Pontifícia Universidade Católica
do Rio de Janeiro

Átila Luna Ambrósio da Silva

Physics Informed Neural Network Applied to Fractional Flow Equations

Dissertação de Mestrado

Thesis presented to the Programa de Pós–graduação em Matemática, do Departamento de Matemática da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Matemática.

Advisor : Prof. Sinesio Pesco
Co-advisor: Prof. Abelardo Borges Barreto Jr.

Rio de Janeiro
September 2023

Pontifícia Universidade Católica
do Rio de Janeiro

Átila Luna Ambrósio da Silva

Physics Informed Neural Network Applied to Fractional Flow Equations

Thesis presented to the Programa de Pós–graduação em Matemática da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Matemática. Approved by the Examination Committee:

Prof. Sinesio Pesco
Advisor
Departamento de Matemática – PUC-Rio

Prof. Abelardo Borges Barreto Jr.
Co-advisor
Departamento de Matemática – PUC-Rio

Dr. Thiago de Menezes Duarte e Silva
Schlumberger Serviços de Petróleo – Matriz

Dr. Emílio José Rocha Coutinho
Petróleo Brasileiro - Rio de Janeiro – Matriz

Dr. Eduardo da Silva Castro
Laboratório Nacional de Computação Científica – LNCC

Dr. José Roberto Pereira Rodrigues
Cenpes/Petrobras

Rio de Janeiro, September 29th, 2023

Átila Luna Ambrósio da Silva

Átila is graduated in mathematics from the Fluminense Federal University in 2022.

Bibliographic data

Silva, Átila Luna Ambrósio da

Physics Informed Neural Network Applied to Fractional Flow Equations / Átila Luna Ambrósio da Silva; advisor: Sinesio Pesco; co-advisor: Abelardo Borges Barreto Jr.. – 2023.

50 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Matemática, 2023.

Inclui bibliografia

1. Matemática – Teses. 2. Inteligência artificial. 3. Aprendizado de máquina. 4. Física. 5. Rede neural. 6. Fluxo fracionário. I. Pesco, Sinesio. II. Barreto Jr., Abelardo Borges. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Matemática. IV. Título.

CDD: 510

Acknowledgments

I want to acknowledge my brothers, Ítalo, Aine, Thomas, and Giuliano, for standing by my side during our hard life. Without them, I would not be where I am. I also would like to thank Jéssica for all the emotional support she gave me.

To my advisors, Sinésio and Abelardo, for every support and motivation they gave me to my post-graduation and even for myself. They always did everything so I could finish my Master's even with so many problems that I've been pass through.

I would like to express my immense gratitude to my post-graduation friends, for their collaboration in times of need and their entertainment during the tough moments.

I thank Petrobras for the financial support.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

Abstract

Silva, Átila Luna Ambrósio da; Pesco, Sinesio (Advisor); Barreto Jr., Abelardo Borges (Co-Advisor). **Physics Informed Neural Network Applied to Fractional Flow Equations**. Rio de Janeiro, 2023. 50p. Dissertação de Mestrado – Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

Physics Informed Machine Learning is the strategy of developing a neural network with physical constraints, commonly expressed in partial differential equations (PDEs) and their initial and boundary conditions. In this approach, the main idea is to incorporate underlying physical laws expressed in these PDEs as prior information for the neural network. In this work, we investigate the applicability of this technique to the direct problem of two-phase fluid transport in porous media, particularly in the context of gas injection in an oil reservoir, whose physical constraints are described using nonlinear first-order hyperbolic PDEs, subject to specific initial and boundary conditions. Initially, we develop the equations governing the problem without considering the fluid volume change factor to study the convergence of the solutions to these PDEs. Based on the obtained results, we introduce the volume change equations to capture the gas phase's behavior better. The fractional flux functions used in our examples were chosen as non-convex to include shock and refraction phenomena in the solutions. We also incorporate a diffusive factor, transforming the hyperbolic PDEs into parabolic ones. Through this approach, the neural network could learn consistent approximate solutions. Consequently, this effect smoothens the solution curves at the points of shock.

Keywords

Artificial intelligence; Machine learning; Physics; Neural network; Fractional flow.

Resumo

Silva, Átila Luna Ambrósio da; Pesco, Sinesio; Barreto Jr., Abelardo Borges. **Redes Neurais Baseadas em Física Aplicadas nas Equações de Fluxo Fracionário**. Rio de Janeiro, 2023. 50p. Dissertação de Mestrado – Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

Aprendizado de máquina baseado em física (Physics Informed Machine Learning), é a estratégia de desenvolver uma rede neural com restrições físicas, comumente expressas em equações diferenciais parciais (EDPs) e suas condições iniciais e de contorno. Nesta abordagem, a ideia principal é incorporar leis físicas subjacentes, expressas nessas EDPs, como informações prévias para a rede neural. Neste trabalho, investigamos a aplicabilidade desta técnica para o problema direto de transporte bifásico de fluidos em meios porosos, particularmente no contexto da injeção de gás em um reservatório de petróleo, cujas restrições físicas são descritas utilizando EDPs hiperbólicas não lineares de primeira ordem, sujeitas a condições iniciais e de contorno específicas. Inicialmente, desenvolvemos as equações que governam o problema sem considerar o fator de mudança de volume dos fluidos a fim de estudar a convergência da solução dessas EDPs. Partindo dos resultados obtidos, introduzimos as equações de mudança de volume para capturar melhor o comportamento da fase gasosa. As funções de fluxo fracionário utilizadas em nossos exemplos foram escolhidas como não-convexas para as soluções conterem fenômenos de choque e refração. Adicionalmente, incorporamos um fator difusivo, transformando assim as EDPs hiperbólicas em parabólicas. Por meio desta abordagem, a rede neural foi capaz de aprender soluções aproximadas consistentes. Como consequência, este efeito suaviza as curvas de solução nos pontos de choque

Palavras-chave

Inteligência artificial; Aprendizado de máquina; Física; Rede neural; Fluxo fracionário.

Table of contents

List of figures

List of tables

List of Abreviations

PINN - Physics Informed Neural Network

PIML - Physics Informed Machine Learning

PDE - Partial Differential Equation

AD - Automatic Differentiation

MOC - Method of Characteristics

I wouldn't have nothing if I didn't have you

Monsters, *Inc.*

1
Introduction

Deep learning has shown almost exponential growth in recent years, but supervised approaches typically require a substantial amount of labeled data to achieve good performance and convergence. The amount of data required for convergence depends on various factors, including the complexity of the task, the architecture of the deep learning model, and the quality of the data. A smaller dataset might be sufficient for simple tasks, while more complex tasks, such as image recognition or natural language processing, usually demand larger datasets. Indeed, in some cases, the amount of data is not a problem.

In most cases involving analysis of physical or biological systems, or systems encompassing knowledge fields and techniques from engineering disciplines, the cost of acquiring data is often expensive and limited, leading to the challenge of making decisions and reaching conclusions with incomplete information. Problems of this type are classified in small-data regimes, and traditional machine learning methods do not even guarantee convergence of results. To address this problem, researchers started focusing on satisfying both the available data (usually limited) and the governing partial differential equations (PDEs) or physical constraints, leading to the emergence of a new area: Physics-informed neural networks (PINN) (RAISSI, 2017b).

Recently, the capabilities of PINN were demonstrated for several problems in computer science: fluid dynamics (DOE; SMITH, 2022), electromagnetics (RAISSI, 2017b), solid mechanics (WANG et al., 2021), optimization (LU; MAO; LING, 2020) and inverse problems (TRIPATHY et al., 2021). The concept of PINN was first introduced by Raissi et al. (2017a). Later, (RAISSI, 2017b), a paper from the same authors, developed and expanded this novel approach. They suggested that if the considered PDE is well-posed, i.e., a solution exists, it is unique, and the solution changes continuously with changes in parameters (details in chapter 3), then the physics-informed machine learning (PIML) method can achieve compelling predictions given a sufficient number of neurons and layers in the neural network architecture and a sufficient number of arbitrary points in the domain called collocation points.

In (FUKS; TCHELEPI, 2020), it is shown that this approach has difficulty in modeling nonlinear hyperbolic PDEs that specifically govern biphasic transport in porous medium. Based on their experience, they also claim that this PIML deficit for problems with hyperbolic PDEs is unrelated to the network architecture or the choice of hyperparameters, i.e., for any

chosen PIML, the solution of the neural network does not converge into the solution of the hyperbolic PDE. On the other hand, in (DIAB; AL-KOBAISI, 2021), they used PIML to solve a gas drainage problem in a porous medium filled with water. Considering the Buckley-Leverett equation (BUCKLEY; LEVERETT, 1942), they obtained surprising results, as they considered gas as an incompressible fluid, which is a crucial assumption for this model. Both articles used the method described in (CRANDALL; LIONS, 1983) to transform a hyperbolic PDE into parabolic by adding a diffusive term to the equation. These works motivated us to dive into some theory of gas injection processes to improve the results.

This work considers an injection of gas into a reservoir filled with oil. As the gas phase encounters the oil in the reservoir, new mixtures are formed and come to equilibrium. The result is a set of separated components during flow where the lighter components propagate faster than the heavy ones. These separations are similar to those that occur during the chemical analysis technique known as chromatography. They are the basis for a variety of enhanced oil recovery processes (RHEE; ARIS; AMUNDSON, 1970; HELFFERICH; KLEIN, 1970). We use the equations developed in (ORR, 2007) to a chromatographic approach. We include volume change equations, which is essential for this study.

PINN is widely used currently, with different methods to implement. One of the most famous is using the Python programming language, which offers many implemented libraries to create a robust model. Several works use the TensorFlow library in Python (DIAB; AL-KOBAISI, 2021; FUKS; TCHELEPI, 2020). In this work, we used the PyTorch library available in Python, comparable to TensorFLow, considering its functionalities and efficiency (STEVENS; ANTIGA; VIEHMANN, 2020).

This work is separated into five chapters. In chapter 2, we dive into the fundamentals of the chromatographic mechanisms that are the basis of this approach. Chapter 3 treats deep learning, specifically, the PINN architecture used in our experiments. Chapter 4 compares the results to analytical solutions for three problems: the same that (DIAB; AL-KOBAISI, 2021) tackle, so we could check if our PINN is in the right direction. Then we implemented the chromatographic equations with and without volume change. Chapter 5 discusses the results, conclusions, and which topics we could improve in future works.

2
Fundamentals of flow in porous media

2.1
Motivation

The oil industry plays a significant role in the global economy and energy sector (BALDINI; BORSI; FALCHETTI, 2019). As petroleum is a finite and valuable resource, optimizing its production is vital. Using computational models and simulations has become increasingly crucial (AYERS; YAO; URBAN, 2020).

With the growth of computational power, researchers in reservoir engineering created simulations where computer models are used to predict the flow of fluids through porous media. These simulations aim to predict the future performance of the reservoir and optimize the recovery of petroleum under some operating conditions (CHEN, 2016).

Flooding gas or water into a reservoir might be necessary to enhance oil production. However, the amount of recovered oil might be limited by the accumulation of gas or water. This phenomenon, known as the "fingering effect", can cause preferential flow paths that trap significant amounts of oil in the reservoir, reducing overall oil recovery efficiency. After experimental observations of the fluid flow through sands, (BUCKLEY; LEVERETT, 1942) introduced a theory that estimates the rate at which one fluid displaces another inside the reservoir and, consequently, the change in fluid saturations. This theory is known as the fractional flow theory.

For two-phase flow, the Buckley–Leverett method ignores gravity and capillary forces under the fluid incompressibility condition. These simplifying assumptions are made to reduce the complexity of the model and facilitate the analysis of fluid flow processes in porous media.

