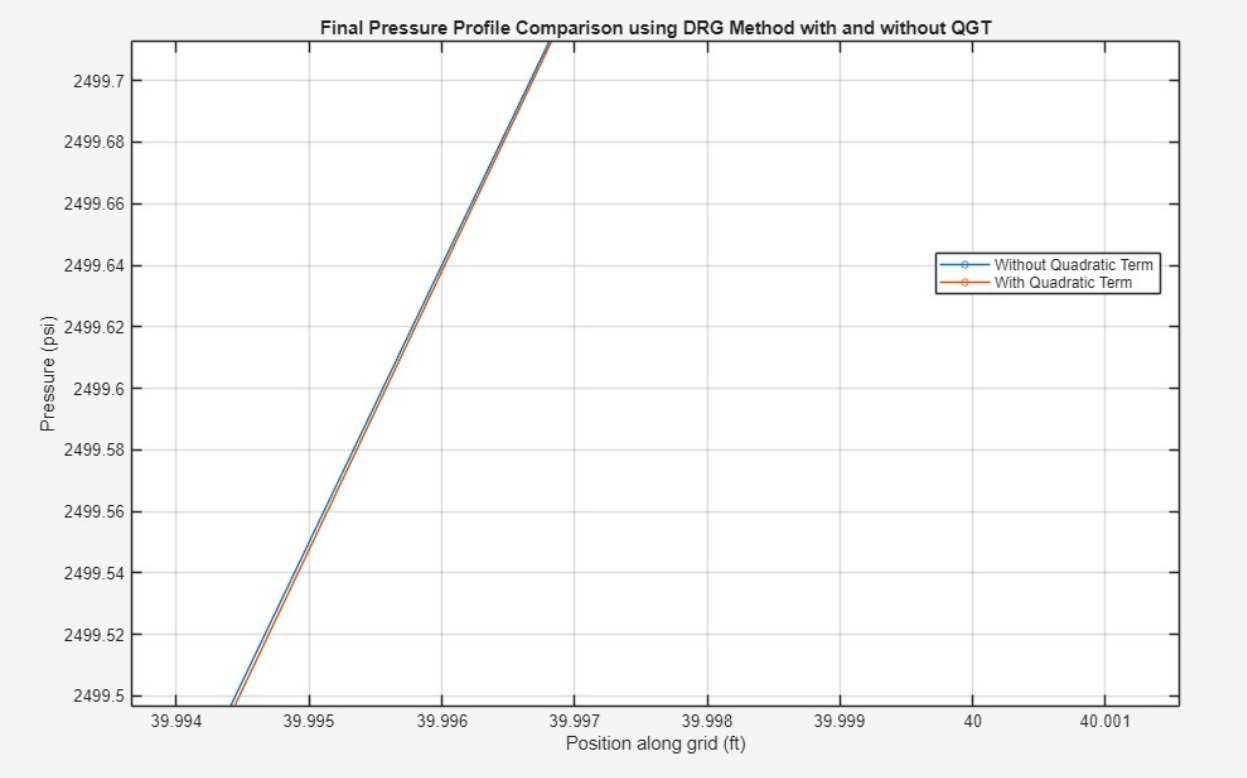

Figure 6.21: A closer view of the small difference between the two cases near the location of the moving boundary (at 40 ft) when the simulation reached the end of the allotted time.

Dynamic Reporting Grid Method Discussion

The dynamic reporting grid method is designed to be added around any one-dimensional flow model, as it utilizes the original matrix system and applies the moving boundary using a purely numerical technique. This can provide flexibility for adding non-linear terms, less complexity when working with multiphase flow and compressible fluids, and may be applicable to other types of problems where threshold behavoir is observed.

This method solves the same type of matrix as the conventional code, but starts by solving a small system (two cells, resulting in a 2×2 matrix) until the TPG is reached. Then the code brings in two more cells, and each time the TPG is reached continues to add two cells, until the total time is reached or the full working model ($2N$ cells) has reached steady state. In contrast, the conventional code for the same model, length, and total time is an N cell model and solves an ($N \times N$s) system until total time or steady state is reached.

Computing Resources Comparison

In general, the Dynamic Reporting Grid approach may require less computing time compared to codes that solve the fixed boundary condition problem. The latter resolves the full $N \times N$ system for every timestep, whereas the working model begins with smaller matrix systems and only escalates to solving a $2N \times 2N$ matrix once the threshold pressure gradient is met across the entire domain. When applied to low-permeability problems as intended, it is unlikely to reach that stage.

To get a rough estimate of the computing time, MATLAB's `tic/toc` commands were added to Code 1 and Code 2, to measure the elapsed time when parameters were varied. In addition, for this analysis, external features such as graphs or varying timesteps, were removed to focus on the differences in the way the pressures were computed. MATLAB's `tic/toc` timing commands can yield variable results across repeated runs of the same code,

so three runs of the same simulations were done and the average elapsed time was used to determine if Code 1 or Code 2 consistently ran faster.

Table 6.6 presents results for simulations with varying permeability values. The domain length was set to $L = 1,000$ ft, discretized into $N = 100$ grid points. The total simulation time was 365 days, with a time step $\Delta t = 3.65$ days. The varying of permeability here had little effect, in each case Code 1 used about a third of the computing time as Code 2. Table 6.7 presents the results when the number of cells was varied. The total length L is 1,000 ft and k is set to 0.1 md. A total time of 365 days is used, with a Δt of 3.65 days.

Table 6.6: Elapsed Time (s) for Varying Permeability at $L = 1000$ ft, $\Delta x = 10$ ft, $t = 365$ days

Model	**k = 5 md**	**k = 1 md**	**k = 0.1 md**
Code 1 (no TPG)	0.014	0.014	0.013
Code 2 (with TPG)	0.047	0.049	0.051

This showed a significant reduction for Code 2 as the number of cells increased. This indicates a potential benefit of the Dynamic Reporting Grid method. In a case where there is a large matrix system to solve, Code 1 solves the entire system for each timestep, but Code 2 begins by solving a very small system for the first time step, and never reaches the full system if the majority of the cells in the model are not reached.

Table 6.7: Elapsed Time (s) for Varying Cells (N) at $L = 1000$ ft, $k = 0.1md$, $t = 365$ days

Model	**N = 100**	**N = 1000**	**N = 10000**
Code 1 (no TPG)	0.014	0.133	8.017
Code 2 (with TPG)	0.047	0.056	0.076

APPLICATIONS OF THE DYNAMIC REPORTING GRID METHOD

In this chapter, two applications of the Dynamic Reporting Grid Method are discussed: scaling the threshold pressure gradient for large-scale problems and investigating the effects of the quadratic gradient term.

Determining Appropriate Threshold Pressure Gradients

Laboratory work [38] reports threshold pressure gradient values, G, obtained from low-pressure experiments on small core samples, and presents a correlation (7.1) in which G depends on permeability and viscosity, two intensive properties. Effective modeling of low-permeability flow in large-scale problems requires investigation into whether the value of G needs to be adjusted to the problem scale, and, if so, how to scale it.

Prada Correlation for the Threshold Pressure Gradient

For the code development, the core with the lowest permeability (56.6 md assuming a viscosity of 1 cP) from the Prada dataset was used [38]. This dataset is given in Figures 6.10 and 6.11. Using the data in Figure 6.12, the authors generated a plot of G (in psi/cm) pressure vs. mobility (in md/cP) shown in Figure 7.1. With a least-squares regression ($R_2 = 0.96$), they determined a correlation between the G and the mobility k/μ (7.1). As this was based on the lab data, it does not include any scaling factors based on the length of the core or the pressure differential. When this formula is added to the codes to estimate G from k, it produces reasonable results with lab-scale problems. In lab-scale simulations, this correlation is used to compute G for the chosen values of k and μ.

$$G = 16(k/\mu)^{-0.8} \tag{7.1}$$

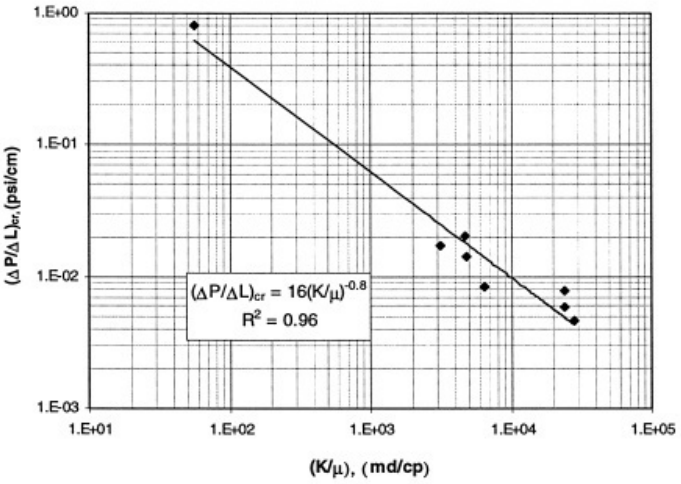

Figure 7.1: Plot of G vs Mobility with regression equation from [38]

Investigating the Need for Scaling the Threshold Pressure Gradient

To determine if scaling G is necessary, Code 2 is run for both typical lab and field scale values, and varying the permeability. The value of G is replaced by the correlation in equation (7.1) and calculated from the supplied k and μ.

Lab Scale Results The lab-scale results are presented in Figure 7.2 for $k = 10$ md and the rest of the values from Table 6.3. With a pressure drop of 12.8 psi across the core, the effect of the threshold pressure gradient remained evident at permeabilities less than 15 md when the simulation timed out at one minute.

Field Scale Results To investigate if the values of G obtained from the Prada Correlation are appropriate for a field scale model, the lower boundary pressure is adjusted to a more typical value. The lower boundary condition pressure in these field-scale models can be thought to represent a wellbore flowing pressure. This pressure is set at a lower level than the formation pressure (initial pressure and moving boundary pressure) to create the pressure gradient that causes the fluid to flow. In Figure 6.8, this boundary pressure is set to 100 psi.

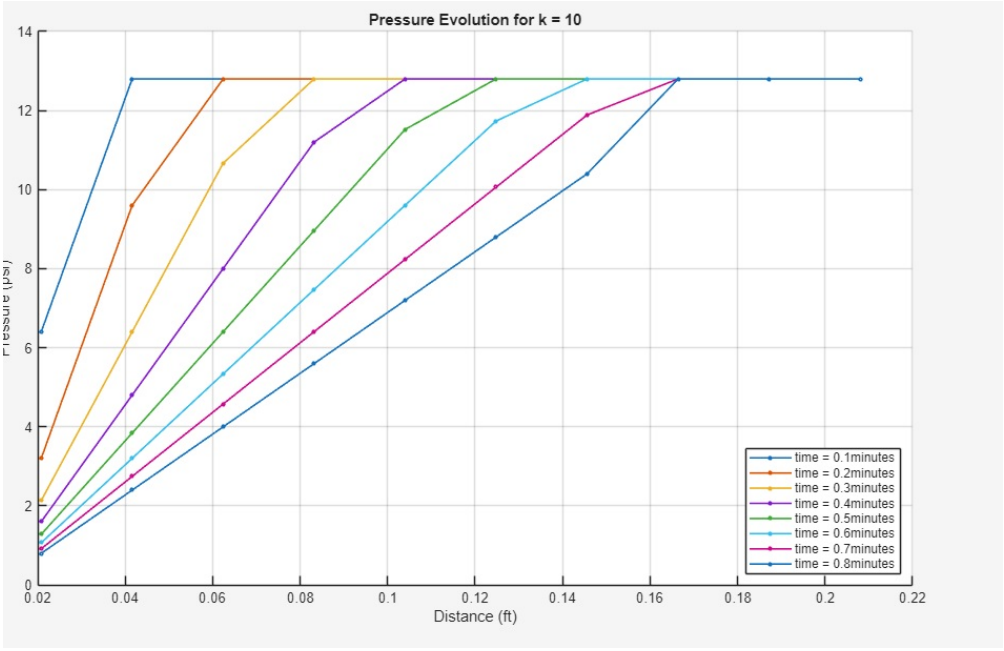

Figure 7.2: Results of Code 2 with lab-scale parameters and G from the Prada Correlation, where the model did not change past 0.8 minutes

This value was chosen in code development to ensure a large pressure gradient across the model, but is too low to represent the typical value, since producing wells with fixed pressure usually maintain values only slightly below the formation's hydrostatic pressure (BHP). A common formula for estimating this hydrostatic pressure setting is given by elSaghier [7].

$$\text{BHP (psi)} = \text{Fluid Gradient (psi/ft)} \times \text{True Vertical Depth (ft)} \tag{7.2}$$

A new pressure of 1500 psi will be used in field-scale cases to investigate permeability variations (Table 7.1). The value is estimated, based on typical field parameters,but could be adjusted for specific problems using (7.2). This results in a lower pressure drop across the model.