2.2
Definitions

Before delving into the porous media flow model, let us introduce some concepts used in our work.

Phase refers to a distinct state of matter within a porous medium. In the context of two-phase fluid flow in porous media, the two primary phases are typically oil and water or gas and water. Each phase is characterized by its physical properties, such as density, viscosity, and saturation, and occupies

separate regions within the porous medium. The interactions and displacement of these phases influence the overall flow behavior and transport of fluids in the porous medium.

Component is a single chemical species. For example, the aqueous phase contains components water (H_2O), sodium chloride ($NaCl$), and dissolved oxygen (O_2), and the oil phase contains a different amount of components, e.g., C_1, C_2, C_3, etc.

Compressibility refers to the measure of a fluid's ability to change in volume or density in response to changes in pressure, particularly at a constant temperature T. Fluids can be broadly classified into three categories based on their compressibility: compressible, slightly compressible, and incompressible. Water is generally considered slightly compressible or incompressible, while natural gas is treated as compressible. For oil and its solution gas, the level of compressibility depends on the pressure conditions in the reservoir. The compressibility factor (c_f) can be defined in terms of the relative volume change (V) or density change (ρ) with respect to pressure (p) at constant temperature T:

$$c_f = -\frac{1}{V}\frac{\partial V}{\partial p}\bigg|_T = -\frac{1}{\rho}\frac{\partial \rho}{\partial p}\bigg|_T. \tag{2-1}$$

Porosity measures the amount of empty space in a porous medium, i.e., the volume fraction which is not occupied by the solid phase. It is defined as the ratio of the volume of empty spaces to the total volume of the porous medium. Porosity is essential for characterizing oil reservoirs, as it affects the amount of fluid that can be stored and flowed within the porous medium. Porosity is commonly denoted by ϕ and varies from 0.25 for a fairly permeable porous media down to 0.1 for a very low permeable one.

Viscosity is a physical property of fluids that measures their resistance to flow. Fluids with high viscosity flow more slowly and have greater resistance to movement, while fluids with low viscosity flow more easily and have less resistance to movement. The molecules are far apart for a gaseous fluid and have a low resistance to flow due to their random motion. On the other hand, a dense fluid has high resistance to flow since the molecules are close to each other. In general, fluid viscosity depends on pressure, temperature, and its compositions and is commonly denoted by μ.

Fractional Flow is a fraction that describes the proportion of a specific fluid in a porous medium. In multiphase flow systems, such as oil reservoirs, the Fractional Flow of a fluid is the ratio of the flow of that fluid to the total flow of all fluids present in the porous medium. Symbols for gas and oil in a two-phase flow system are $f_g = \lambda_g/\lambda$ and $f_o = \lambda_o/\lambda$, where $\lambda = \lambda_g + \lambda_o$ is the

total mobility.

Fluid Saturation refers to the proportion of the pore space occupied by a specific phase (water, oil, or gas) in a porous medium. This work will denote fluid saturation by the symbol S. For example, in a reservoir containing water and gas phases, the respective saturations S_w and S_g are defined such that they jointly fill the voids, satisfying the relationship:

$$S_w + S_g = 1. \tag{2-2}$$

Fluid saturation is critical in multiphase flow phenomena, influencing important functions such as capillary pressure and relative permeability. These functions exhibit strong dependence on saturations and are crucial in characterizing the flow behavior of different fluid phases within the porous medium.

Permeability of a reservoir, denoted by k, measure how easily fluids can move through the porous medium. Higher permeability allows fluids to flow more efficiently, while lower permeability restricts fluid movement. Permeability is crucial in understanding and modeling fluid flow in porous media.

Relative Permeability quantifies the effective permeability of a phase in the presence of other immiscible phases. It accounts for the reduction in flow capacity due to the presence of other fluids. The relative permeabilities to the oil and gas phases are denoted by k_{ro} and k_{rg}, respectively.

Residual Saturation: When a phase displaces another in a reservoir (e.g., gas displacing oil), a fraction of the original phase remains trapped in the pore space. This fraction is known as the residual saturation of the phase, denoted as S_{or} for oil and S_{gc} for gas. At the residual saturation of a phase, the corresponding **relative permeability** of that phase becomes zero. Therefore, at the residual saturation, the trapped phase contributes no effective flow to the overall fluid transport through the reservoir.

Mobility measures the ease with which a fluid can flow through a porous medium. It is defined as the product of the medium's permeability and the reciprocal of the fluid's viscosity. For example, the mobilities of the water, oil, and gas phases are $\lambda_w = k_{rw}/\mu_w$, $\lambda_o = k_{ro}/\mu_o$, and $\lambda_g = k_{rg}/\mu_g$, respectively.

Corey type curves are approximations to relative permeability curves (COREY, 1954). For oil and gas, they can be expressed as follows:

$$k_{rg} = \left(\frac{S_g - S_{gc}}{1 - S_{gc} - S_{or}}\right)^{N_o}, \tag{2-3}$$

and

$$k_{ro} = \left(\frac{1 - S_g - S_{or}}{1 - S_{gc} - S_{or}}\right)^{N_g}, \tag{2-4}$$

where S_g is the saturation of gas, S_{or} is the residual saturation of the oil and S_{gc} is the corresponding connate gas level. The empirical parameters N_o and N_g are "shape parameters". Through observations of the measured data and optimization using numerical simulations, these parameters can be obtained to match experimental results (COREY, 1966; BROOKS; COREY, 1964).

Phase equilibrium is the study of the equilibrium which exists between or within different states of matter, namely solid, liquid and gas. The representation of equilibrium phase behavior is a key part of the models here, because some conservation equations are based on the assumption of local chemical equilibrium (WAALS, 1988). Under that assumption, the compositions of the phases that form at a particular location in the porous medium are determined by the pressure and temperature at that location and, of course, the overall composition of the fluid present. However, in many situations, the magnitude of the pressure drop due to flow is small compared to the pressure level. In field-scale displacements, it is commonly observed that pressure gradients near injection and production wells are significant, but there are large portions of the reservoir for which gradients are small. Under those circumstances it is reasonable to evaluate equilibrium phase behavior at a single pressure and temperature. We make use here of the remarkable fact, shown originally by (GIBBS, 1878) that if the relationship between pressure, temperature, volume, and composition can be specified, as it can by an equation of state, then the composition of equilibrium phases can be calculated utilizing *equations of state*, which describe the state of matter under this set of conditions.

2.3 Darcy's Law

Darcy's Law is a fundamental fluid dynamics equation governing fluid flow through porous media. It was first formulated by Henry Darcy in 1856 and has since become a cornerstone in various fields, such as groundwater hydrology, oil reservoir engineering, and soil mechanics (DARCY, 1856). It establishes a direct relationship between the flow rate of a fluid, the pressure gradient, and the permeability of the porous medium (BEAR, 2013). By understanding Darcy's Law, engineers and scientists can gain valuable insights into fluid behavior in porous media and make informed decisions in various applications.

Darcy's Law can be derived from the Navier-Stokes equation, which is a fundamental equation in fluid dynamics describing the motion of fluid substances, taking into account factors such as viscosity, pressure, and velocity (BATCHELOR, 1967). For single-phase flow, when performing volume aver-

aging of the momentum equations, we arrive at an equivalent form known as Darcy's law (BEAR, 1972). This law states that the local flow velocity is directly proportional to the pressure gradient (BIRD; STEWART; LIGHTFOOT, 2007). Then if the flow velocity of a phase is represented as $\mathbf{v}_j$ we can express the flow velocity of a phase as:

$$\mathbf{v}_j = -\frac{kk_{rj}}{\mu_j}(\nabla P_j + \rho_{mj}\mathbf{g}), \tag{2-5}$$

where, k represents the permeability of the porous medium, k_{rj}, μ_j, P_j and ρ_{mj} denotes the relative permeability, the viscosity, pressure and mass density of the phase j respectively and $\mathbf{g}$ denotes the acceleration due to gravity.

The presence of the phase subscript, j, on the pressure in Equation 2-5 indicates that the pressures can vary across different phases. This variation is necessary when the phases are separated by curved interfaces with non-zero interfacial tension. The relationships between these pressures are typically described by capillary pressure functions of the form:

$$P_j - P_k = P_{ckj}, \qquad j = 1, \cdots, n_p, \quad k = 1, \cdots, n_p, \quad j \neq k. \tag{2-6}$$

where n_p represents the total number of phases and P_{ckj} denotes the capillary pressure across phases j and k. The capillary pressures are usually assumed to be functions of the saturations of the phases and are considered properties of the fluids and the porous medium (BEAR, 1972). These pressures can be measured through independent experiments.

2.4 Two-phase Flow Solution

Here we use the fractional flow theory to simulate the displacement of oil-filled porous media by gas. With that in mind we have assume that:

- we have a horizontal flow in one-dimensional space, i.e., let

$$\Omega = \{(x,t)|x \in \mathbb{R}, t \in \mathbb{R}\},$$

- the porous medium is homogeneous, incompressible and isotropic;
- only two phases are flowing (namely gas and oil), so $n_p = 2$, and the number of components n_c is equal to 2;
- local equilibrium exits;
- the fluids are incompressible;
- capillary pressures are negligible, i.e., Equation 2-6 turns:

$$P_j - P_k = 0, \qquad j = o, g, \quad k = o, g, \quad j \neq k,$$

then $P_j = P_k = P$ for all $j \neq k$.

Honoring these assumptions, Darcy's law (Equation 2-5) becomes:

$$v_j = -\frac{kk_{rj}}{\mu_j}\frac{\partial P}{\partial x}, \tag{2-7}$$

where $j = o$ represents the oleic phase and $j = g$ represents the gas phase, k is the total permeability of reservoir, k_{rj} is the relative permeability of j^{th} phase, μ_j is the viscosity of phase j.

The whole form of the continuity equations for multicomponent, multiphase flow:

$$\frac{\partial}{\partial t}\phi\sum_{j=1}^{2} x_{ij}\rho_j S_j + \nabla\cdot\sum_{j=1}^{2} x_{ij}\rho_j\vec{v_j} - \nabla\cdot\phi\sum_{j=1}^{2}\vec{\vec{K}}_{ij}\cdot\nabla\rho_j x_{ij} = 0, \quad i = 1,2, \tag{2-8}$$

is complex enough that it must be solved numerically unless the aforementioned simplifying assumptions are made. Hereupon, for one-dimensional flow in a Cartesian coordinate system, these equations reduces to:

$$\frac{d}{dt}\phi\sum_{j=1}^{2} x_{ij}\rho_j S_j + \frac{d}{dx}\sum_{j=1}^{2}\left(x_{ij}\rho_j v_j + \phi K_{ij}\frac{\partial\rho_j x_{ij}}{\partial x}\right), \qquad i = 1,2, \tag{2-9}$$

where ϕ is the porosity, ρ_j and S_j are the molar density and the saturation (volume fraction) of the phase j, x_{ij} is the mole fraction of component i in the phase j, v_j is the Darcy's flow velocity of the phase j and K_{ij} is the dispersion tensor for component i in phase j that includes contributions due to diffusion and dispersion.

But note that capillary pressure differences are neglected, so Equation 2-9 can be simplified by eliminating the pressure gradient from the expressions for the flow velocities. Phase flow velocities can then be written in terms of fractional flow functions, f_j defined by:

$$v_j = f_j v = f_j\sum_{k=1}^{2} v_k, \qquad j = 1,2. \tag{2-10}$$

Besides that, the effects of dispersion will be negleted, so the Equation 2-9 reduces to a set of equations that describes the interaction of pure convection with equilibrium phase behavior:

$$\frac{d}{dt}\phi\sum_{j=1}^{2} x_{ij}\rho_j S_j + \frac{d}{dx}v\sum_{j=1}^{2} x_{ij}\rho_j f_j = 0, \qquad i = 1,2. \tag{2-11}$$

Additionally, it is convenient to express Equation 2-11 in a dimensionless form by utilizing the following scaled variables:

$$x_D = \frac{x}{L}, \tag{2-12}$$

$$t_D = \frac{v_{inj}t}{\phi L}, \tag{2-13}$$

$$v_D = \frac{v}{v_{inj}}, \tag{2-14}$$

$$\rho_{jD} = \frac{\rho_j}{\rho_{inj}}, \tag{2-15}$$

where v_{inj} and ρ_{inj} are the flow velocity and density of the injected fluid, and L is the length of the one-dimensional flow system. So with that the Equation 2-11 turns:

$$\frac{d}{dt_D}\sum_{j=1}^{2} x_{ij}\rho_{jD}S_j + \frac{d}{dx_D}v_D\sum_{j=1}^{2} x_{ij}\rho_{jD}f_j = 0, \qquad i = 1, 2, \tag{2-16}$$

and the notation of Equation 2-16 can be symplified defining this two functions:

$$G_i = \sum_{j=1}^{2} x_{ij}\rho_{jD}S_j, \tag{2-17}$$

and

$$H_i = \sum_{j=1}^{2} x_{ij}\rho_{jD}f_j, \tag{2-18}$$

where G_i is an overall concentration (in moles per unit volume) of component i and H_i is an overall molar flow of component i. The equations for multicomponent, multiphase convection are, therefore,

$$\frac{\partial G_i}{\partial t_D} + \frac{\partial H_i}{\partial x_D} = 0, \quad i = 1, 2. \tag{2-19}$$

2.5 No volume change

If the displacement pressure is high enough, then the volume occupied by a component in the gas phase may not change greatly when that component transfers to the liquid phase. Components in the liquid/liquid systems that describe surfactant flooding processes also exhibit minimal volume change on mixing. In such systems, it is reasonable to assume that the partial molar volume of each component is a constant (independent of composition or phase) and hence that ideal mixing applies (ORR, 2007). Under the assumption that each component has a constant molar density, ρ_{ci}, in any phase Equation 2-19 can be simplified further. The local flow velocity is constant everywhere and equal to the injection velocity, so $v_D = 1$. Furthermore, the volume occupied

by component i in one mole of phase j is x_{ij}/ρ_{ci}, and the volume fraction of component i in phase j is:

$$c_{ij} = \frac{x_{ij}/\rho_{ci}}{\sum_{k=1}^{2} x_{kj}/\rho_{ck}}. \tag{2-20}$$

The molar density of a phase is:

$$\rho_j = \left(\sum_{i=1}^{2} \frac{x_{ij}}{\rho_{ci}} \right)^{-1}, \tag{2-21}$$

so comparing Equation 2-20 with Equation 2-21 we have:

$$\rho_{ci} c_{ij} = \rho_j x_{ij}. \tag{2-22}$$

Substituting Equation 2-22 into Equation 2-16 and dividing Equation 2-22 by ρ_{ci} with $v_D = 1$, we get:

$$\frac{\partial}{\partial t_D} \sum_{j=1}^{2} c_{ij} S_j + \frac{\partial}{\partial x_D} \sum_{j=1}^{2} c_{ij} f_j = 0, \qquad i = 1, 2. \tag{2-23}$$

By definition, $c_{1j} + c_{2j} = 1$, so we can express C_2 in terms of C_1 and then, reduce the system to a single equation:

$$\frac{\partial C_1}{\partial t_D} + \frac{\partial F_1}{\partial x_D} = 0, \tag{2-24}$$

where

$$C_1 = \sum_{j=1}^{2} c_{1j} S_j, \tag{2-25}$$

$$F_1 = \sum_{j=1}^{2} c_{1j} f_j, \tag{2-26}$$

and we can simplify the notations: $C_1 = C$, $F_1 = F$, $c_{ij} = cj$, $S_1 = S$ and $f_1 = f$.

2.6 Initial and Boundary Conditions

Before Equation 2-19 and Equation 2-24 can be solved, initial and boundary conditions must be imposed. In the cases that we will treat, solutions will be derived for initial compositions that are constant throughout a finite domain,

$$G_i(x_D, 0) = G_i^{init}, \qquad 0 < x_D <= 1, \quad i = 1, 2, \tag{2-27}$$

or

$$C_i(x_D, 0) = C_i^{init}, \qquad 0 < x_D <= 1, \quad i = 1, 2. \tag{2-28}$$

The only boundary condition required is the composition of the injected fluid,

$$G_i(0, t_D) = G_i^{inj}, \qquad 0 < t_D <= 1, \quad i = 1, 2, \tag{2-29}$$

or

$$C_i(0, t_D) = C_i^{inj}, \qquad 0 < t_D <= 1, \quad i = 1, 2. \tag{2-30}$$

Thus, at time $t_D = 0$, the composition of the fluid at the inlet changes discontinuously from the initial value to the injected value.

Problems in which the initial condition and the boundary condition upstream remain constant are called Riemann problems. These scenarios can be interpreted as depicting the progression of a discontinuity, initially positioned at $x_D = 0$, between constant initial states for $x_D < 0$, representing the injection composition, and for $x_D > 0$, signifying the initial composition. Given that the flow issue starts with advancing a discontinuity, it is not unexpected that the solutions might also exhibit discontinuities recognized as shocks. But the solutions can also have discontinuities even with smooth initial conditions.

We now finished the mathematical fundamentals of compositional changes and representations of chromatography. In chapter 4, we will solve analytically the equations developed here using the method of characteristics, and, in the next Chapter, we will present the architecture that we will use to solve the same equations numerically.

3
Deep Learning for Solving PDEs

In fields like automatic translation, predictive search (SERRANO, 2016), or recommendation engines (VERGANTI L. VENDRAMINELLI, 2020), getting a large amount of data is not a problem. Automatic translation of languages, for example, is a field in which the efficacy of machine learning models is directly proportional to the amount of data available to training (VASWANI et al., 2017). Johnson (JOHNSON et al., 2017) explored the difficulty of translation of languages with little data available. Besides that, it may be possible to improve those algorithms with little data with learning transfer, semi-supervising, or data augmentation strategies (SENNRICH; HADDOW; BIRCH, 2016).

Realistically, in many cases where neural networks are used to find solutions, we have access to noisy, incomplete, and sparse data because it can be extremely costly, resulting in a small amount of data. Those cases are common, specifically in engineering fields, and problems of this nature fall into a category known as the small-data regime. An example of this situation occurs in reservoir simulation. Suppose a deep learning model is trained to model an entire reservoir using a limited dataset due to the cost to get them. As a result, the dataset is small and sparsely distributed. In this scenario, the model may need more data to learn the mechanics of the reservoir in order to converge. The complexity of the deep learning model can lead to overfitting the few available points, resulting in a lack of generalization to the whole reservoir. Additionally, the sparse nature of the data can make it challenging to capture discriminative features effectively (ÖKTEM; BERG; HANSEN, 2018).

Generally, in these types of problems, specific "rules" govern the situation. The major of these rules are physical constraints expressed as partial differential equations (PDEs). Therefore, we can study these rules to implement them into our neural network as prior information. The following section dive into studies of this idea first introduced by (RAISSI, 2017a).

3.1
Physics Informed Neural Networks

When dealing with problems involving partial differential equations (PDEs) or physical principles, one can integrate prior knowledge and physics-based constraints, which can help mitigate problems related to limited data availability. Recently, (RAISSI, 2017a) considered a different approach for

solving nonlinear PDEs utilizing deep neural networks. The Physics Informed Neural Network (PINN) can converge to its solution by minimizing a loss function that represents the PDE constraints. This makes them suitable for problems where collecting extensive training data is challenging or costly.

The key concept in these studies is to approximate the solution of the PDE one aims to solve with a deep neural network. Consider a general PDE $u(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ given by:

$$N(\mathbf{x}, u) = F(\mathbf{x}) \qquad , if \quad \mathbf{x} \in \Omega, \quad \text{and} \tag{3-1}$$

$$B(\mathbf{x}, u) = G(\mathbf{x}) \qquad , if \quad \mathbf{x} \in \partial\Omega, \tag{3-2}$$

with $\Omega \subset \mathbb{R}^d$, N and B the differential operators on the interior and boundary, respectively, and F, G are source functions. These operators and functions define constraints on u that must be satisfied to solve the PDE. Then, to solve Equation 3-1 and Equation 3-2, the authors of (RAISSI, 2017b) suggest rewriting the PDE as:

$$\mathcal{N}(\mathbf{x}, u) := N(\mathbf{x}, u) - F(\mathbf{x}) = 0 \qquad , if \quad \mathbf{x} \in \Omega, \quad \text{and} \tag{3-3}$$

$$\mathcal{B}(\mathbf{x}, u) := B(\mathbf{x}, u) - G(\mathbf{x}) = 0 \qquad , if \quad \mathbf{x} \in \partial\Omega, \tag{3-4}$$

and minimizing the loss

$$\mathcal{L}(\hat{u}) = \frac{1}{n_I} \sum_{i=1}^{n_I} \mathcal{N}(\mathbf{x}_I^i, \hat{u})^2 + \frac{1}{n_B} \sum_{i=1}^{n_B} \mathcal{B}(\mathbf{x}_B^i, \hat{u})^2, \tag{3-5}$$

with a deep neural network. Here, n_I and n_B are the number of collocation points $\mathbf{x}_I^i \in \Omega$ and $\mathbf{x}_B^i \in \delta\Omega$, respectively, and $\hat{u}(\mathbf{x})$ is the output of the neural network that when feeding with $\mathbf{x}$, approximates the true solution $u(\mathbf{x})$.

The following section provides a mathematical motivation for the use of this method. This inspiration begins when we notice that the loss function of Equation 3-5 can be viewed as a Monte-Carlo approximation of a functional (MEER; OOSTERLEE; BOROVYKH, 2022).

3.2 Loss theory

Most of the time, computing loss functionals exactly is almost impossible work, resulting in a rare usage for neural network training. However, these loss functionals are way easier to analyze mathematically. One can avoid issues associated with collocation point distribution by focusing on a continuous form of them rather than their approximations. So let defining a continuous and

generalized loss functional:

Definition 3.1 *Let $0 < c_1, c_2 \in \mathbb{R}$ be real constants, $p \geq 1$ and $|\cdot|$ be the L^p-norm. The generalized loss functional is defined by:*

$$\hat{\mathcal{L}}(\hat{u}) = c_1 \int_\Omega |\mathcal{N}(\mathbf{x}_I^i, \hat{u})|^p + c_2 \int_\Omega |\mathcal{B}(\mathbf{x}_B^i, \hat{u})|^p. \tag{3-6}$$

Note that this loss functional is positive, but only zero if $\mathcal{N}$ and $\mathcal{B}$ are zero everywhere in Ω. Thus its global minimizer coincides with the solution of the PDE defined in Equation 3-3 and Equation 3-4. This is a property that the Monte Carlo approximation of the loss function in Equation 3-5 lacks. Minimizing it only ensures that the PDE is satisfied at a finite number of collocation points. Nevertheless, the minimization of the loss function of Equation 3-5 to solve a PDE can still be motivated by the properties of the loss functional of Equation 3-6 if the PDE has certain behavior. The following definition is the necessary property that a PDE must have so that we can continue to analyze the generalized loss functional besides the Monte-Carlo approximation.