The results for a variety of permeability values after one year of production are shown in Figure 7.3. This suggests a need for scaling the value of G for lab scale, as the very high

permeability case shows evidence of low permeability behavior. For comparison, Code 1 is run for the same model at permeabilities of 1 and 500 md, for the same total length of one year. These figures show that in the high permeability case, the flow goes to steady state almost immediately. This confirms the need for some sort of scaling of the G factor, which is done using a technique called similitude described in the following section.

Table 7.1: This table contains the parameter values used for the next field-scale simulation experiments, with varying permeability and threshold pressure gradients found from (7.1) with potential scaling.

Parameters for Field Scale Experiments		
Field scale model: L = 1000 ft, total time = 1 year		
Names	**value**	**unit**
initial pressure	2500	psi
left boundary pressure	1500	psi
right boundary pressure	2500	psi
total compressibility	0.000061	1/psi
porosity	0.12	
viscosity	1	cP
permeability	varies	md
TPG	varies	psi/ft

Similitude

To determine how to properly scale G, a process modeled on similitude is used. Similitude (or similarity) is the process of correctly predicting full-scale prototype performance from observations from a model. This requires an understanding of how model scale affects the model results [8, 12]. To achieve similitude between a model and its prototype, the

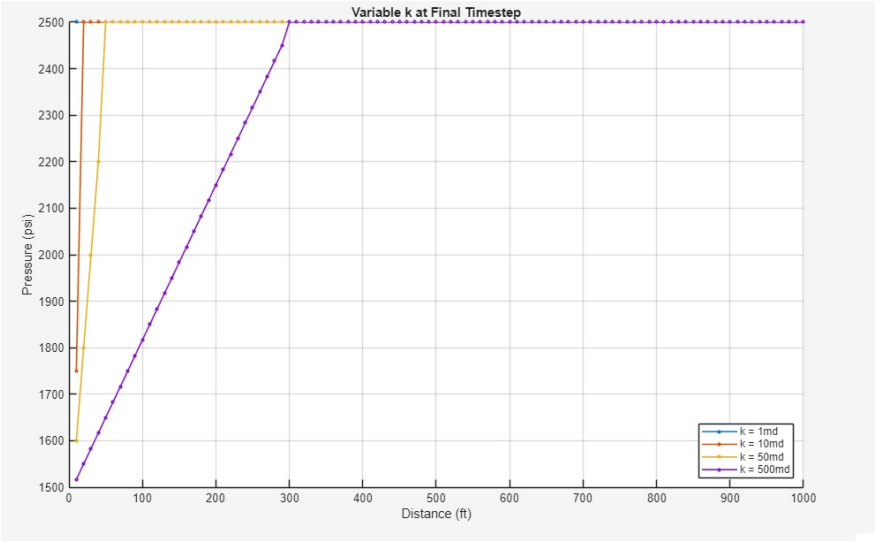

Figure 7.3: Results of Code 2 at the final timestep (at the end of one year) with field-scale parameters in Table 6.3 and varying permeability, using G from (7.1).This shows a large fraction of the model where G was not met after a simulation time of one year, even with very high permeability.

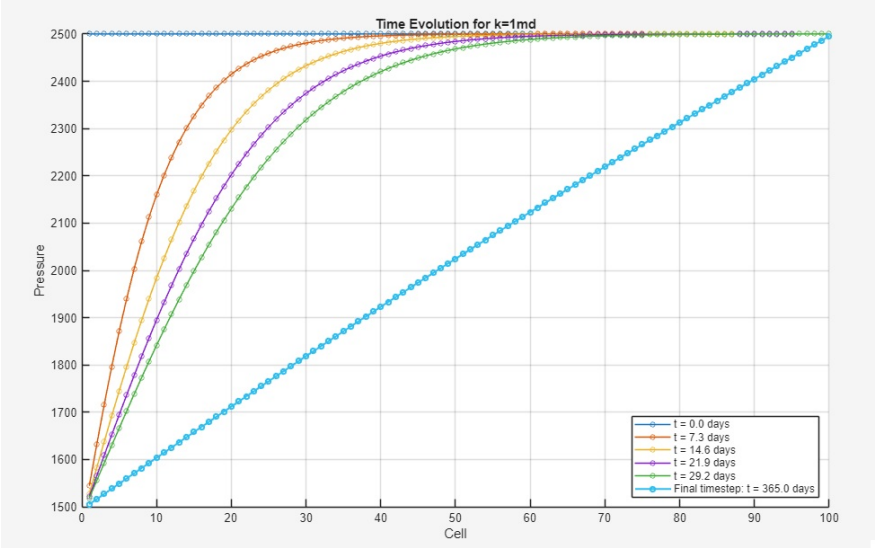

Figure 7.4: Results of Code 1 (ending after one year of simulation time) with field-scale parameters and permeability set to 1 md. While this shows steady state being achieved in one year (suggesting a need for the TPG model), it also shows that it requires many timesteps to get to steady state.

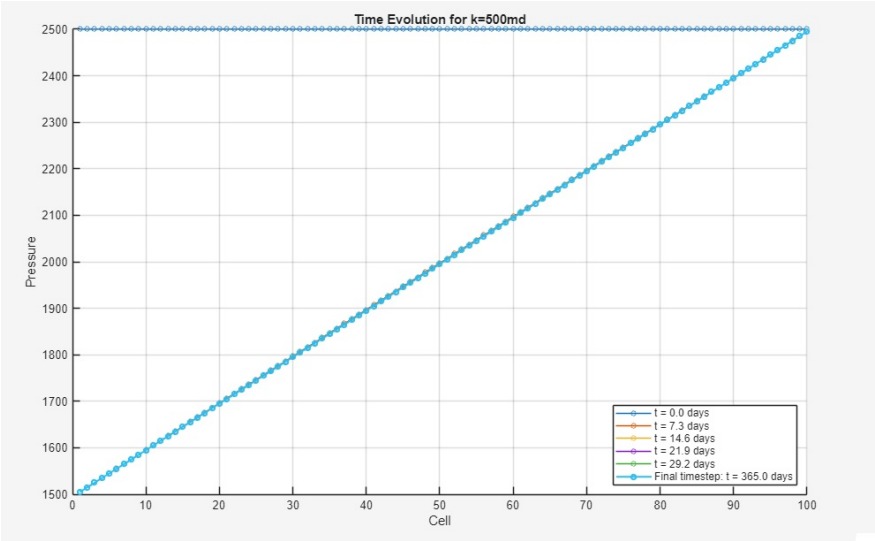

Figure 7.5: Results of Code 1 (ending after one year of simulation time) with field-scale parameters and permeability set to 500 md. This immediate drop to steady state is due to the high permeability.

values of the relevant PI group (or groups) should be the same for model and prototype. For example, when modeling a problem involving viscous forces, the model should be designed so that the Reynolds number (a dimensionless Π group) is the same as for the full-scale prototype being modeled [8]. The process to develop a scaling factor for G begins by finding the appropriate Π group, in this case, a dimensionless G. Dimensional analysis, as done to develop the definition of G_D and other parameters used to nondimensionalize the PDEs (7.3), produces a set of independent Π groups, but these are not unique and choosing the right Π group for the application requires physical insight [35].

$$G_D = \frac{kc_tG}{L} \tag{7.3}$$

In this case, when scaling to consider the length and the pressure gradient, compressibility c_t and permeability k are omitted as they are independent of pressure or position. They are omitted and replaced with the pressure gradient across the model in such a way as to form

another definition of G_D (7.4).

$$G_D = \frac{LG}{\Delta P_T} \tag{7.4}$$

where ΔP_T is the total pressure change $|P_{x=L} - P_{x=0}|$.

To scale G, this definition is applied, and is set equal for lab and field scales (f for field scale and l for lab scale). Treating the lab scale values as the model, we then rearrange to solve for G_f:

$$G_{Df} = G_{Dl} \tag{7.5}$$

$$\frac{L_f G_f}{\Delta P_{Tf}} = \frac{L_l G_l}{\Delta P_{Tl}} \tag{7.6}$$

$$G_f = G_l \frac{\Delta P_{Tf} L_l}{\Delta P_{Tl} L_f} \tag{7.7}$$

With this relationship, we use the lab-scale data from Prada [38] to create a formula to find G_f based on the given field scale parameters. Table 6.11 gives a range of values for the total pressure change across the core for the shaly sandstone and the length of the core is given in Table 6.10. Using the mean total pressure change ($\Delta P_{ave} = 14.8psi$) and the length for the shaly sandstone $L = 6.35cm = 0.167ft$, a total pressure gradient of 88.6psi/ft is obtained. This will be the baseline value for $\Delta P_{Tl}/L_l$, producing

$$G_f = G_l \frac{\Delta P_{Tf}/L_f}{88.6} \tag{7.8}$$

Applying this scaling to the same case, the results indicate that the high-permeability model exhibits no threshold pressure gradient effect, while it is evident in the low-permeability model. The G scaling formula in equation (7.8) is incorporated into the dynamic reporting grid codes, so that first the G_l value is computed using (7.1) with the provided values for k and μ. Then, using the values of ΔP_{Tf} and L_f, the appropriate G_f is found.

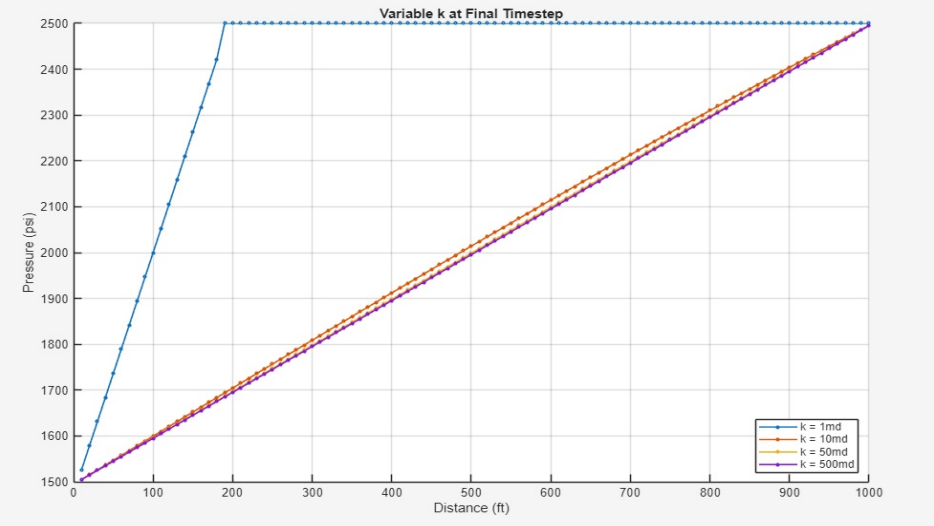

Figure 7.6: This shows the results at the end of one year for the same permeability variations with G scaling applied. A permeability of 1 md results in only about a fifth of the model involved, a permeability of 10 md involves the entire model but has not achieved steady-state, and the cases where permeability is 50 and 500 md are both steady-state.

Permeability Variations

With reasonable values of "low permeability" being hard to define precisely for field scale (Chapter 1), simulations are run for field scales for a range of low permeabilities up to 5 md. Comparing Figures 7.6, and 7.7 suggest that for this data, a permeability value between 5 and 10 md separates threshold gradient behavior from conventional flow behavior. With the permeability set to 5 md, the pressure profiles are plotted at different timesteps in Figure 7.8.

Significance of the Quadratic Pressure Gradient Term

With a clearer understanding of what constitutes low permeability across different scales, the effect of the QGT is examined to assess its potential significance under such conditions. Liu et al. [26] noted that the QGT term may become important when a larger pressure gradient is required—an outcome often associated with low-permeability systems.