Definition 3.2 (Wesseling, 2000) *Let $\Omega \subset \mathbb{R}^d$ be a sufficiently smooth domain and $\mathcal{N}$ and $\mathcal{B}$ defined as Equation 3-2 with N, B the operators that define the PDE, and F, G source functions. Such a PDE is called* ***well-posed*** *if for all F, G there exists a unique solution, and if for every two sets of data F_1, G_1 and F_2, G_2, the corresponding solutions u_1 and u_2 satisfy:*

$$|u_1 - u_2| \leq C(|F_1 - F_2| + |G_1 - G_2|), \tag{3-7}$$

for some fixed and finite constant $C \in \mathbb{R}$. Such a constant C will be referred to as the Lipschitz constant of the PDE.

So far, it is clear that minimizing the loss functional exactly yields a solution of the PDE. In practical problems, however, it is highly unlikely that this exact minimum will be achieved. Even if it is possible to find a solution for which the approximated loss function is zero, there is no guarantee that the loss function will be zero in the entire domain. The following theorem bridges the gap between this approximation and exact optimization of the loss functionals (MEER; OOSTERLEE; BOROVYKH, 2022).

Theorem 3.3 (Meer, Oosterlee, Borovykh, 2022) *Consider the well-posed PDE of order k given by:*

$$\mathcal{N}(\mathbf{x}, u) := N(\mathbf{x}, u) - F(\mathbf{x}) = 0 \qquad , if \quad \mathbf{x} \in \Omega, \quad and \tag{3-8}$$

$$\mathcal{B}(\mathbf{x}, u) := B(\mathbf{x}, u) - G(\mathbf{x}) = 0 \qquad , if \quad \mathbf{x} \in \partial\Omega. \tag{3-9}$$

Let the exact solution of this PDE be given by u and let the loss functional be given by Equation 3-6 for some fixed $p \geq 1$ and $c_1, c_2 > 0$. Consider some approximate solution $\hat{u}$ of which the first k (partial) derivatives exist and have finite L^p norm. Then, for any $\epsilon > 0$ there exists a $\delta > 0$ such that for the approximate solution $\hat{u}$,

$$\hat{\mathcal{L}}(\hat{u}) < \delta \implies |\hat{u} - u| < \epsilon. \tag{3-10}$$

The theorem 3.3 can then be applied to conclude that a low loss function implies that the approximations are accurate, validating the use of the approximation for well-posed problems. This result, however, does not guarantee that it is reasonable to expect neural networks to reach such small loss values. To overcome this issue, Sirignamo proposed and proved the following theorem in (SIRIGNANO; SPILIOPOULOS, 2018).

Theorem 3.4 (Sirignano and Spiliopoulos) *Let $u_n^\theta(t,x) : \mathbb{R}^{1+d} \to \mathbb{R}$ be a neural network with a single hidden-layer with n hidden units defined as:*

$$u_n^\theta(t,x) = \sum_{i=1}^{n} \beta_i \sigma \left(\alpha_{1,i} t + \sum_{j=1}^{d} \alpha_{j,i} x_j + c_j \right), \tag{3-11}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is a nonlinear activation function such as sigmoid or hyperbolic tangent function and $\theta = \{\alpha_{j,i}, \beta_i, c_j\}$ are the set of hyperparameters. Suppose that u_n^θ minimizes $\hat{\mathcal{L}}(u_n^\theta)$. Then, under certain conditions there exists a set θ of hyperparameters such that, $\hat{\mathcal{L}}(u_n^\theta) \to 0$ as $n \to \infty$

Theorem 3.4 motivates us to construct an architecture of neural network sufficiently large to guarantee the loss function approaches zero. That is sufficient to get an approximation of the true solution of the PDE displayed in Equation 3-2 by Theorem 3.3. Now we can construct an architecture that makes the loss function approaches zero. In the next chapter the architecture that we will use and details of the problems that we tackle in this study is presented.

4
Solving Volume Change Equations Using a Physics Informed Neural Network

In this chapter we apply a PINN approach to solve Equation 2-19 and Equation 2-24 and reproduce solutions of (ALMAJID; ABU-ALSAUD, 2020) and (FUKS; TCHELEPI, 2020). We see that it is necessary to introduce a diffusive factor into the PDE to the neural network converge, as seen in (FUKS; TCHELEPI, 2020). In section 4.1, we introduce the proposed architecture and their hyperparameters. Then, using the same hypothesis as (ALMAJID; ABU-ALSAUD, 2020), in section 4.2, we solve a simpler PDE to see the potential of the architecture that we use. Then in the section 4.3 we develop the analytical solution of Equation 2-24 and solve it with our methodology based on the PINN approach. In the last section, we solve Equation 2-19 that considers volume change by using our neural network and compare it to the analytical solution obtained by using the method of characteristics (MOC).

4.1
Architecture setup

To achieve consistency, we follow the same neural network definition as (ALMAJID; ABU-ALSAUD, 2020). Let $u_\theta(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}$ be an $(L-2)$-hidden layers neural network, with the number of layers equal to L, N_l neurons in l^{th} layer and $\mathbf{x} = (x_D, t_D)$, containing coordinates of dimensionless space and time, be the input vector. We denote the weight matrix in the l^{th} layer by $\mathbf{W}_l$, the bias vector by $\mathbf{b}_l$ and the combined set of all weight matrices and bias vectors as $\theta = \{\mathbf{W}_l, \mathbf{b}_l\}_{2 \leq l \leq L}$. We apply σ elementwise where σ is the hyperbolic tangent ($tanh$). Therefore, the structure of the neural network is defined as follows:

$$\text{input layer: } u_1(\mathbf{x}) = \mathbf{x}, \tag{4-1}$$

$$\text{hidden layers: } u_l(\mathbf{x}) = \sigma(\mathbf{W}_l u_{l-1}(\mathbf{x}) + \mathbf{b}_l) \text{ for } 2 \leq l \leq L-1, \quad \text{and} \tag{4-2}$$

$$\text{output layer: } u_\theta(\mathbf{x}) = u_L(\mathbf{x}) = \mathbf{W}_{L-1} u_{L-1}(\mathbf{x}) + \mathbf{b}_{L-1}. \tag{4-3}$$

where the hyperparameters we can see in Table 4.1 where we consider 8 hidden layers ($L = 10$) with 20 neurons per hidden layer ($N_l = 20$) and 20000 training steps (epochs).

To inform this neural network with physics, we define the residual PDE $\mathcal{N}$ as in Equation 3-3. The initial and boundary conditions will depend on the

	$N_{ic} = N_{bc}$	N_{cp}	L	N_l	epochs	λ	learning rate
Case 1	100	10000	10	20	20000	0.005	0.001
Case 2-3	200	20000				0.0025	

Table 4.1: Hyperparameters used in the Physics Informed Neural Network architecture. L is the number of total layers, including input and output layers, N_l is the number of neurons per layer, epoch is the number of training steps, λ is the diffusion term in each case and the learning rate is the used in Adam optimization method to training the PINN.

problem that we tackle. We introduce them in the following sections. In the same manner that u_x and u_t can be found by differentiating u with respect to its variables to build the PDE, u_θ can be differentiated with respect to its inputs to build a surrogate of the PDE. This can be achieved through automatic differentiation (AD). AD computes derivatives by applying the chain rule repeatedly. AD can be leveraged through well-documented machine learning packages such as Tensorflow (ABADI et al., 2016). In this work, differently from (ALMAJID; ABU-ALSAUD, 2020), we will use the PyTorch library (PASZKE et al., 2019) and Adam optimization method to training the PINN. While Tensorflow is an easy-to-develop models, the PyTorch library focuses on usability with careful performance considerations and is more popular in the research community (WADAWADAGI, 2023).

4.2 Buckley-Leverett Problem with a Nonconvex Flux Function

A conservation law is a PDE that describes the time evolution of some quantity or quantities that are conserved in time. Let $u(x,t)$ be an unknown scalar conserved quantity that we are solving for, and $f(u)$ be a scalar-valued function called the flux function. The scalar conservation law is defined as

$$u_t + f(u)_x = 0, \tag{4-4}$$

where u_t is the partial derivative of the dependent quantity u with respect to time (t), $f(u)$ is a nonlinear flux function, and $f(u)_x$ is the partial derivative of the flux function with respect to position (x). The solution to this problem involves a shock wave whose solution is discontinuous. In this case, $f(u)$ is nonconvex ($f''(u)$ changes sign), resulting in a more complicated solution involving both a shock and a rarefaction wave.

In the Buckley-Leverett problem proposed by (ALMAJID; ABU-ALSAUD, 2020), gas displaces water in a reservoir. The two phases are incompressible and immiscible, and capillary effects are negligible. The water saturation is $S_w(x,t) = u(x,t)$ and the gas saturation is defined as

$S_g(x,t) = 1 - S_w(x,t)$ as $0 \leq S_w \leq 1$. The domain considered is $x \in [0,1]$ and $t \in [0,1]$. The problem is governed by a coupled system of conservation equations that is supplemented by the Darcy equation (Equation 2-5). The conservation equation for the gas phase with associated initial and boundary conditions can be written as follows:

$$\frac{\partial S_g}{\partial t_D} + \frac{\partial f_g}{\partial x_D} = 0, \quad S_g(x_D, 0) = 0, \quad \text{and} \quad S_g(0, t_D) = 1 - S_{wr}, \tag{4-5}$$

where S_{wr} is the water residual saturation and the flux function $f_w(S_w)$ is also called the water fractional flow defined as follows:

$$f_g = \frac{1}{1 + \frac{k_{rw}\mu_g}{k_{rg}\mu_w}}, \tag{4-6}$$

where, k_{rw} and k_{ro} are the water and oil relative permeabilities respectively, given by the Corey model presented as:

$$k_{rw} = k^0_{rw}\left(\frac{S_w - S_{wr}}{1 - S_{gr} - S_{wr}}\right)^{N_w}, \tag{4-7}$$

and

$$k_{rg} = k^0_{rg}\left(\frac{1 - S_w - S_{gr}}{1 - S_{gr} - S_{wr}}\right)^{N_o}. \tag{4-8}$$

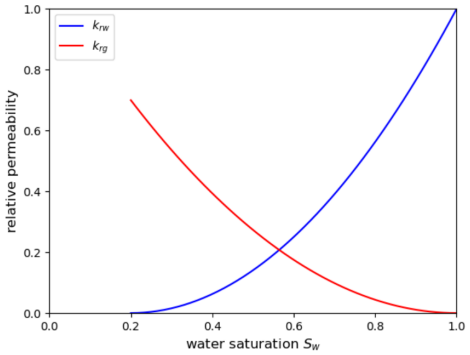

Figure 4.1: The water and gas relative permeability curves (k_{rw} and k_{rg}, respectively.

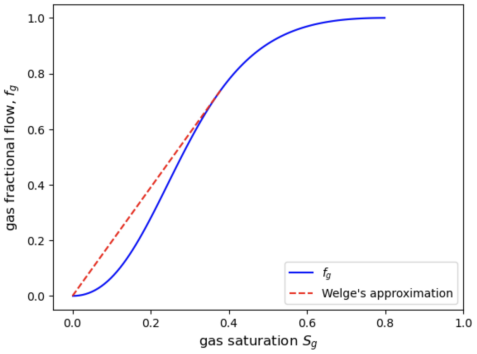

Figure 4.2: The gas fractional flow function and the constructed Welge approximation to the Buckley Leverett solution.

Here, k^0_{ro}, and k^0_{rw} are the gas and water end-point relative permeabilities respectively. Table 4.3 lists the multiphase parameters used to generate the relative permeability and the gas flux function curves. In the Equation 4-5, x_D is the dimensionless distance and t_D is the dimensionless time defined as Equation 2-12 and Equation 2-13.