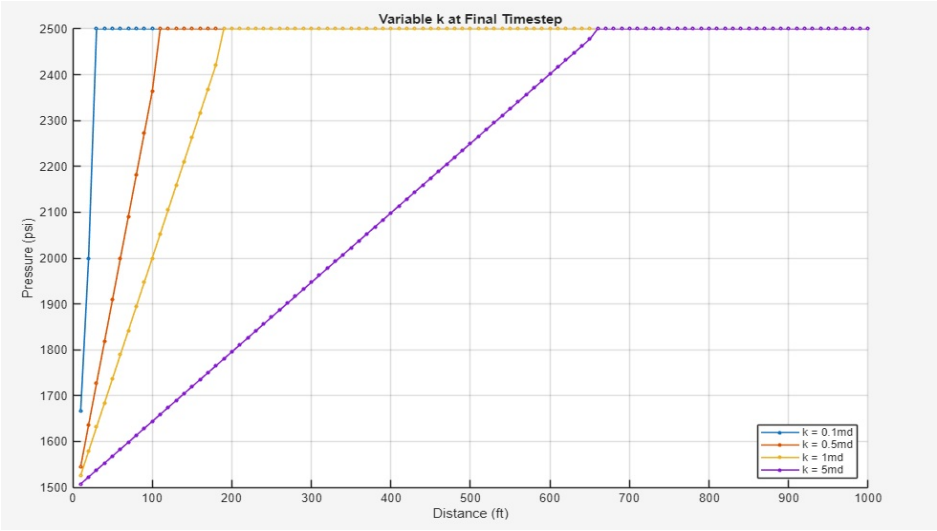

Figure 7.7: Permeability variations for the field scale model at the final time with all values exhibiting threshold pressure behavior.

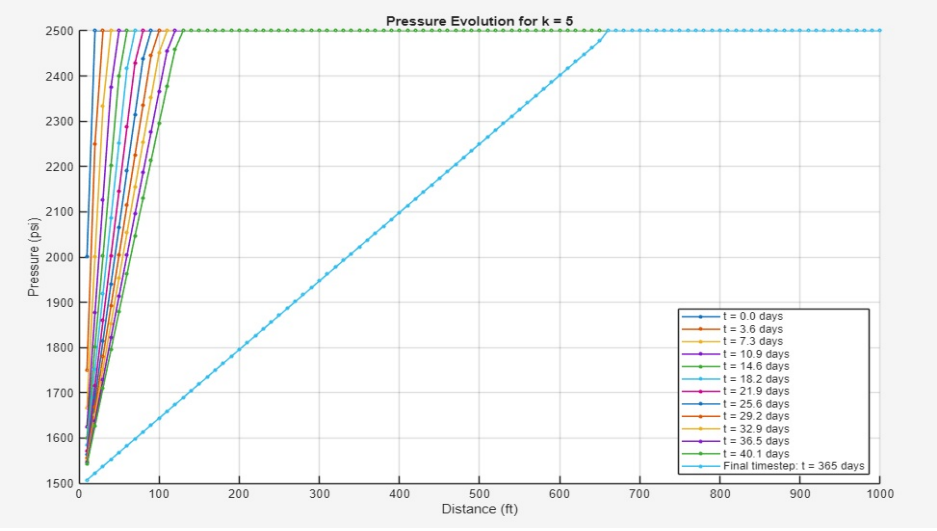

Figure 7.8: Pressure and model evolution for k = 5md

Codes 3 and 4 were developed to compare approximate solutions of the flow equation with and without the quadratic gradient term. Code 3 solves the fixed boundary condition problem of Code 1, and compares to a modified Code 1 that includes the QGT. Simulations are run using the parameters in Table 7.1, with varying permeabilities and left boundary pressures. In all codes, the right boundary pressure ($P(x = L, t)$) and initial pressure ($P(x, t = 0)$) are both set at 2500 psi. The pressure gradient that would be achieved if the model reached steady state is calculated by the following equation:

$$\text{Steady State Gradient} = \frac{P(x = L, t) - P(x = 0, t)}{L} = \frac{2500 - P(x = 0, t)}{1000} \tag{7.9}$$

This is the same as the total pressure gradient used to scale G. As the left boundary pressure can be no lower than 0 gauge pressure, the lowest possible value is 14.7 psi absolute.

Impact of QGT on Fixed Boundary Condition Models

The quadratic gradient term is often neglected in conventional flow models [16, 42], but is included here in the fixed boundary condition model to assess its potential impact under conditions of reduced permeability or increased pressure drop, and to compare to the impact when it is added to the threshold pressure gradient model under the same conditions.

Impact of Permeability Reduction with/without QGT Using Code 3, the first three runs used the left boundary pressure of 1200 psi, a reasonable field scale value. The permeability was set first to 10 md, then to 1 md, and finally to 0.1 md. In each case, the result of including the quadratic gradient term was to lower the pressure across the domain, and the maximum difference between final pressures from the two methods only slightly decreased as permeability decreased. At a permeability of 10 md, the results at the end of the allotted time of one year are close to steady state, but as the permeability decreases, the simulation is further from steady state. (Figures 7.9, 7.10, and 7.11). Table

7.2 contains the information for the varied parameters for these runs, and two runs where the left boundary pressure was reduced.

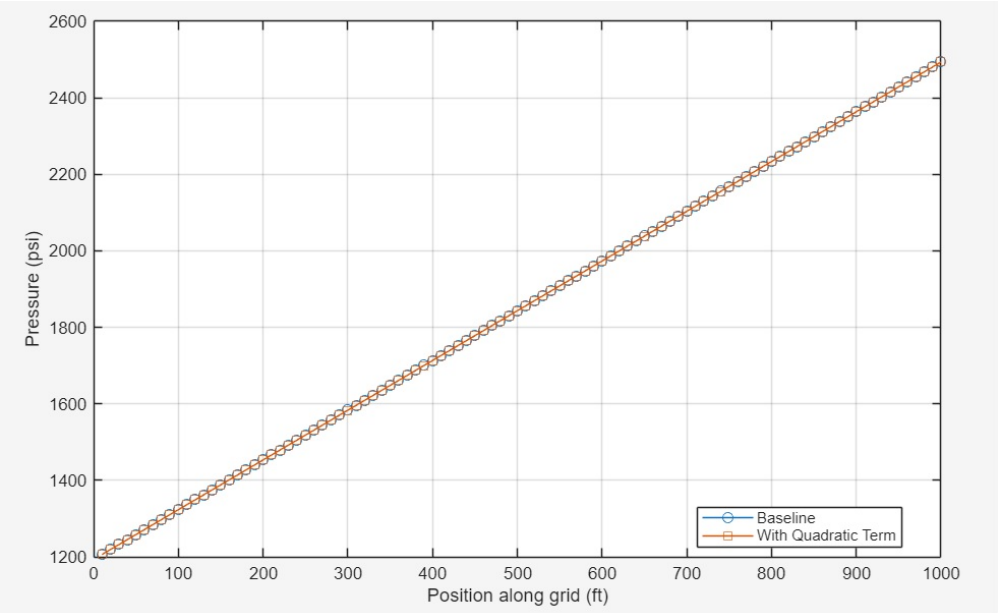

Figure 7.9: Comparison of final pressures with and without the QGT, k = 10 md

Impact of Left Boundary Pressure Reduction with/without QGT To investigate the strategy of lowering the left boundary pressure to increase fluid flow in the lowest permeability case (k = 0.1 md), $P(0,t)$ is reduced first to 100 psi, then to 14.7 psi. This results in a steady state pressure gradient of 2.4 psi/ft and 2.485 psi/ft, respectively. For these two cases, there was an increase in the maximum pressure difference, indicating that the quadratic pressure gradient term had a growing influence on the results (Table 7.2, Figures 7.12, 7.13, 7.14). While these differences are not visible on the graphs of final pressure over the length of the problem, focusing on a smaller range shows that the inclusion of the QGT results in a lower pressure.

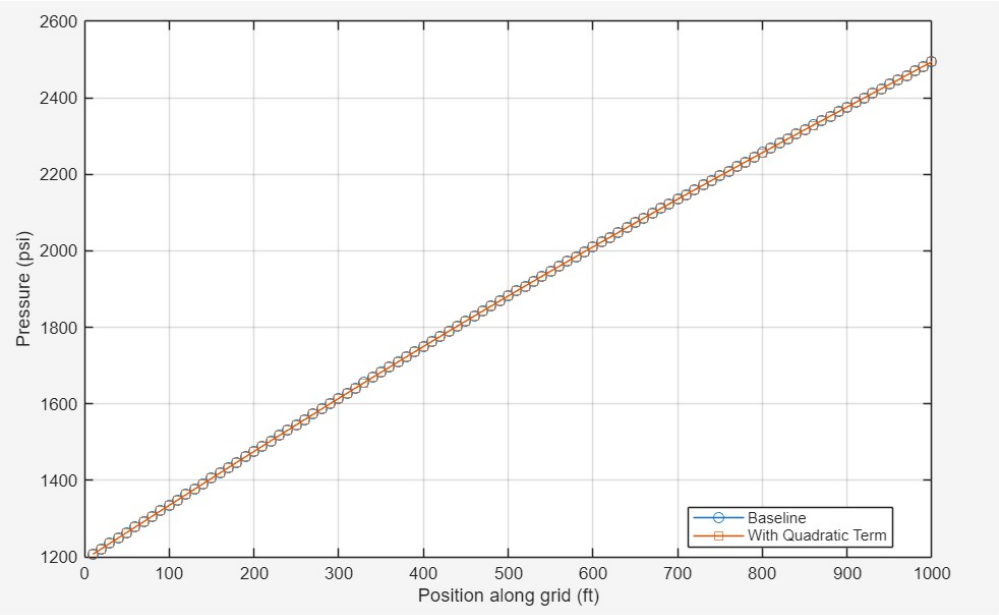

Figure 7.10: Comparison of final pressures with and without the QGT, k = 1 md

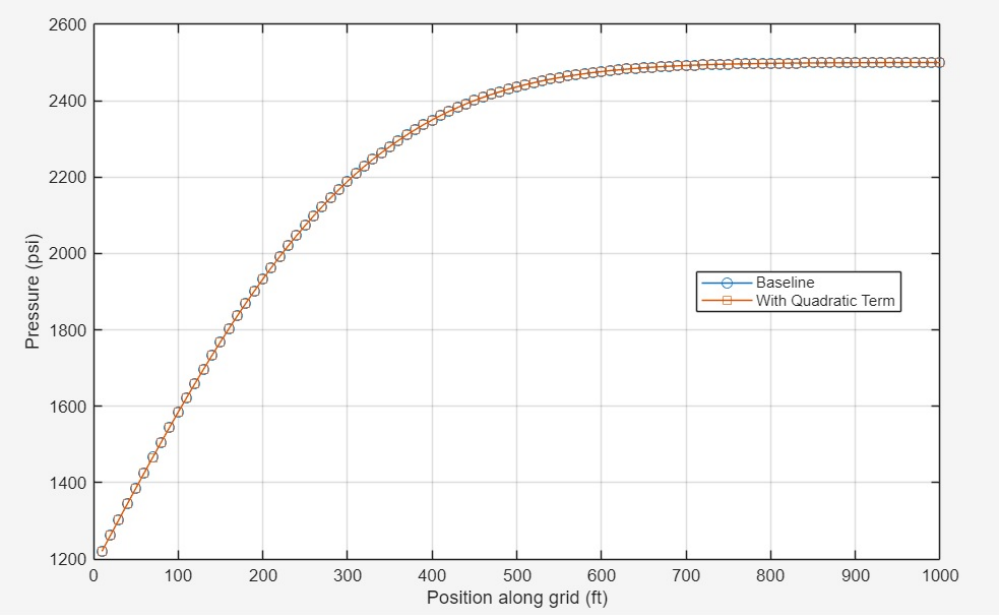

Figure 7.11: Comparison of final pressures with and without the QGT, k = 0.1 md

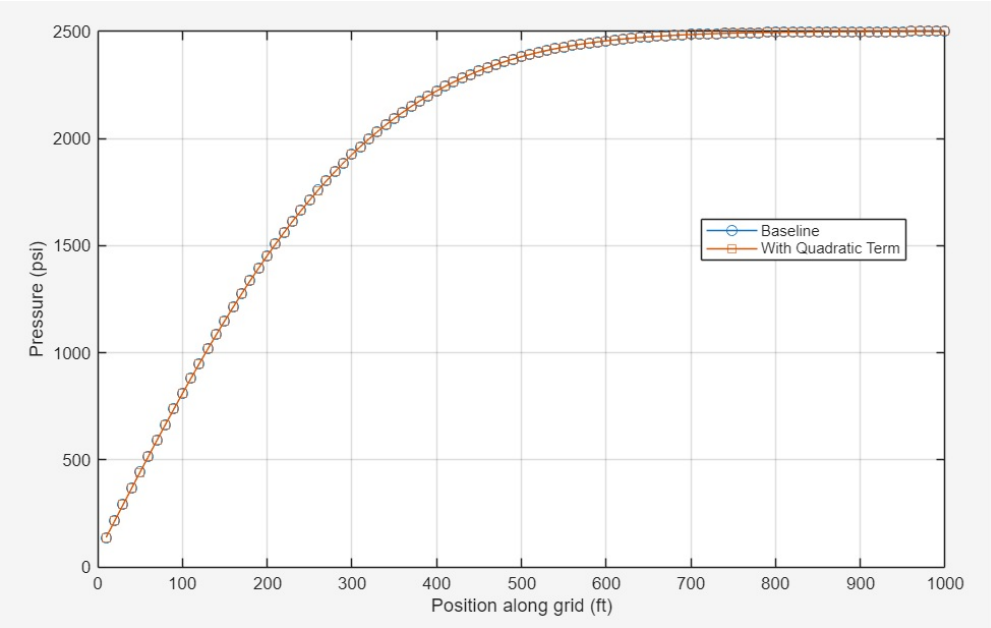

Figure 7.12: Comparison of final pressures with and without the QGT, average pressure gradient 2.4 psi/ft

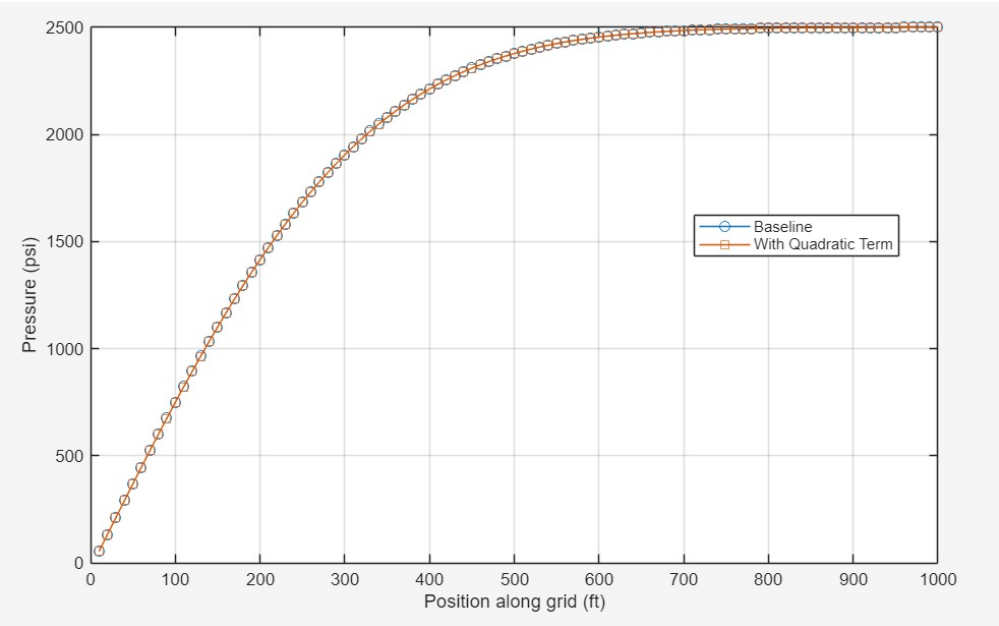

Figure 7.13: Comparison of final pressures with and without the QGT, average pressure gradient 2.458 psi/ft

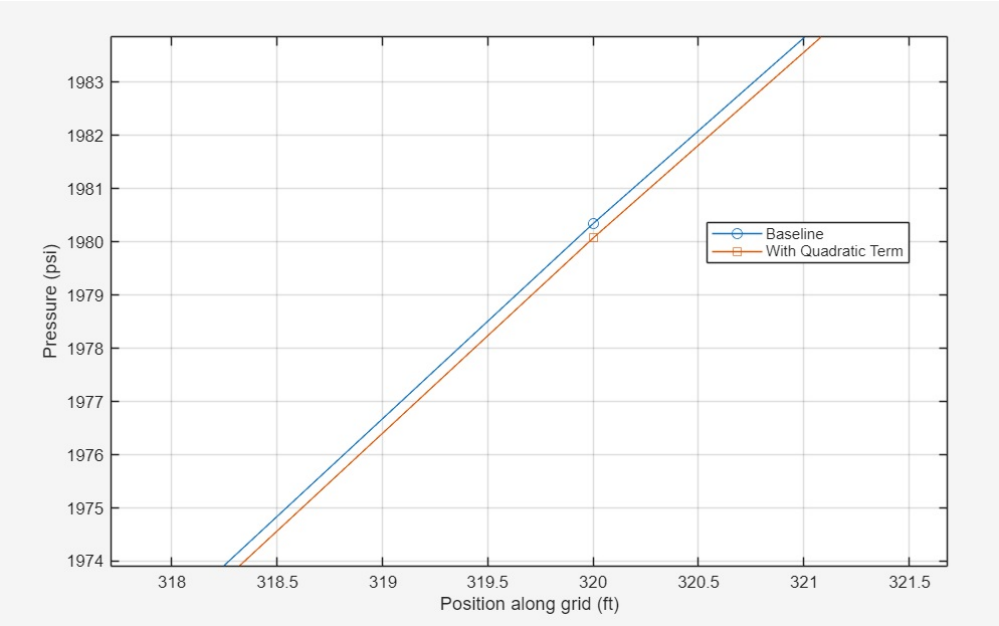

Figure 7.14: A closer view of the case with the highest average pressure gradient, showing that the inclusion of the QGT has the effect of reducing pressure

Table 7.2: Maximum absolute pressure difference observed with QGT inclusion versus exclusion

Run	k	$P(x=0,t)$	**S.S. Gradeint eq.** (7.9)	**Max. Pressure Difference**
1	10 md	1200 psi	1.3 psi/ft	0.11593 psi
2	1 md	1200 psi	1.3 psi/ft	0.11547 psi
3	0.1 md	1200 psi	1.3 psi/ft	0.11415 psi
4	0.1 md	100 psi	2.4 psi/ft	0.38906 psi
5	0.1 md	14.7 psi	2.485 psi/ft	0.4172 psi

Impact of QGT on Threshold Pressure Gradient Models

Code 4 does the same comparison as described above; however, each case includes the TPG moving boundary using the Dynamic Reporting Grid. The maximum pressure difference is still used as the measure, but in this case, the effects of adding the quadratic gradient are expected to be highest in the interior and diminish near the moving boundary.

Additional considerations are required when working with the QGT in a problem with a moving boundary condition.

As shown in the derivation of equation (6.20), the QGT is estimated here using a backwards difference instead of a central difference. This is to prevent it from reaching forward to a cell that is not yet reached by the model and using that pressure in the estimation of $(\partial p/\partial x)^2$. Finer timesteps may be required to prevent exaggerated gradients from the backward difference scheme, which can obscure the expected damping near the moving boundary. In Table 7.3, additional columns are added to track the cell number of the location in the model where the maximum pressure difference is located, and the size of the timestep Δt. Values of Δt are typically set to a value of 0.5 days instead of 3.65 days, keeping the total time of 365 days.

The permeability ranges in the fixed boundary case went from 10 md to 0.1 md. However, when code 4 was run using the permeability of 0.1 md, there were too few cells involved (only two at the end of the simulation time of one year) to investigate the effects of the quadratic gradient term. This created a very steep gradient across the few active cells. In varying the permeability, only 10 md and 1 md were used, and the varying lower boundary pressures were applied to the 1 md case instead of the 0.1 md case as before.

Impacts of Lowering Permeability and Left Boundary Pressure Reduction with/without QGT As in the case with the fixed boundary conditions (Code 3), some similar observations can be made. First, in all cases, the addition of the quadratic gradient term produced lower

Table 7.3: Maximum absolute pressure difference observed with QGT inclusion versus exclusion in the moving boundary case. The cell where the maximum pressure difference was observed and the timestep used are also provided.

Run	k	$P(x=0,t)$	**S.S. Gradeint eq.** (7.9)	**Max.** ΔP	**Cell**	Δt
1	10 md	1200 psi	1.3 psi/ft	2.8558 psi	50	0.5 days
2	1 md	1200 psi	1.3 psi/ft	2.4989 psi	10	0.5 days
3	1 md	500 psi	2 psi/ft	5.9128 psi	10	0.5 days
4	1 md	100 psi	2.4 psi/ft	8.5131 psi	10	0.5 days
5	1 md	14.7 psi	2.485 psi/ft	9.1286 psi	10	0.5 days

pressures in the active portion of the domain. In addition, the maximum pressure difference decreased slightly with a decrease in permeability. Finally, the maximum pressure difference increased with a decrease in the lower boundary pressure.

However, a major difference is that the maximum pressure difference is much higher in all cases than in the corresponding fixed boundary cases. This agrees with Liu et al. [26] in that the QGT should be considered relevant in cases where a threshold pressure gradient model is appropriate.

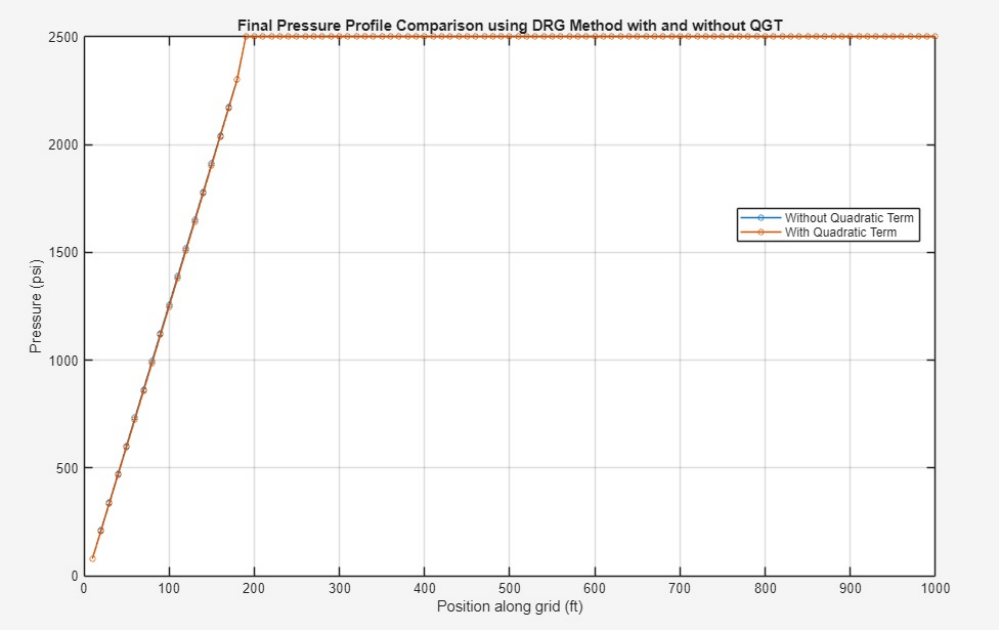

Figure 7.15: A comparison of the final pressures incorporating the TPG, with and without the QGT, with a permeability of 1 md and left boundary condition of 14.7 psi (Run 5) at the final simulation time of one year.

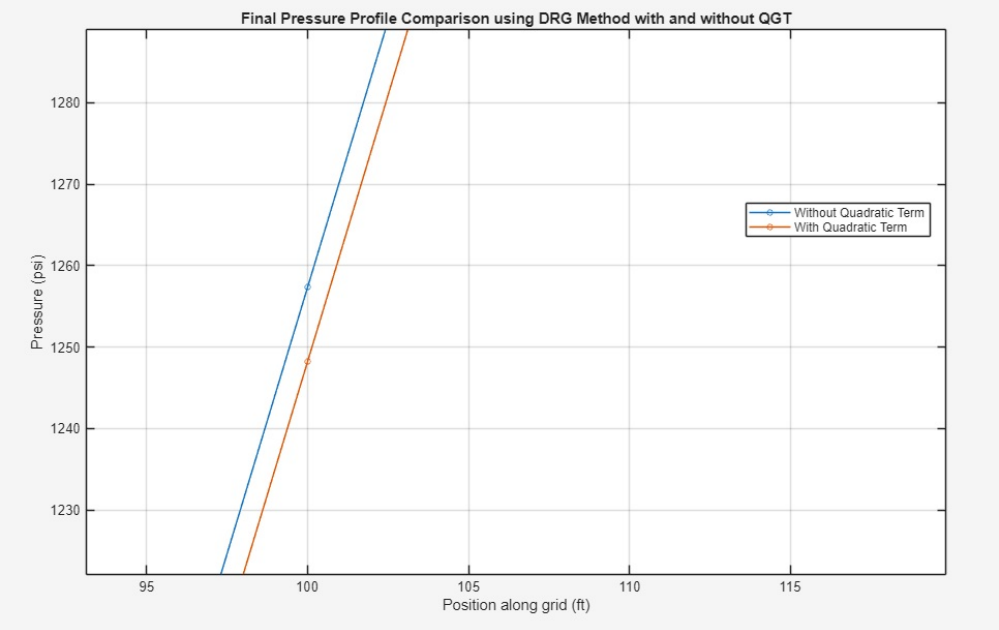

Figure 7.16: A closer look at Figure 7.15, where the pressure gradient is highest at cell 10, corresponding with a distance of 100 ft from the left boundary.