The residual PDE $\mathcal{N}$ is defined as the left side of Equation 4-5 and we proceed approximating $S_g(x_D, t_D)$ by our neural network that we call S_θ:

Parameter	Gas	Water
End-point relative permeability - k^0_{rj}	0.7	1.0
Relative permeability shape parameter - n_j	2.0	2.0
Residual saturation - S_{jr}	0.0	0.2
Viscosity - μ_j	0.2	1.0

Table 4.2: Multiphase parameters used to generate the relative permeabilities for the problem. The sub-index j corresponds to g (gas) or w (water).

$$\mathcal{N}_\theta = \frac{\partial S_\theta}{\partial t_D} + \frac{\partial f_g(S_\theta)}{\partial x_D}. \tag{4-9}$$

Once $\mathcal{N}$ is defined, it can be obtained by leveraging the automatic differentiation capability of PyTorch. The set of hyperparameters in S_θ can then be learned my minimizing the loss:

$$\mathcal{L} = \mathcal{L}_{cp} + \mathcal{L}_{bc} + \mathcal{L}_{ic}, \tag{4-10}$$

where

$$\mathcal{L}_{cp}(\theta) = \frac{1}{N_{cp}} \sum_{k=1}^{N_{cp}} |\mathcal{N}_\theta(x^k_{cp}, t^k_{cp})|^2, \tag{4-11}$$

$$\mathcal{L}_{bc}(\theta) = \frac{1}{N_{bc}} \sum_{k=1}^{N_{bc}} |\mathcal{N}_\theta(x^k_{bc}, t^k_{bc}) - \mathcal{N}^k_{bc}|^2, \quad \text{and} \tag{4-12}$$

$$\mathcal{L}_{ic}(\theta) = \frac{1}{N_{ic}} \sum_{k=1}^{N_{ic}} |\mathcal{N}_\theta(x^k_{ic}, t^k_{ic}) - \mathcal{N}^k_{ic}|^2. \tag{4-13}$$

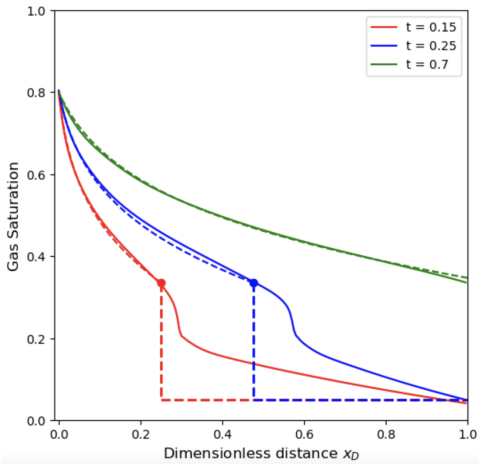

Figure 4.3: Prediction of the PINN without the diffusive term. The analytical solution is in dashed line and the prediction is the solid line.

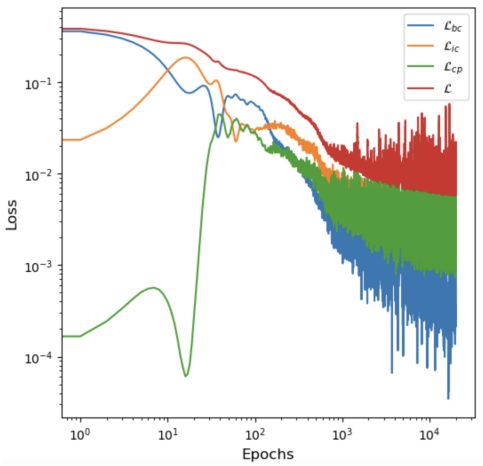

Figure 4.4: Evolution if the loss function in each training step. We can see the loss function $\mathcal{L}_{cp}, \mathcal{L}_{ic}, \mathcal{L}_{bc}$ and $\mathcal{L}$ that is the sum of all three.

Figure 4.3 shows that our PINN is not able to accurately predict the

shock. Interestingly, at late times where no shock is present, PINN has no problem predicting the solution. In Figure 4.5 we experimented adding a diffusive term to Equation 4-9 such that it is written as:

$$\mathcal{N}_\theta = \frac{\partial S_\theta}{\partial t_D} + \frac{\partial f_g(S_\theta)}{\partial x_D} - \lambda \frac{\partial^2 S_\theta}{\partial x_D^2}, \tag{4-14}$$

where λ is a hyperparameter that we choose to be as written in Table 4.1 following the study of (FUKS; TCHELEPI, 2020).

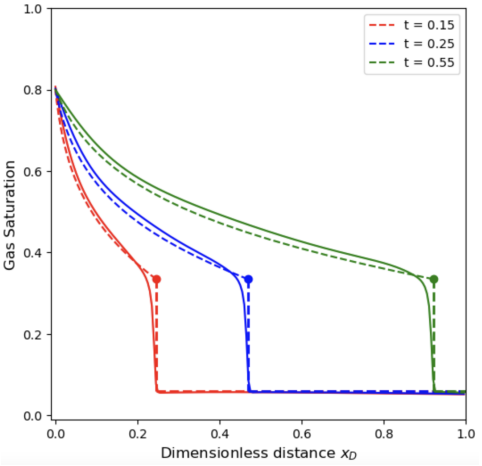

Figure 4.5: Prediction of the PINN without the diffusive term. The analytical solution is in dashed line and the prediction is the solid line.

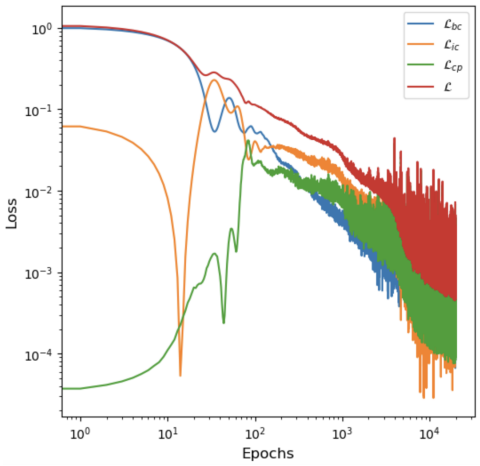

Figure 4.6: Evolution if the loss function in each training step. We can see the loss function $\mathcal{L}_{cp}, \mathcal{L}_{ic}, \mathcal{L}_{bc}$ and $\mathcal{L}$ that is the sum of all three.

These results are very close to the ones in (ALMAJID; ABU-ALSAUD, 2020), so we proceed to use this architecture on following cases. In the next section, we introduce the chromatographic equations to the problem and solve them with PINN with and without the diffusive term to compare the results.

4.3 No Volume Change Equations

With the motivation of the previous case, we will use the chromatographic approach displayed in chapter 2 of two-phase flow with no volume change on mixing. For this problem, we use a simple two-phase flow problem, displacement of an oil by a gas. Consider the flow function with two components: a one-dimensional flow in which a gas, displaces an oil containing a liquid hydrocarbon. In this situation, some of the liquid vaporizes into the gas phase, and some of the gas dissolves in the liquid phase, indicating limited mutual solubility between the components. As seen before, this problem is a generalization of the Buckley-Leverett problem of oil displacement by water (BUCKLEY; LEVERETT, 1942).

If the displacement pressure is high enough, then the volume occupied by a component in the gas phase may not change greatly when that component transfers to the liquid phase. Components in the liquid/liquid systems that describe surfactant flooding processes also exhibit minimal volume change on mixing. In such systems, it is reasonable to assume that the partial molar volume of each component is a constant (independent of composition or phase) and hence that ideal mixing applies. In other words, the volume occupied by a given amount of a component is constant no matter what phase the component appears in.

The overall volume fraction (C) and fractional volumetric flow (F) PDE for a two-component problem reduces to the equations given by Equation 2-25 and Equation 2-26:

$$C = c_1 S + c_2(1 - S), \text{and} \tag{4-15}$$

$$F = c_1 f + c_2(1 - f), \tag{4-16}$$

where C and F are the overall volume fraction and overall fractional volumetric flow of component 1 (CO_2), respectively, c_1 and c_2 are the equilibrium phase compositions, S is the saturation and f is the fractional flow function of gas phase, respectively. Therefore the system of this problem reduces to a single quasi-linear equation:

$$\frac{\partial C}{\partial t_D} + \frac{\partial F}{\partial C}\frac{\partial C}{\partial x_D} = 0. \tag{4-17}$$

The equilibrium phase compositions, c_1 and c_2, are fixed by the assumption of local chemical equilibrium and the value of each constant is described in Table 4.3. The fractional flow function is given by:

$$f = \frac{k_{rg}/\mu_g}{k_{rg}/\mu_g + k_{ro}/\mu_o}, \tag{4-18}$$

where k_{rg} and k_{ro} are the relative permeability and μ_g and μ_o are the viscosity of each phase. The relative permeability functions are described by the Corey type curves given by Equation 4-7 and Equation 4-8:

$$\begin{aligned} k_{r1} &= 0, && S < S_{gc}, \\ k_{r1} &= \left(\frac{S - S_{gc}}{1 - S_{gc} - S_{or}}\right)^2, && S_{gc} < S < 1 - S_{or}, \\ k_{r1} &= 1, && S > 1 - S_{or}, \end{aligned} \tag{4-19}$$

Parameters	Values
Residual Oil Saturation - S_{or}	0.10
Critical Gas Saturation - S_{gc}	0.05
Injection of gas - C_{inj}	1.0
Volume percent of CO_2 - C_{init}	0.05
Equilibrium of gas phase (%) - c_1	0.95
Equilibrium of liquid phase (%) - c_2	0.20
Mobility Ratio - M	2

Table 4.3: Multiphase parameters used to solve the fractional flow problem using the Method of Characteristics and PINN.

and

$$\begin{aligned} k_{r2} &= 0, && 1 - S < S_{or}, \\ k_{r2} &= \left(\frac{1 - S - S_{or}}{1 - S_{gc} - S_{or}}\right)^2, && S_{gc} < S < 1 - S_{or}, \\ k_{r2} &= 1, && 1 - S > 1 - S_{gc}, \end{aligned} \tag{4-20}$$

where S_{gc} represents a critical gas saturation, below which the vapor phase has zero relative permeability. Similarly, S_{or} is a residual oil saturation. When the liquid saturation is less than S_{or}, the liquid phase relative permeability is zero.

The phase viscosities are fixed, because the phase compositions are fixed (local equilibrium hypothesis), and the phase relative permeabilities are assumed to be functions of saturation only. Therefore, the value of C determines the value of F, because F depends only on f, which depends only on S, which in turn depends only on C as we can see if we rearrange the Equation 4-15:

$$S = \frac{C - c_2}{c_1 - c_2}, \tag{4-21}$$

and with that we can see the plot of the curve $F(C)$ in Figure 4.7.

Substitution of Equation 4-19 and Equation 4-20 into Equation 4-18 gives:

$$\begin{aligned} f &= 0, && S < S_{gc}, \\ f &= \frac{(S - S_{gc})^2}{(S - S_{gc})^2 + (1 - S - S_{or})^2/M}, && S_{gc} < S < 1 - S_{or}, \\ f &= 1, && S > 1 - S_{or}, \end{aligned} \tag{4-22}$$

where M is the viscosity ratio $\dfrac{\mu_o}{\mu_g}$.

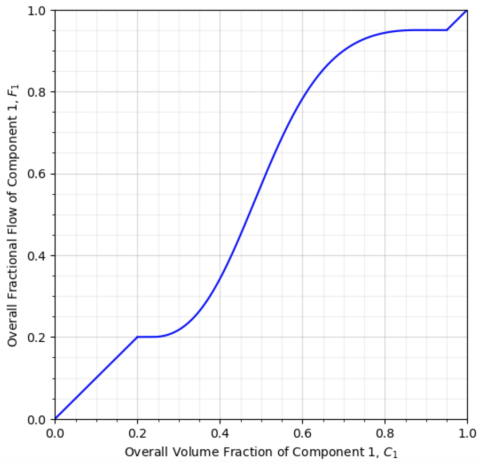

Figure 4.7: Overall Flow curve $F(C)$.

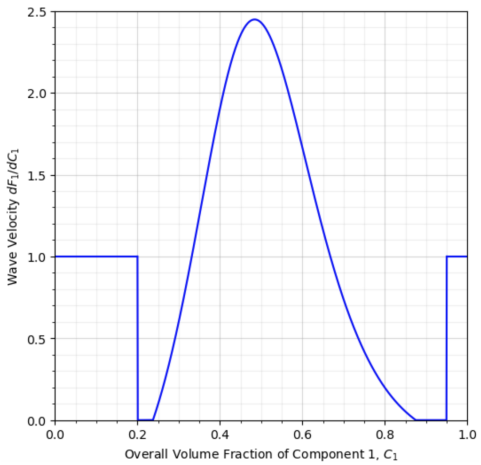

Figure 4.8: Derivative of Overall Flow curve.

If we substitute Equations 4-22 into Equation 4-16 we can get F with respect to C and this result is plotted in Figure 4.7. Within the two-phase region in Figure 4.7 , the fractional flow function has the typical S-shape often seen in measured fractional flow curves.

To find $\dfrac{\partial F}{\partial C}$ we need to differentiate Equation 4-22 with respect to C:

$$\frac{dF}{dC} = (c_1 - c2)\frac{df}{dC}, \tag{4-23}$$

and here as Equation 4-21 shows S as a function of C, we can derive it:

$$(c_1 - c2)\frac{df}{dC} = (c_1 - c2)\frac{df}{dS}\frac{dS}{dC}, \tag{4-24}$$

and finally, as we have in Equation 4-21 that $\dfrac{dS}{dC} = \dfrac{1}{c_1 - c2}$, we find that:

$$\frac{dF}{dC} = \frac{df}{dS}. \tag{4-25}$$

Therefore the Equations 4-22 can be differentiated to obtain expressions for $\dfrac{df}{dS}$ and using the method of characteristics (RHEE; ARIS; AMUNDSON, 1986; RHEE; ARIS; AMUNDSON, 1989), we can get the resulting values of $\dfrac{dF}{dC}$, plotted in Figure 4.8.

$$\begin{aligned}
&\frac{dF}{dC} = 1, && 0 < C < a, && S = 0, \\
&\frac{dF}{dC} = 0, && a < C < b, && 0 < S < S_{gc}, \\
&\frac{dF}{dC} = \frac{df}{dS}, && b < C < c, && S_{gc} < S < 1 - S_{or}, \\
&\frac{dF}{dC} = 0, && c < C < d, && 1 - S_{or} < S < 1, \\
&\frac{dF}{dC} = 1, && d < C < 1, && S = 1,
\end{aligned} \tag{4-26}$$

where

$$\begin{aligned}
a &= c_2, \\
b &= S_{gc}(c_1 - c_2) + c_2, \\
c &= S_{or}(c_2 - c_1) + c_1, \\
d &= c_1.
\end{aligned}$$

The method of characteristics is essential to help us to get the analytical solution of the PDE so we can create a "measure stick" that will help us to check if the results are consistent. To complement this method, we also use the Welge approximation. This approach is well-known for the case of Water/Oil displacement where the two fluids are incompressible.

Now, we utilize PINN to estimate the concentration profile $C(x_D, t_D)$ along the dimensionless time and distance of the system. In a similar processed as presented before, we begin by defining $\mathcal{N}_\theta(x_D, t_D)$ to be the left-hand side of Equation 4-17:

$$\mathcal{N}_\theta(x_D, t_D) = \frac{\partial C_\theta}{\partial t_D} + \frac{\partial F(C_\theta)}{\partial x_D}. \tag{4-27}$$

The solution $C(t_D, x_D)$ to the PDE is approximated by the PINN parameterized by the set of parameters θ as in Equation 4-1, Equation 4-2 and Equation 4-3 with the activation function σ being the hyperbolic tangent function and the set of hyperparameters are described in Table 4.4. In this case, as C is a function of the saturation S, the neural network will be S_θ. It will learn the parameters to approximate the function S and we can use the constants c_1 and c_2 to get C_θ.

Now we define the loss function used to train the PINN. Let the loss $\mathcal{L}$ be composed of three components described as:

$$\mathcal{L}(\theta) = \mathcal{L}_{cp}(\theta) + \mathcal{L}_{bc}(\theta) + \mathcal{L}_{ic}(\theta), \tag{4-28}$$

where $\mathcal{L}_{bc}(\theta)$ is the mean squared error from the boundary conditions, $\mathcal{L}_{ic}(\theta)$

Hyperparameters	Values
Hidden-layers $(L-2)$	6
Neurons per layer (N_l)	20
Epochs	20000
Diffusive term (λ)	0.005

Table 4.4: Hyperparameters used in the architecture of PINN.

is the mean squared error from the initial conditions and $\mathcal{L}_{cp}(\theta)$ is the mean squared error from the residual evaluated at collocation points within the physical domain:

$$\mathcal{L}_{cp}(\theta) = \frac{1}{N}\sum_{k=1}^{N_{cp}} |\mathcal{N}_\theta(x_{cp}^k, t_{cp}^k)|^2, \tag{4-29}$$

$$\mathcal{L}_{bc}(\theta) = \frac{1}{N}\sum_{k=1}^{N_{bc}} |C_\theta(x_{bc}^k, t_{bc}^k) - C_{bc}^k|^2, \quad \text{and} \tag{4-30}$$

$$\mathcal{L}_{ic}(\theta) = \frac{1}{N}\sum_{k=1}^{N_{ic}} |C_\theta(x_{ic}^k, t_{ic}^k) - C_{ic}^k|^2, \tag{4-31}$$

where $\{(x_{ic}^k, t_{ic}^k), C_{ic}^k\}_{k=1}^{N_{ic}}$, and $\{(x_{bc}^k, t_{bc}^k), C_{bc}^k\}_{k=1}^{N_{bc}}$ represent the training data on initial and boundary conditions while $\{x_{cp}^k, t_{cp}^k\}_{k=1}^{N_{cp}}$ denotes the collocation points for the residual, $\mathcal{N}_\theta(x_D, t_D)$, sampled randomly throughout the domain of interest.

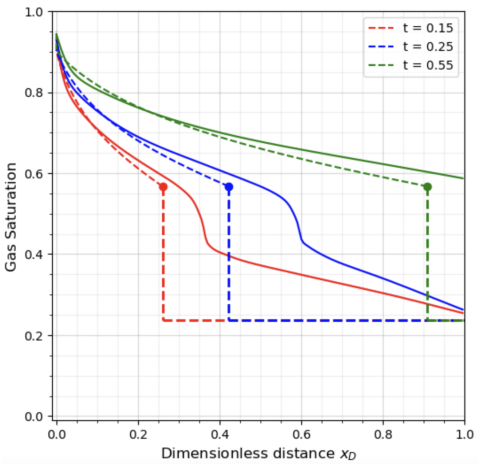

Figure 4.9: Prediction of the neural network in solid line vs the Analytical Solution by the method of characteristics in dashed line.

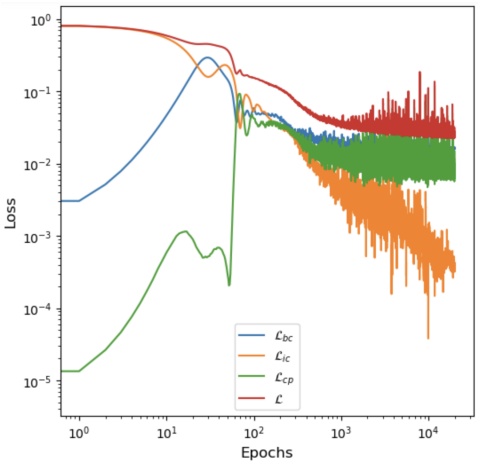

Figure 4.10: Evolution if the loss function in each training step. We can see the loss function $\mathcal{L}_{cp}, \mathcal{L}_{ic}, \mathcal{L}_{bc}$ and $\mathcal{L}$ that is the sum of all three.

In Figure 4.9, the neural network fails to catch the position of the shock in the solution. It predict a little behavior of the solution when the dimensionless

distance x_D is less than the position of the shock for each t_D. Besides that, is interesting that to x_D close to zero, as we talk before, exists a discontinuity in the solution due to the initial and boundary conditions, but the PINN tries to catch this discontinuity and shows a more "convex behavior" than the true solution.

As before, now we implement a diffusive term to Equation 4-27. The new residual function $\mathcal{N}$ is written as:

$$\mathcal{N}(x_D, t_D) = \frac{\partial C_\theta}{\partial t_D} + \frac{\partial F(C_\theta)}{\partial x_D} - \lambda \frac{\partial^2 C_\theta}{\partial x_D^2}, \tag{4-32}$$

where λ is a hyperparameter that we choose to be 0.003.

Figure 4.11 shows the results after adding this diffusive term. When compared to case without diffusion, the predictions match the true solution much better and the rate at which the shock moves seems to be more accurately matched with the addition of the diffusive term. In all three t_D that we tested the PINN predict the position of the shock very accurately. Also curiously in x_D closes to zero, the solution approaches the initial condition in all times. In general, we observe a greatly improvement in this specific case.

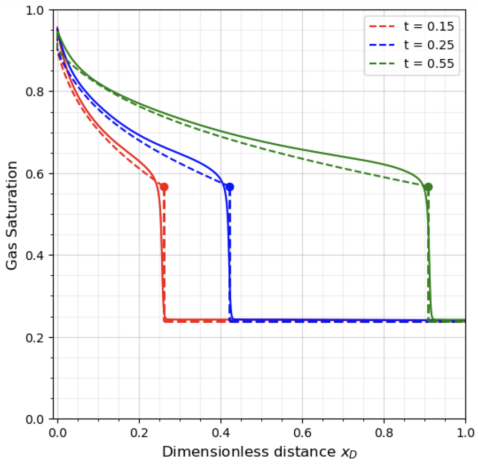

Figure 4.11: Prediction of the neural network with diffusion term. The solid line is the prediction of PINN and the Analytical Solution by the method of characteristics is in dashed line.

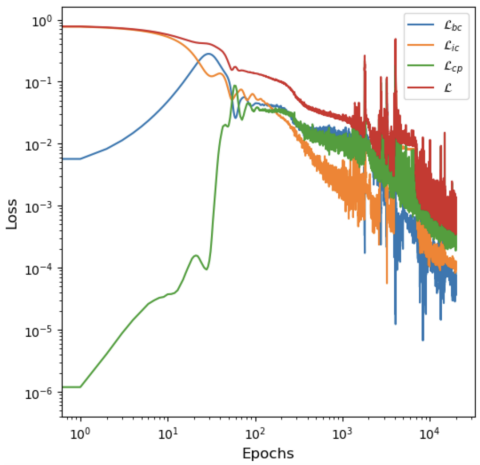

Figure 4.12: Evolution if the loss function in each training step. We can see the loss function $\mathcal{L}_{cp}, \mathcal{L}_{ic}, \mathcal{L}_{bc}$ and $\mathcal{L}$ that is the sum of all three.

Now we explore even more the limits of our neural network. We add the effects of volume change into the loss function, developed in chapter 2. To incorporate this phenomena, we have to dive in a more realistic case, where the components, and some properties of the medium are introduced.

4.4 Volume change Equations

The effect of pressure and temperature on the densities of liquids and solids is small. The compressibility for a typical liquid or solid is 10^{-6} bar^{-1} (1 bar = 0.1 MPa) and a typical thermal expansivity is 10^{-5} per Kelvin. That roughly means that is necessary around ten thousand times atmospheric pressure to reduce the volume of a typical liquid or solid just by one percent.