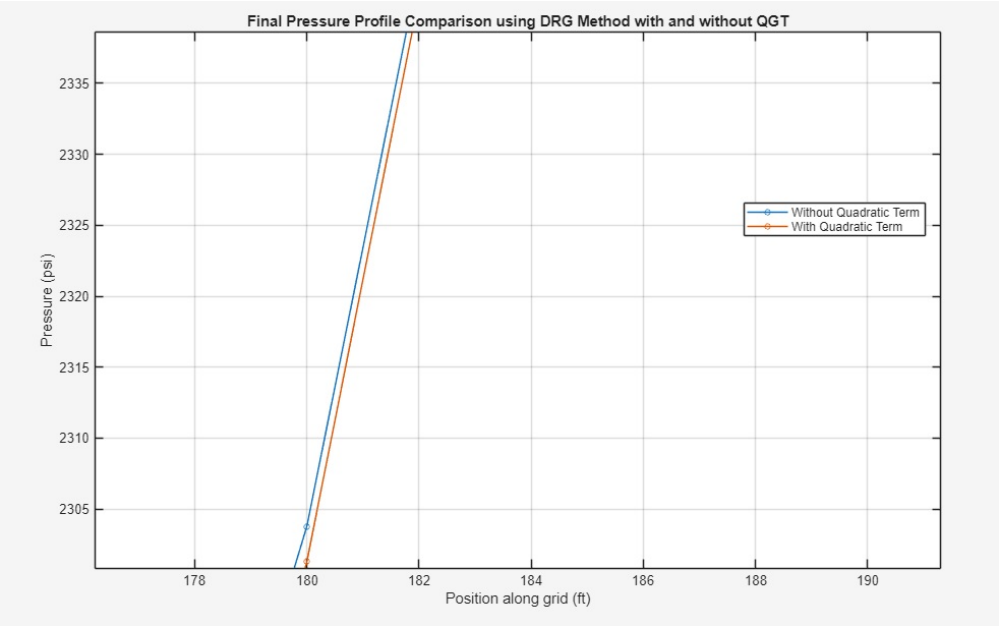

Figure 7.17: A closer look at Figure 7.15, near the moving boundary, at cell 18

CONCLUSIONS AND FUTURE WORK: EXPERIMENTAL VALIDATION

Conclusions

1. The Dynamic Reporting Grid method is a computational framework for embedding threshold behavior into PDE-based flow models. It leverages a numerical mechanism—triggering a threshold pressure in the working grid—to update the reporting grid, simulating a moving boundary condition. The method expands the grid by adding cells only when the pressure gradient exceeds a critical threshold, allowing the moving boundary to be incorporated without altering the structure of the linear system. In models with a large number of cells and a low permeability, the Dynamic Reporting Grid Method is shown to require less computing time, likely due to only solving the matrix system in the cells where the threshold pressure gradient has been met.

2. The use of the dynamic reporting grid method to model fluid flow in low-permeability porous media shows that scaling the threshold pressure gradient G produces results that align laboratory observations of low-permeability cores with observations of low-permeability reservoirs.

3. The quadratic gradient term has a small effect in the fixed boundary case, and even in a case with low permeability and the minimum low pressure boundary of 14.7 psi, it can be considered insignificant. In the case of modeling low permeability flow with a threshold pressure gradient, the quadratic pressure gradient term has a much larger effect and should be considered.

Future Work

A potential next step in the research is to compare the codes' results to those obtained in an experiment that measures pressures at different points in a core. Laboratory experiments

conducted by Prada [38] involved flowing brine through cores and sandpacks at relatively low pressure. An experiment that allows for a slightly compressible fluid and is conducted at a high enough pressure to keep gas dissolved would provide better values of pressures for further understanding the lab and field scale flow at high pressures..

To set up an experiment that can model low-permeability flow of a slightly compressible fluid, the following things need to be considered:

1. A low-permeability core of sufficient length needs to be obtained.
2. Fixed pressures need to be maintained at the inlet and outlet of the core with a pressure differential allowing fluid flow.
3. Liquid containing dissolved gas needs to be prepared and injected into the core.
4. The pressures need to be high enough to ensure that the liquid retains dissolved gas and flows linearly, requiring a robustly designed core holder.
5. There must be a way to measure pressure at intervals throughout the core.

An experimental apparatus is proposed that should meet these requirements, centered on a core holder based on a design developed at New Mexico Tech and documented by K. Wavrik [47]. The core holder for high-pressure flow from that work is shown in the images below. The core holder is built to hold a six-inch core, which is first coated with a high-heat epoxy, then cast in Cerrotru (a metal with a low melting point). Pressure taps allow for pressure measurement inside the core, and the end plates contain the inlet and outlet and provide additional strength. This setup is designed to allow fluids to only flow through the core, not between the core and the casing.

Appropriate low-permeability core samples are available for this work. Additional pressure taps will be added to the design, and the relevant parameters (viscosity of the fluid, permeability and porosity of the core, length of the core, and boundary and initial

Figure 8.1: An end view showing the thickness of the casting metal, allowing high pressures to be applied while maintaining linear flow.

Figure 8.2: The inlet of the core holder.

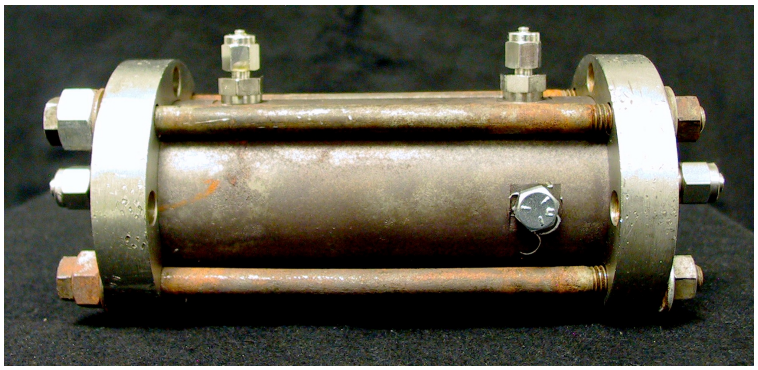

Figure 8.3: A sidelong view showing the location of two pressure taps in this model.

pressures) will be measured before the core is encased and the experiment is run. These values will be used in the codes, and the measured pressures from the pressure taps will be compared to the pressures produced by the codes.

To set up the liquid saturation process before injecting the fluid into the core, a promising method follows the work of Li et al. [23] who injected live oil in their study of the effectiveness of CO_2 floods as an enhanced oil recovery technique. The experimental set-up used in that work is shown in Figure 8.4. If this approach is used, neither the CO_2 injection tools nor the confining pressure pump would be required.

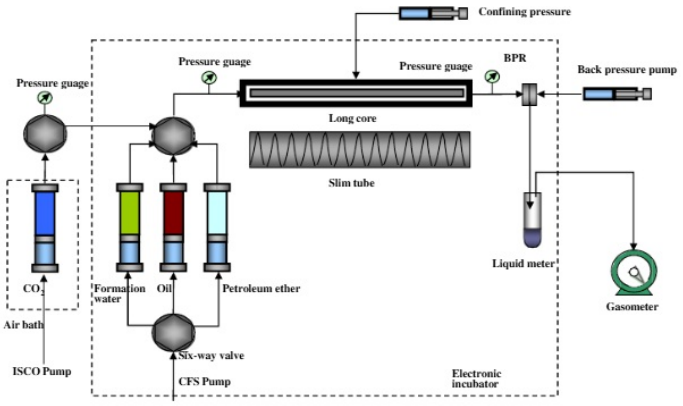

Figure 8.4: The experimental diagram from Li et al. [23] showing their method of preparation and injection of live oil (oil that contains dissolved gas)

With a significant amount of the equipment on hand or easily obtained, this will be presented as a potential joint project between two engineering departments as a Master's or Senior Design project.

REFERENCES CITED

[1] Rwanda Education Board. Geography sse, 2022. Accessed: 17 June 2025.

[2] P.F. Boulin, P.Bretonnier, N. Gland, and J.M. Lombard. Contribution of the Steady State Method to Water Permeability Measurement in Very Low Permeability Porous Media. *Oil & Gas Science and Technology - Rev. IFP Energies nouvelles*, 67:387–401, 2012.

[3] G.O. Brown. Henry Darcy and the making of a law. *Water Resources Journal*, 38(7):1006, 2002.

[4] H. Cander. What are unconventional resources? A simple definition using viscosity and permeability, 2012.

[5] Chayan Chakrabarty, S. M. Farouq Ali, and W. S. Tortike. Analytical solutions for radial pressure distribution including the effects of the quadratic-gradient term. *Water Resources Research*, 29(4):1171–1177, 1993.

[6] Ruel Vance Churchill. *Modern operational mathematics in engineering*. McGraw-Hill, 1944.

[7] Rana M. El-Saghier, Mahmoud Abu El Ela, and Ahmed El-Banbi. A model for calculating bottom-hole pressure from simple surface data in pumped wells. *Journal of Petroleum Exploration and Production Technology*, 10(5):2069–2077, 2020.

[8] D.F. Elger, B.A. LeBret, C.T. Crowe, and J.A. Robertson. *Engineering Fluid Mechanics*. Wiley, 2016.

[9] U.S. EPA. Assessment of the potential impacts of hydraulic fracturing for oil and gas on drinking water resources (final report), 2016. EPA/600/R-16/236F.

[10] Song Fuquan, Song Xingxing, Wang Yong, and Sun Yeheng. Single- and two-phase flow model in low-permeability reservoir. *Journal of Petroleum Science and Engineering*, 5:183–190, 2019.

[11] KTH Numerical Analysis Group. Finite volume discretization of the heat equation, 2013. Lecture Notes, DN2255.

[12] Valentin Heller. Scale effects in physical hydraulic engineering models. *Journal of Hydraulic Research*, 49(3):293–301, 2011.

[13] Sam Howison. Free boundary problems. `https://people.maths.ox.ac.uk/howell/Infomm/notes1.pdf`, 2005. Lecture notes, University of Oxford.

[14] Y. Huang, Z. Yang, Y. He, and Y. Wang. An overview on nonlinear porous flow in low permeability porous media. *Theoretical and Applied Mechanics Letters*, 3(2), 2013.

[15] R.Z. Jiang and R. F. Yang. *Nonlinear percolation theory and numerical simulation techniques in low permeability reservoirs*. Petroleum Industry Press, Beijing, 2010.

[16] D.L.V. Katz. *Handbook of Natural Gas Engineering*. Chemical engineering series. McGraw-Hill, 1959.

[17] Ammar Khanfer, Lazhar Bougoffa, and Smail Bougouffa. A nonclassical stefan problem with nonlinear thermal parameters of general order and heat source term. *Axioms*, 13(1):14, 2024.

[18] Shumon Koga and Miroslav Krstic. *Phase Change Model: Stefan Problem*, pages 1–13. Springer International Publishing, Cham, 2020.

[19] Yuzheng Lan, Rouzbeh Ghanbarnezhad Moghanloo, and Davud Davudov. Pore compressibility of shale formations. *SPE Journal*, 22(6):1778–1789, 2017.

[20] John (W. John) Lee. *Well testing / John Lee.* SPE textbook series ; v. 1. Society of Petroleum Engineers of AIME, New York, 1982.

[21] Qun Lei, Wei Xiong, Jiangru Yuan, Shusheng Gao, and Yu-Shu Wu. Behavior of flow through low-permeability reservoirs. *SPE EAGE Conference and Exhibition Proceedings*, June 2008.

[22] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*, volume 31 of *Cambridge Texts in Applied Mathematics*. Cambridge University Press, 2002.

[23] Fanmao Li, S. Yang, Hao Chen, X. Zhang, D. Yin, L. He, and Z. Wang. An improved method to study co2–oil relative permeability under miscible conditions. *Journal of Petroleum Exploration and Production Technology*, 5, 03 2014.

[24] P.C. Lichtner, C.I. Steefel, E.H. Oelkers, Mineralogical Society of America, and Geological Society of America. *Reactive Transport in Porous Media*. Reviews in Mineralogy & Geochemistry. Mineralogical Society of America, 1996.