On the other hand, the density of gases is strongly affected by pressure. Each component has an specific molar density that varies with temperature and pressure by the following relation known as ideal gas law (SMITH et al., 2020):

$$\rho = \frac{MP}{RT}, \tag{4-33}$$

where M is the molar mass, P is the pressure, R is the universal gas constant, and T is the absolute temperature.

An ideal gas is a theoretical model in physics and chemistry which describes the simplified behavior of an gas in ideal conditions. From a realistic point of view there is a difference between a real gas and an ideal gas. This deviation, in literature, is called compressibility factor (Z). It is simply defined as the ratio of the molar volume of a gas to the molar volume of an ideal gas at the same temperature and pressure (SMITH et al., 2020):

$$Z = \frac{MP}{\rho RT}, \tag{4-34}$$

where, for an ideal gas, $Z = 1$ by definition.

At high pressures, molecules are colliding more often. This allows repulsive forces between molecules to have a noticeable effect, making the molar volume of the real gas greater than the molar volume of the corresponding ideal gas, which causes Z to exceed one. Similarly, when pressures are lower, the molecules are free to move. In this case, attractive forces dominate, making Z less than 1.

As $\rho = M/V$, we can substitute this relation in Equation 4-33 to get the following:

$$V = \frac{RT}{P}. \tag{4-35}$$

This also means that, if a gas displaces oil at high pressure, it occupies much less volume when dissolved in a liquid phase than it does in a vapor phase. In systems like this, the local flow velocity can vary substantially over the displacement length (DINDORUK, 1992; DUMORÉ; HAGOORT; RISSEEUW, 1984). Thus, for this gas displacement problem, it will be important to include the effects of volume change on mixing.

Fluid	x_{CO_2}	$x_{C_{10}}$	ρ (g/cm^3)
Initial Oil	0	1	0.6869
Equilibrium Oil	0.2733	0.7267	0.6810
Equilibrium Gas	0.9976	0.0024	0.0610
Injected Gas	1	0	0.0605

Table 4.5: Peng-Robinson Equilibrium Phase Compositions and Fluid Properties at 34 atm and 71 C.

To illustrate how volume change affects flow behavior, we need to calculate the molar density of the components to proceed with an example. So we consider the displacement of a hydrocarbon, decane (C_{10}), by a gas, carbon dioxide (CO_2), at 34 atm and 71 C. Table 4.5 reports the Peng-Robinson equilibrium phase compositions and the initial injection and phase molar densities.

In Equation 4-17, the local flow velocity is considered constant everywhere and equal to the injection velocity, so $v_D = 1$. Furthermore, the volume occupied by component i in one mole of phase j is x_{ij}/ρ_{ci}, and the volume fraction of component i in phase j is

$$c_{ij} = \frac{x_{ij}/\rho_{ci}}{\sum_{k=1}^{2} x_{kj}/\rho_{ck}}, \tag{4-36}$$

where here we assumed that each component has a constant molar density ρ_{ci}. Considering changing on those values, the Equation 4-17 changes to:

$$\frac{\partial}{\partial t}\sum_{j=1}^{2} x_{ij}\rho_{jD}S_j + \frac{\partial}{\partial x}v_D\sum_{j=1}^{2} x_{ij}\rho_{jD}f_j = 0. \tag{4-37}$$

The notation of Equation 4-37 can be simplified by defining two additional functions, G_i and H_i, as:

$$G_i = \sum_{j=1}^{2} x_{ij}\rho_{jD}S_j, \quad \text{and} \tag{4-38}$$

$$H_i = v_D\sum_{j=1}^{2} x_{ij}\rho_{jD}f_j, \tag{4-39}$$

where G_i is an overall concentration (in moles per unit volume) of component i and H_i is an overall molar flow of component i. Finally the multicomponent equations for multiphase convection is, therefore,

$$\frac{\partial G_i}{\partial t_D} + \frac{\partial H_i}{\partial x_D} = 0, \qquad i = 1, 2, \tag{4-40}$$

The phase compositions, viscosities and moles are fixed, so phase relative

permeabilities (Equations 4-19 and 4-20) are functions of saturation only. Rearranging the Equation 4-38 we have:

$$S_i = \frac{G_i - x_{i2}\rho_{2D}}{x_{i1}\rho_{1D} - x_{i2}\rho_{2D}}, \tag{4-41}$$

The characteristic equations can now be obtained just as they were in the previous example Equations 4-26. Arguments similar to those given indicate that H_1 is a function of G_1 only, and hence,

$$\frac{\partial G_1}{\partial t} + \frac{dH_1}{dG_1}\frac{\partial G_1}{\partial x} = 0, \tag{4-42}$$

considering $x(\eta)$ and $t(\eta)$ we have:

$$\frac{dG_1}{d\eta} = \frac{\partial G_1}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial G_1}{\partial t}\frac{\partial t}{\partial \eta}, \tag{4-43}$$

and therefore, with Equation 4-42 and Equation 4-43 we have the characteristic equations:

$$\begin{aligned} \frac{dG_1}{d\eta} &= 0, \\ \frac{dt}{d\eta} &= 1, \\ \frac{dx}{d\eta} &= \frac{dH_1}{dG_1}. \end{aligned} \tag{4-44}$$

Differentiation of Equation 4-39 gives:

$$\frac{\partial H_1}{\partial G_1} = v_D(x_{11}\rho_{1D} - x_{12}\rho_{2D})\frac{\partial f_1}{\partial G_1}, \tag{4-45}$$

and substituting Equation 4-41 we have:

$$\frac{\partial f_1}{\partial G_1} = \frac{\partial f_1}{\partial S_1}\frac{\partial S_1}{\partial G_1} = \frac{1}{x_{11}\rho_{1D} - x_{12}\rho_{2D}}\frac{\partial f_1}{\partial S_1}. \tag{4-46}$$

Therefore, we have the relation:

$$\frac{\partial H_1}{\partial G_1} = v_D\frac{\partial f_1}{\partial S_1}. \tag{4-47}$$

Hence for compositions within the two-phase region, the wave velocity is the wave velocity for constant volume flow scaled by the appropriate local flow velocity within the two-phase region.

The distinction between *flow* velocity and *wave* velocity is an important one. The flow velocity is the total volumetric flow rate of all the phases per unit area. The wave velocity is the speed at which a given composition propagates. When volume is not conserved, the flow velocity does not change when the composition varies along a single tie line, but it does change at shocks that enter

or leave the two-phase region. In other words, there are three flow velocities: The injection flow velocity (v_{inj}) which is given, changes instantaneously when the flow enters in the reservoir; this flow velocity in the two-phase region remains constant; and a different flow velocity appears ahead of the leading shock. The calculations of these velocities are outside of the scope of this work. But we can expect a decreasing flow velocity. As a result, C_{10} is recovered much more slowly after the arrival of the leading shock at the outlet.

As before, lets defining the residual function $\mathcal{N}_\theta(x_D, t_D)$. Unlike the previous cases, this time we have two PDEs to handle. This means that to each pair (x_D, t_D) we have a vector defined as:

$$\mathcal{N}_\theta(x_D, t_D) = \left(\frac{\partial G_1}{\partial t_D} + \frac{\partial H_1}{\partial x_D}, \frac{\partial G_2}{\partial t_D} + \frac{\partial H_2}{\partial x_D}\right), \tag{4-48}$$

where the neural network solution S_θ parametrized by θ will learn the parameters so we can use all constants to get G_1^θ and G_2^θ.

Despite $\mathcal{N}_\theta(x_D, t_D)$ being a vector, the loss function $\mathcal{L}$ is composed in the same way as before:

$$\mathcal{L}(\theta) = \mathcal{L}_{cp}(\theta) + \mathcal{L}_{bc}(\theta) + \mathcal{L}_{ic}(\theta), \tag{4-49}$$

where $\mathcal{L}_b(\theta)$ is the mean squared error from the boundary conditions, $\mathcal{L}_i(\theta)$ is the mean squared error from the initial conditions and $\mathcal{L}_f(\theta)$ is the mean squared error from the residual evaluated at collocation points:

$$\mathcal{L}_{cp}(\theta) = \frac{1}{N}\sum_{k=1}^{N_{cp}} \|\mathcal{N}_\theta(x_{cp}^k, t_{cp}^k)\|^2, \tag{4-50}$$

$$\mathcal{L}_{bc}(\theta) = \frac{1}{N}\sum_{k=1}^{N_{bc}} \| \left(G_1^\theta(x_{bc}^k, t_{bc}^k) - G_{1,bc}^k, G_2^\theta(x_{bc}^k, t_{bc}^k) - G_{2,bc}^k\right) \|^2, \tag{4-51}$$

$$\mathcal{L}_{ic}(\theta) = \frac{1}{N}\sum_{k=1}^{N_{ic}} \| \left(G_1^\theta(x_{ic}^k, t_{ic}^k) - G_{1,ic}^k, G_2^\theta(x_{ic}^k, t_{ic}^k) - G_{2,ic}^k\right) \|^2, \tag{4-52}$$

where $\{(x_{ic}^k, t_{ic}^k), G_{1,ic}^k, G_{2,ic}^k\}_{k=1}^{N_{ic}}$, and $\{(x_{bc}^k, t_{bc}^k), G_{1,bc}^k, G_{2,bc}^k\}_{k=1}^{N_{bc}}$ represent the training data on initial and boundary conditions while $\{x_{cp}^k, t_{cp}^k\}_{k=1}^{N_{cp}}$ denotes the collocation points for the residual, $\mathcal{N}_\theta(x_D, t_D)$, sampled randomly throughout the domain of interest.

In this case, before train our neural network, we implement the diffusive term into Equation 4-48. The residual function turns into:

$$\mathcal{N}_\theta(x_D, t_D) = \left(\frac{\partial G_1}{\partial t_D} + \frac{\partial H_1}{\partial x_D} - \lambda\frac{\partial^2 G_1}{\partial x_D^2}, \frac{\partial G_2}{\partial t_D} + \frac{\partial H_2}{\partial x_D} - \lambda\frac{\partial^2 G_2}{\partial x_D^2}\right). \tag{4-53}$$

Figure 4.13 shows the comparison between the PINN solution and the

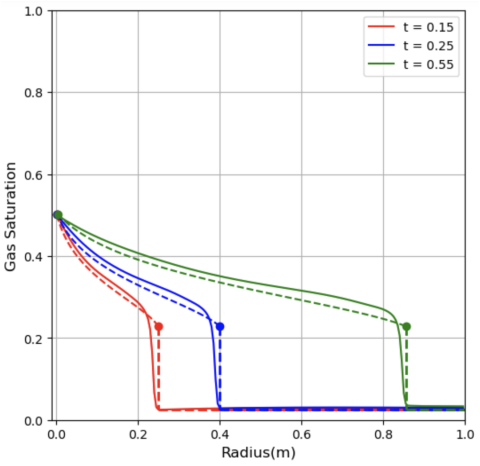

Figure 4.13: Prediction of the neural network of the Volume Change equations in solid line versus the analytical solution by MOC in dashed line. In this case, the fractional flow curve was assumed to be Equations 4-22, with $S_{or} = S_{gc} = 0$.

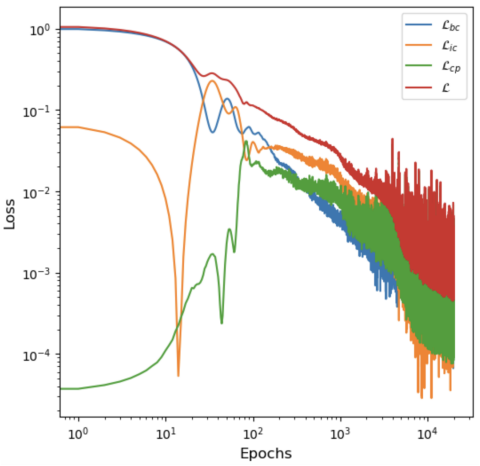

Figure 4.14: Evolution of $\mathcal{L}_{cp}$, $\mathcal{L}_{ic}$, $\mathcal{L}_{bc}$ and $\mathcal{L}$ in each training step.

analytical obtained by MOC. As previous cases, the neural network give us consistent results. The initial and boundary conditions are seen in the solution as well as the position of the shock. In this training process, the minimal value of the loss function we get was 10^{-3}.

Using the PINN architecture proposed in this study, we observed similar loss values to the ones obtained by the study of (FUKS; TCHELEPI, 2020). This is a great improvement since they use observed data to improve their training process and we do not. Figure 4.4 shows the evolution of the loss function in each training step when we do not include the diffusive term. Figure 4.6 shows the same evolution but with the diffusive term included.

To the second case, the implementation of the diffusive term is of a big importance to improve the results of the PINN solution. A we can see in Figure 4.12, we could get an an loss value orders of magnitude better than the solution without this term (Figure 4.10).

Based in those cases, we observe that the diffusive term λ is crucial for the convergence of the PINN solution. Then, so we implement this term without testing without it before for the problem. In Figure 4.14 we can see the evolution of each term of the loss function in this case.

In the next chapter, we will discuss what we found and concluded from this study. We will discuss our expectations and accomplishments. Then, we will also outline our plans for future research efforts.

5
Conclusion

The results in (DIAB; AL-KOBAISI, 2021) are surprising. However, the authors added a wrong diffusion term (1×10^{-3}) instead of the one suggested by (FUKS; TCHELEPI, 2020) (not less than 2.5×10^{-3}), and this can be a reason to not get an optimal convergency. They also incorporated observational data into the PINN. This allows the network predict the behavior of the solution very effectively.

Motivated by problems like those seen in (RAISSI, 2017b), we remove the external data and construct a neural network to solve the equation given in (ALMAJID; ABU-ALSAUD, 2020) with only initial and boundary conditions. We get losses similar to the ones achieved by the authors.

Using an approach called chromatography presented by (ORR, 2007), we implemented more realistic gas injection processes in reservoirs. This study leads us to the second case of this work. At first we do not implement the volume change equations, just to see the potential of the architecture proposed in this work. We observe a great improvement of the training just observing the evolution of the loss functions. In addition to decreasing the variance of the values of the loss function, we reduced these values by orders of magnitude.

By implementing the volume change equations, which transformed the residual function into a vector function, we obtained a result very close to that described in (ORR, 2007). Based on previous cases, we note that the diffusion term was necessary to the PINN solution converge to the true solution, so we implement this term at first time, and get promising results.

For future work, some enhancements may be implemented to the PINN approach. In (COUTINHO et al., 2023), the authors implement the parameter λ as trainable in the neural network. Their study proposes three methods to add this term, called artificial viscosity. This method seems to be appropriated for solving hyperbolic PDEs with shocks using PINN.

Another topic to be implemented in the training process is described in (WANG; SANKARAN; PERDIKARIS, 2022). They reformulate the loss function as follows

$$\mathcal{L}(\theta) = \omega_{cp}\mathcal{L}_{cp}(\theta) + \omega_{bc}\mathcal{L}_{bc}(\theta) + \omega_{ic}\mathcal{L}_{ic}(\theta),$$

where ω_{cp}, ω_{bc}, and ω_{ic} are hyperparameters that allow the flexibility of assigning a different learning rate to each loss term to balance their interplay during model training. They show that this implementation works for different

types of PDEs, increasing cost but improving learning.

In actual field-scale gas injection projects, the flow is never one-dimensional. Therefore, many additional factors influence the performance of these multidimensional flows: viscous instability, gravity segregation, reservoir heterogeneity, and crossflow due to viscous and capillary forces. All of these factors can be introduced into the equations to provide more realistic solutions (ORR, 2007).

6
Bibliography

ABADI, M. et al. TensorFlow: A system for large-scale machine learning. p. 265–283, 2016.

ALMAJID, M. M.; ABU-ALSAUD, M. O. Prediction of fluid flow in porous media using physics informed neural networks. **Society of Petroleum Engineers**, 2020.

AYERS, R.; YAO, L.; URBAN, M. A. Environmental and economic implications of shale gas development strategies: Analysis of china's national carbon emissions. **Applied Energy**, Elsevier, v. 276, p. 115462, 2020.

BALDINI, M.; BORSI, I.; FALCHETTI, A. Drivers of renewable energy: A bibliometric review. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 109, p. 473–490, 2019.

BATCHELOR, G. K. **An Introduction to Fluid Dynamics**. [S.l.]: Cambridge University Press, 1967.

BEAR, J. **Dynamics of Fluids in Porous Media**. [S.l.]: American Elsevier Pub. Co., 1972.

BEAR, J. **Dynamics of Fluids in Porous Media**. [S.l.]: Dover Publications, 2013.

BIRD, R. B.; STEWART, W. E.; LIGHTFOOT, E. N. **Transport Phenomena**. [S.l.]: John Wiley & Sons, 2007.

BROOKS, R. H.; COREY, A. T. Hydraulic properties of porous media. **Hydrology Paper**, v. 3, n. 1, p. 27–28, 1964.

BUCKLEY, S. E.; LEVERETT, M. C. Mechanism of fluid displacement in sands. **Transactions of the AIME**, Society of Petroleum Engineers, v. 146, n. 01, p. 107–116, 1942.

CHEN, Z. **Reservoir Simulation: Mathematical Techniques in Oil Recovery**. [S.l.]: Society for Industrial and Applied Mathematics, 2016.

COREY, A. T. The interrelation between gas and oil relative permeabilities. **Production Monthly**, p. 19–23, 1954.

COREY, A. T. The hydraulic properties of porous media. **Publications in Hydrology**, v. 3, n. 3, p. 33–42, 1966.

COUTINHO, E. J. R. et al. Physics-informed neural networks with adaptive localized artificial viscosity. **Journal of Computational Physics**, v. 489, p. 112265, 2023. ISSN 0021-9991. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0021999123003601>.

CRANDALL, M. G.; LIONS, P. L. Viscosity solutions of hamilton-jacobi equations. **Transactions of the American Mathematical Society**, American Mathematical Society, v. 277, n. 1, p. 1–42, 1983.

DARCY, H. Les fontaines publiques de la ville de dijon. **Victor Dalmont, Paris**, 1856.

DIAB, W.; AL-KOBAISI, M. PINNs for the solution of the hyperbolic Buckley-Leverett problem with a non-convex flux function. 2021.

DINDORUK, B. Analytical theory of multiphase, multicomponent displacement in porous media. Stanford, CA, 1992.

DOE, J.; SMITH, J. Physics-informed neural networks for fluid flow predictions: A comprehensive review. **Journal of Computational Fluid Dynamics**, 2022.

DUMORÉ, J.; HAGOORT, J.; RISSEEUW, A. An analytical model for one-dimensional, three-component condensing and vaporizing gas drives. **Soc. Pet. Eng. J.**, v. 24, p. 169–179, April 1984.

FUKS, O.; TCHELEPI, H. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. 07 2020.

GIBBS, J. W. On the equilibrium of heterogeneous substances (parts i and ii). **Transactions of the Connecticut Academy of Arts and Sciences**, v. 3, p. 108–248, 1878.

HELFFERICH, F.; KLEIN, G. **Multicomponent Chromatography**. New York: Marcel Dekker Publishing Co., 1970.

JOHNSON, M. et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. **Transactions of the Association for Computational Linguistics**, v. 5, p. 339–351, 2017.

LU, Y.; MAO, Z.; LING, J. Physics-informed neural networks for shape optimization. **Computer Methods in Applied Mechanics and Engineering**, v. 363, p. 112892, 2020.

MEER, R. van der; OOSTERLEE, C. W.; BOROVYKH, A. Optimally weighted loss functions for solving pdes with neural networks. **Journal of Computational and Applied Mathematics**, v. 405, 2022.

ORR, J. F. M. **Theory of gas injection processes**. Holte, Denmark: Tie-Line Publications, 2007.

PASZKE, A. et al. PyTorch: An imperative style, high-performance deep learning library. **Advances in Neural Information Processing Systems (NeurIPS)**, v. 32, 2019. NeurIPS.

RAISSI, M. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. **arXiv preprint arXiv:1711.10561**, 2017.

RAISSI, M. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. **arXiv preprint arXiv:1711.10566**, 2017.

RHEE, H.; ARIS, R.; AMUNDSON, N. **First-Order Partial Differential Equations: Volume I**. Englewood Cliffs, NJ: Prentice-Hall, 1986.

RHEE, H.; ARIS, R.; AMUNDSON, N. **First-Order Partial Differential Equations: Volume II**. Englewood Cliffs, NJ: Prentice-Hall, 1989.

RHEE, H. K.; ARIS, R.; AMUNDSON, N. R. On the theory of multicomponent chromatography. **Phil. Trans. Roy. Soc. London**, v. 267 A, p. 419–455, 1970.

SENNRICH, R.; HADDOW, B.; BIRCH, A. Improving neural machine translation models with monolingual data. **arXiv preprint arXiv:1606.06122**, 2016.

SERRANO, W. A big data intelligent search assistant based on the random neural network. **INNS Conference on Big Data.**, 2016.

SIRIGNANO, J.; SPILIOPOULOS, K. Dgm: A deep learning algorithm for solving partial differential equations. **Journal of Computational Physics**, Elsevier, v. 375, p. 1339–1364, 2018.

SMITH, J. M. et al. **Chemical Engineering Thermodynamics**. [S.l.]: McGraw-Hill Education, 2020.

STEVENS, E.; ANTIGA, L.; VIEHMANN, T. **Deep Learning with PyTorch**. New York, NY, USA: Manning Publications, 2020.

TRIPATHY, S. et al. Physics-informed neural networks for solving inverse problems in imaging. **SIAM Journal on Imaging Sciences**, v. 14, n. 3, p. 1480–1504, 2021.

VASWANI, A. et al. Attention is all you need. p. 5998–6008, 2017.

VERGANTI L. VENDRAMINELLI, M. I. R. Innovation and design in the age of artificial intelligence. **Journal of Product Innovation Management**, 2020.

WAALS, J. van der. **On the Continuity of the Gaseous and Liquid States**. [S.l.]: North-Holland Physics Publishing, 1988. 83–140 p.

WADAWADAGI, V. Pytorch vs tensorflow: Deep learning frameworks [2023]. **PyTorch vs TensorFlow: Deep Learning Frameworks [2023]**, 2023. Available from Internet: <https://www.knowledgehut.com/blog/data-science/pytorch-vs-tensorflow#difference-between%C2%A0tensorflow%C2%A0and%C2%A0pytorch%C2%A0>.

WANG, Q. et al. Physics-informed neural networks for solving elasticity equations. **Computer Methods in Applied Mechanics and Engineering**, v. 384, p. 113879, 2021.

WANG, S.; SANKARAN, S.; PERDIKARIS, P. Respecting causality is all you need for training physics-informed neural networks. **arXiv preprint arXiv:2203.07404**, 2022.

ÖKTEM, O.; BERG, E. van den; HANSEN, P. C. Deep learning for sparse small-data problems: Applications to electromagnetic inverse problems. **Inverse Problems**, IOP Publishing, v. 34, n. 12, p. 124004, 2018.

A
PINN architecture