[25] Knut-Andreas Lie. *An Introduction to Reservoir Simulation Using MATLAB/GNU Octave*. Cambridge University Press, 2019. Open access edition available via Cambridge Core.

[26] Wenchao Liu and Jun Yao. Numerical investigations of the effect of nonlinear quadratic pressure gradient term on a moving boundary problem of radial flow in low-permeable reservoirs with threshold pressure gradient. *Mathematical Problems in Engineering*, 2015:1–12, 2015.

[27] Wenchao Liu, Jun Yao, Zhangxin Chen, and Yuewu Liu. Effect of quadratic pressure gradient term on a one-dimensional moving boundary problem based on modified darcy's law. *Acta Mechanica Sinica*, 32(1):38–53, 2016.

[28] Wenchao Liu, Jun Yao, and Yueying Wang. Exact analytical solutions of moving boundary problems of one-dimensional flow in semi-infinite long porous media with threshold pressure gradient. *International Journal of Heat and Mass Transfer*, 55(21):6017–6022, 2012.

[29] J.D. Logan. *Applied Mathematics*. Wiley, 2013.

[30] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra, Second Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2023.

[31] L. Miaomioa, S. Benbiao, and M. Changbing, T. Xianyu. Relationship between pore throat and permeability of porous carbonate reservoir in the middle east. *Arabian Jouranl of Geosciences*, 129, 12 2021.

[32] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press, Cambridge, second edition, 2005.

[33] Douglas V. Nance. Finite volume algorithms for heat conduction. Technical Report AFRL-RW-EG-TR-2010-049, Air Force Research Laboratory, Munitions Directorate, Eglin Air Force Base, FL, 2010. Approved for public release; distribution unlimited.

[34] A. S. Odeh and D. K. Babu. Comparison of solutions of the nonlinear and linearized diffusion equations. *SPE Reservoir Engineering*, 3(04):1202–1206, 11 1988.

[35] R.C. Pankhurst. *Dimensional Analysis and Scale Factors*. Institute of Physics and the Physical Society. Monographs for students. Institute of Physics and the Physical Society by Chapman and Hall, 1964.

[36] Suhas V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Series in Computational Methods in Mechanics and Thermal Sciences. Hemisphere Publishing Corporation, Washington, D.C., 1980.

[37] G. E. Petrosky and F. Farshad. Pressure-volume-temperature correlations for gulf of mexico crude oils. *SPE Reservoir Evaluation & Engineering*, 1(5):416–420, 1998.

[38] A. Prada and F. Civan. Modification of Darcy's law for the threshold pressure gradient. *Petroleum, KeAi Communications Co.*, 22:237–240, 1999.

[39] S. Rajasekaran. *Engineering Mechanics Statics And Dynamics*. Vikas Publishing House Pvt Limited, 2009.

[40] David Reichhardt. *Variable Scale Pore System Effects on Unconventional Reservoir Fluid Properties and Production Mechanisms*. PhD thesis, Montana Technological University, May 2024.

[41] Yuntian Teng, Zihao Li, and Cheng Chen. A comprehensive review of pre-darcy flows in low-permeability porous media. *arXiv*, 2024.

[42] B.L. Towler. Spe textbook series. In *Fundamental Principles of Reservoir Engineering*. Society of Petroleum Engineers, 2002.

[43] University of North Dakota Energy & Environmental Research Center. Bakken formation, n.d. Accessed August 13, 2025.

[44] A.F. Van Everdingen and W. Hurst. The Application of the Laplace Transformation to Flow Problems in Reservoirs. *Journal of Petroleum Technology*, 1(12):305–324, 12 1949.

[45] Huimin Wang, Jianguo Wang, Xiaolin Wang, and Andrew Chan. Multi-scale insights on the threshold pressure gradient in low-permeability porous media. *Symmetry*, 12(3), 2020.

[46] Z. Wang, Z. Zhang, Y. Liu, et al. Experimental study on porosity & permeability characteristics of typical tight oil reservoirs. *Chemical Technology and Fuels and Oils*, 60:80–87, 2024.

[47] K. Wavrik. *How to Construct a 6" Core for Flooding.* Petroleum Recovery Research Center, New Mexico Tech, 2010.

[48] Hui Xu, Nannan Liu, Yan Chen, Yapeng Tian, Zhenghuai Guo, Wanjun Jiang, and Yanfeng He. A novel equivalent numerical simulation method for non-darcy seepage flow in low-permeability reservoirs. *Energies*, 15(22), 2022.

[49] C. Xu-long, T. Deng-ke, and W. Rui-he. Exact solutions for nonlinear transient flow model including a quadratic gradient term. *Applied Mathematics and Mechanics*, 25(1):1171–1177, 2024.

[50] Jun Yao, Wenchao Liu, and Zhangxin Chen. Numerical solution of a moving boundary problem of one-dimensional flow in semi-infinite long porous media with threshold pressure gradient. *Mathematical Problems in Engineering*, 2013(1):384246, 2013.

APPENDICES

APPENDIX A

SELECTED MATLAB CODES

Code 1: Conventional, Fixed Boundary Conditions

Analytic Solution Comparison

```
clear
% solver for 1D linear oil problem, pressure boundary
   conditions
%Section 1 - input variables
%
%variable inputs set to compare with analytic solution and
   example given in Katz
delx=1;
N=100;
t=6; %time in minutes
delt=delt/(60*24); %convert to days
por=0.12;
k=15.6;%input('Input the permeability in md:  ');
vis=0.2;%input('Input the viscosity in cp:  ');
ct=0.00061;%input('Input the total compressibility in 1/psi:
   ');
PI=500;%input('Input the initial pressure:  ');
PL=500;%input('Input the left boundary pressure:  ');
PR=300;%input('Input the right boundary pressure:  ');



%section 2: calculate alpha
alpha=(158*por*vis*ct*delx^2)/(k*delt);

%section 3 build matrix A
A=zeros(N,N);

%Build A

A(1,1)=-3-(3/4)*alpha;
A(N,N)=-3-(3/4)*alpha;
for i=2:N-1
    A(i,i)=-2-alpha;
end

for i=1:N-1
    A(i,i+1)=1;
    A(i+1,i)=1;
```

```
end



%Build d
d(1,1)=-(3/4)*alpha*PI-2*PL;
d(N,1)=-(3/4)*alpha*PI-2*PR;
for i=2:N-1
    d(i,1)=-alpha*PI;
end

%Built in Matlab solver
p=A\d;
%analytic solution from Katz
%time in hours
for i=1:N
tD=(2.634*10^(-4)*k*(t/60))/(por*ct*vis*i^2);
cef=erfc(1/(2*tD^(.5)));
pex(i,1)=PI-(PI-PR)*cef
end
```

Code 1:Addition of Graphing of Pressure at Multiple Timesteps

```
clear all
% Model setup
L=1000;
n=100;

% Rock properties
poro = 0.13;
k = 0.2;

% Fluid properties
mu = 2.3;
ct = 0.0000005;


% Time setup
totalt = 365;
%totalt = totalt / (60 * 24);  % convert to days
tstep = 12;
delt = totalt / tstep;
```

```
% Pressure boundary conditions
PI = 1500;
PL = 1000;
PR = 1500;


%Length
dx=L/n;

PMat=zeros(n,tstep+1);
% Put pressure in a vector
for i=1:n
    Pinit(i,1)=PI;
end
PMat(:,1)=Pinit;
%first one gets inital pressure
% Build Alpha numerical properties (cfl) and physical
   properties(props)
props=158.02*(poro*mu*ct)/(k);
cfli=dx^2/delt;
alpha=props*cfli;

% Build matrix A
A=zeros(n,n);
A(1,1)=-3-3/4*alpha;
A(n,n)=-3-3/4*alpha;
for i=2:n-1
    A(i,i)=-2-alpha;
end
% 1's above and below main diagonal
for i=1:n-1
    A(i,i+1)=1;
    A(i+1,i)=1;
end

%ITS=zeros(tstep,1);
% Time loop
for j=1:tstep
% Build\update D
D(1,1)=-.75*alpha*Pinit(1,1)-2*PL;
D(n,1)=-.75*alpha*Pinit(n,1)-2*PR;
for i=2:n-1
    D(i,1)=-alpha*Pinit(i,1);
```

```
end

% Matlab solver
p=A\D;



%display(p,'pressure in psi');
PMat(:,j+1)=p;


% Updating pressures

Pinit=p;
end
%two vectors inside plot statement - first define horizontal
   axis
figure(1);
hold on;
for i=1:tstep
 plot(1:n, PMat(:, i), '-o', 'MarkerSize', 4);
 xlabel('Cell');
 ylabel('Pressure');
 %title('Time evolution');
 grid on;
end
legend('t=0','t=1','t=2','t=3','t=4')
hold off
```

Code 2: Unconventional Flow with TPG-driven Moving Boundary Condition

```
clear
clc

% Domain and grid setup
L = 1000;
N = 100;
N2 = 2 * N;
delbigx = L / N;
delx = L / N2;

% Time setup
```

```
total_days = 365;
delt = 3.65;

%sets the total steps the process will run for to compare
%with the fixed bc
max_steps = round(total_days / delt);

% Rock and fluid properties minus perm
por = 0.12;
vis = 1;
ct = 0.000061;

%boundary and initial conditions flow right to left
PI = 2500;
PL = 1500;
PR = 2500;

% Permeability values to test
kperm_list = [5]; %can be single or multiple values
colors = lines(length(kperm_list));

figure;
hold on;

for kidx = 1:length(kperm_list)  %loop to compare multiple perm
    kperm = kperm_list(kidx);
    alpha = (158 * por * vis * ct * delx^2) / (kperm * delt);

    % Threshold pressure gradient from Prada with scaling
    G = (16 * kperm^(-0.8)) / 0.0328;
    G = G * (PR - PL) / (L * 88.6);

    PF = PI * ones(N, 1); %initialize the reporting grid
    Pinit = PI * ones(N2, 1); %initialize the working grid

    reached_cells = false(N2, 1); %tracks which working cells
       are reached
    reached_cells(1) = true; % sets leftmost cell to reached

    PMat = zeros(N, N);
    PMat(:,1) = PF;

    global_steps = 0;
```

```
global_time = 0;

bj = 1; %tracks domain size reporting grid
count = 2;

while bj <= N && global_steps < max_steps
    expanded_this_block = false;

    while global_steps < max_steps
        global_steps = global_steps + 1;
        global_time = global_time + delt;

        % Build matrix A for current size
        A = zeros(count, count);
        A(1,1) = -3 - (3/4)*alpha;
        A(count,count) = -3 - (3/4)*alpha;
        for i = 1:count-1
            A(i,i+1) = 1;
            A(i+1,i) = 1;
        end
        if count > 2
            for i = 2:count-1
                A(i,i) = -2 - alpha;
            end
        end

        % Build RHS
        d = zeros(count, 1);
        d(1) = -(3/4)*alpha * Pinit(1) - 2*PL;
        d(count) = -(3/4)*alpha * Pinit(count) - 2*PR;
        for i = 2:count-1
            d(i) = -alpha * Pinit(i);
        end

        % Solve system
        p = A \ d;
        Pinit(1:count) = p; %updates p for RHS

        % Identify reached cells
        for j = 1:count-1
            grad = abs(p(j+1) - p(j)) / delx;
            if grad > G
                reached_cells(j+1) = true;
```

```
            end
        end

        % Update PF (reporting grid)
        for l = 1:(count/2)
            idx1 = 2*l - 1;
            idx2 = 2*l;
            if reached_cells(idx1) && reached_cells(idx2)
                PF(l) = (p(idx1) + p(idx2)) / 2;
            end
        end

        % Store PF for printing multiple timesteps
        PMat(:, bj) = PF;

        % Expand grid if threshold met
        if bj < N && reached_cells(count) && ~
           expanded_this_block
            bj = bj + 1;
            count = 2 * bj; %brings more cells
            expanded_this_block = true;
            fprintf('Expanded to %d cells at time %.2f days
                after %d steps\n', count, global_time,
               global_steps);
            break;  % exit inner loop, continue with
               updated domain
        end
    end %end total time counter

    if ~expanded_this_block
        disp(['Simulation ends at bj = ', num2str(bj)]);
        break;
    end %exit display at end of change loop
end %big while loop

% Display total simulation time
%disp(['Total simulation time for k = ', num2str(kperm), '
   md: ', num2str(global_time), ' days']);
disp(['Actual timesteps executed: ', num2str(global_steps)
   ]);

% Plot pressure evolution
xvals = (1:N) * delbigx;
```

```
    figure;
    hold on;
    %comment out if only looking for final timestep (mult perm)
    for i = 1:min(bj, 12)
        tmin = (i - 1) * delt ;
        plot(xvals, PMat(:, i), '-o','MarkerSize', 2, '
           LineWidth', 1.2, ...
            'DisplayName', ['t = ' num2str(tmin, '%.1f') ' days
               ']);
    end
    %stop commenting out here
    final_time = (bj - 1) * delt;
    plot(xvals, PMat(:, bj), '-o', 'MarkerSize', 2, 'LineWidth'
       , 1.2, ...
    'DisplayName', ['Final timestep: t = ' num2str(total_days)
       ' days']);

    xlabel('Distance (ft)');
    ylabel('Pressure (psi)');
    title(['Pressure Evolution for k = ' num2str(kperm)]);
    legend show;
    grid on;
    hold off;
end %end multiple perm loop
```

Codes 3 and 4 (Codes 1 and 2 with the Addition of the Quadratic Gradient Term)

Code 3: Conventional Flow (Fixed Boundary) with Option for QGT

```
clear
clc

% Shared Model Setup
n = 10;
dx = 1;
poro = 0.10;
k = 0.1;
mu = 2;
co=0.0003; %values based on literature for this and ct
ct = 0.00005+co;
%co = 0.9 * ct;
beta = co;              % Scaling quadratic gradient term
totalt = 365;  % Total time in days
```

```
tstep = 100;
delt = totalt / tstep;

PI = 2500;
PL = 2500;
PR = 100;

L = dx * n;
cfli = dx^2 / delt;
props = 158.02 * (poro * mu * ct) / k;
alpha = props * cfli;

% Initialize Pressure Profiles
P_baseline = PI * ones(n, 1);
P_quad = PI * ones(n, 1);

% Build Coefficient Matrix A
A = zeros(n, n);
A(1,1) = -3 - 3/4 * alpha;
A(n,n) = -3 - 3/4 * alpha;
for i = 2:n-1
    A(i,i) = -2 - alpha;
end
for i = 1:n-1
    A(i,i+1) = 1;
    A(i+1,i) = 1;
end

% === Baseline Code ===
for j = 1:tstep
    D = zeros(n, 1);
    D(1) = -0.75 * alpha * P_baseline(1) - 2 * PL;
    D(n) = -0.75 * alpha * P_baseline(n) - 2 * PR;
    for i = 2:n-1
        D(i) = -alpha * P_baseline(i);
    end
    P_new = A \ D;
    P_baseline = P_new;
end

% === Quadratic Term Code ===
for j = 1:tstep
    D = zeros(n, 1);
```

```
    D(1) = -0.75 * alpha * P_quad(1) - 2 * PL;
    D(n) = -0.75 * alpha * P_quad(n) - 2 * PR;

    for i = 2:n-1
        dpdx = (P_quad(i+1) - P_quad(i-1)) / (2 * dx);
        D(i) = -alpha * P_quad(i) + beta * dpdx^2;
    end
    P_new = A \ D;
    P_quad = P_new;
end
max_diff = max(abs(P_baseline - P_quad));
disp(['Max difference between baseline and quadratic: ',
   num2str(max_diff)]);
% === Plot Final Pressure Profiles ===
grid_pts = (1:n) * dx;
plot(grid_pts, P_baseline, '-o', 'DisplayName', 'Baseline');
hold on;
plot(grid_pts, P_quad, '-s', 'DisplayName', 'With Quadratic
   Term');
xlabel('Position along grid (ft)');
ylabel('Pressure (psi)');
title('Final Pressure Profile Comparison');
legend('Location', 'best');
grid on;
```

Code 4: Moving Boundary Code with QGT Included

```
%DRG method with and without QGT (Code 4)

clear
clc

% Shared model setup
L =1000;
N = 100;
N2 = 2 * N;
delbigx = L / N;
delx = L / N2;

% Time setup
total_days = 365;
delt = 0.5;
```

```
%sets the total steps the process will run
max_steps = round(total_days / delt);

% Rock and fluid properties minus perm
por = 0.10;
vis = 1;
co=0.0003; %refeenced value for low perm as co is a feature
ct=0.00005+co; %0.0005 is form comp
beta=co; %co scales the quadratic gradeitnt
kperm=1;
%boundary and initial conditions flow right to left
PI = 2500;
PL = 14.7;
PR = 2500;



    alpha = (158 * por * vis * ct * delx^2) / (kperm * delt);

    % Threshold pressure gradient from Prada with scaling
    G = (16 * kperm^(-0.8)) / 0.0328;
    %comment out for lab scale or comparison to unscaled
    G = G * (PR - PL) / (L * 88.6);

%baseline case
    PF = PI * ones(N, 1); %initialize the reporting grid
    Pinit = PI * ones(N2, 1); %initialize the working grid

    reached_cells = false(N2, 1); %tracks which working cells
       are reached
    reached_cells(1) = true; % sets leftmost cell to reached



    global_steps = 0; %prevents from running too long
    global_time = 0;

    bj = 1; %tracks domain size reporting grid
    count = 2;

    while bj <= N && global_steps < max_steps
```

```
expanded_this_block = false;

while global_steps < max_steps
    global_steps = global_steps + 1;
    global_time = global_time + delt;

    % Build matrix A for current size (same for both)
    A = zeros(count, count);
    A(1,1) = -3 - (3/4)*alpha;
    A(count,count) = -3 - (3/4)*alpha;
    for i = 1:count-1
        A(i,i+1) = 1;
        A(i+1,i) = 1;
    end
    if count > 2
        for i = 2:count-1
            A(i,i) = -2 - alpha;
        end
    end

    % Build RHS for baseline case
    d = zeros(count, 1);
    d(1) = -(3/4)*alpha * Pinit(1) - 2*PL;
    d(count) = -(3/4)*alpha * Pinit(count) - 2*PR;
    for i = 2:count-1
        d(i) = -alpha * Pinit(i);
    end

    % Solve system for baseline
    p = A \ d;
    Pinit(1:count) = p; %updates p for RHS


    % Identify reached cells
    for j = 1:count-1
        grad = abs(p(j+1) - p(j)) / delx;
        if grad > G
            reached_cells(j+1) = true;
        end
    end

    % Update PF (reporting grid)
    for l = 1:(count/2)
```

```
                idx1 = 2*l - 1;
                idx2 = 2*l;
                if reached_cells(idx1) && reached_cells(idx2)
                    PF(l) = (p(idx1) + p(idx2)) / 2;
                end
            end

            % Expand grid if threshold met
            if bj < N && reached_cells(count) && ~
               expanded_this_block
                bj = bj + 1;
                count = 2 * bj; %brings more cells
                expanded_this_block = true;

                break;  % exit inner loop, continue with
                   updated domain
            end
        end %end total time counter

        if ~expanded_this_block
            %disp(['Simulation ends at bj = ', num2str(bj)]);
            break;
        end %exit display at end of change loop
    end %big while loop


%qgt case
    PFQGT = PI * ones(N, 1); %initialize the reporting grid
    PinitQGT = PI * ones(N2, 1); %initialize the working grid

    reached_cellsq = false(N2, 1); %tracks which working cells
       are reached
    reached_cellsq(1) = true; % sets leftmost cell to reached

global_steps = 0; %prevents from running too long
    global_time = 0;

    bj = 1; %tracks domain size reporting grid
    count = 2;

    while bj <= N && global_steps < max_steps
        expanded_this_block = false;
```

```
while global_steps < max_steps
    global_steps = global_steps + 1;
    global_time = global_time + delt;

    % Build matrix A for current size (same for both)
    A = zeros(count, count);
    A(1,1) = -3 - (3/4)*alpha;
    A(count,count) = -3 - (3/4)*alpha;
    for i = 1:count-1
        A(i,i+1) = 1;
        A(i+1,i) = 1;
    end
    if count > 2
        for i = 2:count-1
            A(i,i) = -2 - alpha;
        end
    end


    % Build RHS for QGT
    dq = zeros(count, 1);
    dq(1) = -(3/4)*alpha * PinitQGT(1) - 2*PL;
    %dq(count) = -(3/4)*alpha * PinitQGT(count) - 2*PR;
        out to
    %include it in the place where the qgt is added

    for i = 2:count-1
     if reached_cellsq(i-1) && reached_cellsq(i+1) %
        prevents reach ahead
         dpdx = (PinitQGT(i) - PinitQGT(i-1)) / (delx);
          dq(i) = -alpha * PinitQGT(i) + beta * dpdx^2;
        else
            dq(i) = -alpha * PinitQGT(i); % fallback to
                baseline
     end %conditional to make sure cells reached
    end %updating interior

    % Right boundary
    if reached_cellsq(count-1)
        dpdx = (PinitQGT(count) - PinitQGT(count-1)) /
           delx;
        dq(count) = -(3/4)*alpha * PinitQGT(count) - 2*
           PR + beta * dpdx^2;
```

```
    else
        dq(count) = -(3/4)*alpha * PinitQGT(count) - 2*
           PR;
    end %conditional to make sure cells reached


     % Solve system for qgt
    pq = A \ dq;
    PinitQGT(1:count) = pq; %updates pq for RHS

    % Identify reached cells
    for j = 1:count-1
            gradq = abs(pq(j+1) - pq(j)) / delx;
        if gradq > G
            reached_cellsq(j+1) = true;
        end
    end


    % Update PFQGT (reporting grid)
    for l = 1:(count/2)
        idx1 = 2*l - 1;
        idx2 = 2*l;
        if reached_cellsq(idx1) && reached_cellsq(idx2)
            PFQGT(l) = (pq(idx1) + pq(idx2)) / 2;
        end
    end

    % Expand grid if threshold met
    if bj < N && reached_cellsq(count) && ~
       expanded_this_block
        bj = bj + 1;
        count = 2 * bj; %brings more cells
        expanded_this_block = true;

        break;  % exit inner loop, continue with
           updated domain
    end
end %end total time counter

if ~expanded_this_block
    %disp(['Simulation ends at bj = ', num2str(bj)]);
    break;
```

```
        end %exit display at end of change loop
    end %big while loop

    % Plot pressure evolution
PF_clean = PF;
PF_clean(isnan(PF_clean)) = PI;

PFQGT_clean = PFQGT;
PFQGT_clean(isnan(PFQGT_clean)) = PI;

coarse_count = length(PF);
reach_mask = false(coarse_count, 1);
for l = 1:coarse_count
    idx1 = 2*l - 1;
    idx2 = 2*l;
    if reached_cells(idx1) && reached_cells(idx2) &&
       reached_cellsq(idx1) && reached_cellsq(idx2)
        reach_mask(l) = true;
    end
end

diff_array = abs(PF_clean - PFQGT_clean);
true_diff_array = diff_array(reach_mask);
[max_true_diff, max_true_idx_local] = max(true_diff_array);
max_true_idx = find(reach_mask);
max_true_idx = max_true_idx(max_true_idx_local);

disp(['True max difference = ', num2str(max_true_diff), ' psi
   at coarse block index = ', num2str(max_true_idx)]);


% === Plot Final Pressure Profiles ===
figure;
grid_pts = (1:N) * delbigx;
plot(grid_pts, PF_clean, '-o','MarkerSize', 4, 'DisplayName', '
   Without Quadratic Term');
hold on;
plot(grid_pts, PFQGT_clean, '-o','MarkerSize', 4, 'DisplayName'
   , 'With Quadratic Term');
xlabel('Position along grid (ft)');
ylabel('Pressure (psi)');
title('Final Pressure Profile Comparison using DRG Method with
   and without QGT');
```

```
legend('Location', 'best');
grid on;
diff = (abs(PF_clean - PFQGT_clean));
```

APPENDIX B

VERIFYING DIMENSIONLESS PARAMETERS

Conventional Flow Model

The process to verify these proposed definitions was to substitute them into the rectangular flow models and simplify them to get the dimensionless model.

$$
\begin{aligned}
\frac{\partial^2 p}{\partial x^2} &= \varphi c_t \frac{\mu}{k}\frac{\partial p}{\partial t} \\
\frac{\partial^2}{\partial x^2}\left(p_{init} - \frac{\ell^2}{kc_t}p_D\right) &= \varphi c_t \frac{\mu}{k}\frac{\partial}{\partial t}\left(p_{init} - \frac{\ell^2}{kc_t}p_D\right) \\
\frac{\partial^2}{\partial x^2}\left(\frac{\ell^2}{kc_t}p_D\right) &= \varphi c_t \frac{\mu}{k}\frac{\partial}{\partial t}\left(\frac{\ell^2}{kc_t}p_D\right) \\
\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\left(\frac{\ell^2}{kc_t}p_D\right)\right) &= \varphi c_t \frac{\mu}{k}\frac{\ell^2}{kc_t}\frac{\partial p_D}{\partial t_D}\cdot\frac{\partial}{\partial t}\left(\frac{kt}{\varphi\mu c_t L^2}\right) \\
\frac{\ell^2}{kc_t}\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\left(p_D\right)\right) &= \varphi c_t \frac{\mu}{k}\frac{\ell^2}{kc_t}\frac{k}{\varphi\mu c_t \ell^2}\frac{\partial p_D}{\partial t_D}\cdot\frac{\partial}{\partial t}\left(t\right) \\
\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\left(p_D\right)\right) &= \frac{1}{\ell^2}\frac{\partial p_D}{\partial t_D}\cdot\frac{\partial}{\partial t}\left(t\right) \\
\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\left(p_D\right)\right) &= \frac{1}{\ell^2}\frac{\partial p_D}{\partial t_D} \\
\frac{\partial}{\partial x}\left(\frac{\partial p_D}{\partial x_D}\frac{\partial}{\partial x}\left(\frac{x}{\ell}\right)\right) &= \frac{1}{\ell^2}\frac{\partial p_D}{\partial t_D} \\
\frac{\partial}{\partial x}\left(\frac{1}{\ell}\frac{\partial p_D}{\partial x_D}\right) &= \frac{1}{\ell^2}\frac{\partial p_D}{\partial t_D} \\
\frac{1}{\ell^2}\frac{\partial^2 p_D}{\partial {x_D}^2} &= \frac{1}{\ell^2}\frac{\partial p_D}{\partial t_D} \\
\frac{\partial^2 p_D}{\partial {x_D}^2} &= \frac{\partial p_D}{\partial t_D}
\end{aligned}
$$

This confirms the dimensionless form of the flow model.

Unconventional Flow Model

Here, the non-dimensionalization of the PDEs will be verified first, followed by the boundary conditions. In the first case, where the small term and the quadratic pressure gradient term are both neglected, this has already been verified above. When the PDE includes the quadratic gradient terms (4.3), we have the addition of the c_{oD} term.

$$\frac{\partial^2}{\partial x^2}\left(p_{init}-\frac{L^2}{kc_t}p_D\right)+c_o\left(\frac{\partial}{\partial x}\left(p_{init}-\frac{L^2}{kc_t}p_D\right)\right)^2=\varphi c_t\frac{\mu}{k}\frac{\partial}{\partial t}\left(p_{init}-\frac{L^2}{kc_t}p_D\right)$$

$$-\frac{L^2}{kc_t}\frac{\partial^2}{\partial x^2}\left(p_D\right)+c_o\frac{L^4}{k^2{c^2}_t}\left(\frac{\partial}{\partial x}\left(p_D\right)\right)^2=-\varphi c_t\frac{\mu}{k}\frac{L^2}{kc_t}\frac{\partial}{\partial t}\left(p_D\right)$$

We use the following result

$$c_{oD}=\frac{L^2c_o}{kc_t}\rightarrow c_o=\frac{kc_t}{L^2}c_{oD}$$

and substitute for c_o

$$-\frac{L^2}{kc_t}\frac{\partial^2}{\partial x^2}\left(p_D\right)+\frac{kc_t}{L^2}c_{oD}\frac{L^4}{k^2{c^2}_t}\left(\frac{\partial}{\partial x}\left(p_D\right)\right)^2=-\varphi c_t\frac{\mu}{k}\frac{L^2}{kc_t}\frac{\partial}{\partial t}\left(p_D\right)$$

$$\begin{aligned}
\frac{\partial^2}{\partial x^2}\left(p_D\right)-c_{oD}\left(\frac{\partial}{\partial x}\left(p_D\right)\right)^2&=\varphi c_t\frac{\mu}{k}\frac{\partial p_D}{\partial t_D}\frac{\partial}{\partial t}\left(p_D\right)\\
\frac{\partial}{\partial t}\left(p_D\right)&=\frac{\partial p_D}{\partial t_D}\cdot\frac{\partial}{\partial t}\left(\frac{kt}{\varphi\mu c_tL^2}\right)\\
\frac{\partial^2}{\partial x^2}\left(p_D\right)-c_{oD}\left(\frac{\partial}{\partial x}\left(p_D\right)\right)^2&=\varphi c_t\frac{\mu}{k}\frac{k}{\varphi\mu c_tL^2}\frac{\partial p_D}{\partial t_D}\cdot\frac{\partial}{\partial t}\left(t\right)\\
\frac{\partial^2}{\partial x^2}\left(p_D\right)-c_{oD}\left(\frac{\partial}{\partial x}\left(p_D\right)\right)^2&=\frac{1}{L^2}\frac{\partial p_D}{\partial t_D}\\
\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\left(p_D\right)\right)-c_{oD}\left(\frac{\partial}{\partial x}\left(p_D\right)\right)^2&=\frac{1}{L^2}\frac{\partial p_D}{\partial t_D}\\
\frac{\partial}{\partial x}\left(\frac{\partial p_D}{\partial x_D}\frac{\partial}{\partial x}\left(\frac{x}{L}\right)\right)-c_{oD}\left(\frac{\partial p_D}{\partial x_D}\frac{\partial}{\partial x}\left(\frac{x}{L}\right)\right)^2&=\frac{1}{L^2}\frac{\partial p_D}{\partial t_D}\\
\frac{\partial}{\partial x}\left(\frac{1}{L}\frac{\partial p_D}{\partial x_D}\right)-c_{oD}\left(\frac{1}{L}\frac{\partial p_D}{\partial x_D}\right)^2&=\frac{1}{L^2}\frac{\partial p_D}{\partial t_D}\\
\frac{1}{L^2}\frac{\partial^2 p_D}{\partial {x_D}^2}-\frac{c_{oD}}{L^2}\left(\frac{\partial p_D}{\partial x_D}\right)^2&=\frac{1}{L^2}\frac{\partial p_D}{\partial t_D}\\
\frac{\partial^2 p_D}{\partial {x_D}^2}-c_{oD}\left(\frac{\partial p_D}{\partial x_D}\right)^2&=\frac{\partial p_D}{\partial t_D}
\end{aligned}$$

If all three terms are kept, including the small term, the dimensionless form of the threshold pressure gradient G_D is also required.

$$\frac{\partial^2 p}{\partial x^2}+c_o\left(\frac{\partial p}{\partial x}\right)^2-c_o\,G\,\frac{\partial p}{\partial x}=\frac{\mu\varphi c_t}{k}\frac{\partial p}{\partial t}$$

$$\frac{\partial^2}{\partial x^2}\left(p_{init}-\frac{L^2}{kc_t}p_D\right)+\frac{kc_t}{L^2}c_{oD}\left(\frac{\partial}{\partial x}\left(p_{init}-\frac{L^2}{kc_t}p_D\right)\right)^2-\frac{kc_t}{L^2}c_{oD}\left(\frac{G_DL}{kc_t}\right)\frac{\partial}{\partial x}\left(p_{init}-\frac{L^2}{kc_t}p_D\right)$$
$$=\varphi c_t\frac{\mu}{k}\frac{\partial}{\partial t}\left(p_{init}-\frac{L^2}{kc_t}p_D\right)$$

Continuing we have

$$-\frac{L^2}{kc_t}\frac{\partial^2}{\partial x^2}(p_D)+\frac{kc_t}{L^2}c_{oD}\frac{L^4}{k^2{c_t}^2}\left(\frac{\partial}{\partial x}(p_D)\right)^2+\frac{kc_t}{L^2}c_{oD}\left(\frac{G_DL}{kc_t}\right)\frac{L^2}{kc_t}\frac{\partial}{\partial x}(p_D)=-\varphi c_t\frac{\mu}{k}\frac{L^2}{kc_t}\frac{\partial}{\partial t}(p_D)$$
$$-\frac{\partial^2}{\partial x^2}(p_D)+\frac{kc_t}{L^2}c_{oD}\frac{L^2}{kc_t}\left(\frac{\partial}{\partial x}(p_D)\right)^2+\frac{kc_t}{L^2}\frac{L}{kc_t}G_Dc_{oD}\frac{\partial}{\partial x}(p_D)=-\varphi c_t\frac{\mu}{k}\frac{\partial}{\partial t}(p_D)$$
$$-\frac{\partial^2}{\partial x^2}(p_D)+c_{oD}\left(\frac{\partial}{\partial x}(p_D)\right)^2+\frac{G_D}{L}c_{oD}\frac{\partial}{\partial x}(p_D)=-\varphi c_t\frac{\mu}{k}\frac{\partial}{\partial t}(p_D)$$
$$-\frac{\partial^2}{\partial x^2}(p_D)+c_{oD}\left(\frac{\partial}{\partial x}(p_D)\right)^2+\frac{G_D}{L}c_{oD}\frac{\partial}{\partial x}(p_D)=-\varphi c_t\frac{\mu}{k}\frac{k}{\varphi\mu c_tL^2}\frac{\partial p_D}{\partial t_D}.$$
$$-\frac{\partial^2}{\partial x^2}(p_D)+c_{oD}\left(\frac{\partial}{\partial x}(p_D)\right)^2+\frac{G_D}{L}c_{oD}\frac{\partial}{\partial x}(p_D)=-\frac{1}{L^2}\frac{\partial p_D}{\partial t_D}.$$
$$-\frac{\partial}{\partial x}\left(\frac{\partial p_D}{\partial x_D}\frac{\partial}{\partial x}\left(\frac{x}{L}\right)\right)+c_{oD}\left(\frac{\partial p_D}{\partial x_D}\frac{\partial}{\partial x}\left(\frac{x}{L}\right)\right)^2+\frac{G_D}{L}c_{oD}\left(\frac{\partial p_D}{\partial x_D}\frac{\partial}{\partial x}\left(\frac{x}{L}\right)\right)=-\frac{1}{L^2}\frac{\partial p_D}{\partial t_D}$$
$$-\frac{\partial^2 p_D}{\partial {x_D}^2}+c_{oD}\left(\frac{\partial p_D}{\partial x_D}\right)^2+G_Dc_{oD}\frac{\partial p_D}{\partial x_D}=-\frac{\partial p_D}{\partial t_D}$$
$$\frac{\partial^2 p_D}{\partial {x_D}^2}-c_{oD}\left(\frac{\partial p_D}{\partial x_D}\right)^2-G_Dc_{oD}\frac{\partial p_D}{\partial x_D}=\frac{\partial p_D}{\partial t_D}$$

To develop the initial and boundary conditions in non-dimensionalized form, the parameter s_D is also used.

Non-dimensionalizing the initial conditions

$$\begin{aligned}&t=0\rightarrow t_D=\tfrac{kt}{\varphi\mu c_tL^2}=0\\&p|_{t=o}=p_{init}\rightarrow p_D(x_D,0)=0\quad 0<x_D<1\\&s|_{t=0}=0\rightarrow s_D(t_D=0)=0\end{aligned}\tag{B.1}$$

Non-dimensionalizing the stationary right boundary condition

$$\begin{aligned}&p(x=L)=P_L\\&p_D(x_D=1)=P_{LD}\rightarrow p_D(1,t_D)=P_{LD}\quad t_D>0\end{aligned}\tag{B.2}$$

Non-dimensionalizing the moving left boundary conditions

$$
\begin{aligned}
&p(x = L - s) = P_0 = p_{init} \\
&p_D(x_D = 1 - s_D(t_D)) = 0 \rightarrow p_D(1 - s_D(t_D), t_D) = 0 \quad t_D > 0 \\
&\left.\frac{\partial p}{\partial x}\right|_{x=L-s(t)} = G \\
&\frac{\partial p}{\partial x} = \frac{\partial}{\partial x}\left(p_{init} - \frac{L^2}{kc_t} p_D\right) = -\frac{L^2}{kc_t}\frac{\partial p_D}{\partial x} = -\frac{L^2}{kc_t}\frac{\partial p_D}{\partial x_D}\frac{\partial}{\partial x}\left(\frac{x}{L}\right) \\
&G_D = \frac{kc_t G}{L} \rightarrow G = \frac{L}{kc_t} G_D \\
&-\frac{L}{kc_t}\frac{\partial p_D}{\partial x_D} = \frac{L}{kc_t} G_D \\
&\left.\frac{\partial p_D}{\partial x_D}\right|_{1-s_D(t_D)} = -G_D \quad t_D > 0
\end{aligned}
\tag{B.3}
$$