COPPE
UFRJ
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia

DETERMINING PERMEABILITY THROUGH DEEP NEURAL NETWORKS TRAINED WITH LATTICE BOLTZMANN DATA

João Ricardo Friggo e Barros

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Química, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Química.

Orientadores: Frederico Wanderley Tavares
Pedro Henrique Constantino

Rio de Janeiro
Março de 2024

DETERMINING PERMEABILITY THROUGH DEEP NEURAL NETWORKS TRAINED WITH LATTICE BOLTZMANN DATA

João Ricardo Friggo e Barros

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA QUÍMICA.

Orientadores: Frederico Wanderley Tavares
Pedro Henrique Constantino

Aprovada por: Prof. Gabriel Gonçalves da Silva Ferreira
Prof. Paulo Couto

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2024

Friggo e Barros, João Ricardo

Determining permeability through deep neural networks trained with lattice Boltzmann data/João Ricardo Friggo e Barros. – Rio de Janeiro: UFRJ/COPPE, 2024.

XVII, 73 p.: il.; $29,7$cm.

Orientadores: Frederico Wanderley Tavares
Pedro Henrique Constantino

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Química, 2024.

Referências Bibliográficas: p. 62 – 73.

1. Porous Media. 2. Lattice Boltzmann Method. 3. Convolutional Neural Networks. I. Tavares, Frederico Wanderley *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Química. III. Título.

"Trust the simulation,
don't you let it break"
Troye Sivan

Agradecimentos

Gostaria de agradecer a todos que compartilharam dessa experiência comigo e tornaram este trabalho possível.

Em primeiro lugar, aos diretamente envolvidos no trabalho, Ramon Lourenço e Marlon Gama, profissionais de capacidade ímpar que me deram suporte e me ensinaram muito ao longo de todo o processo de desenvolvimento deste texto, bem como aos meus orientadores Frederico Tavares e Pedro Constantino, que me desafiaram e acreditaram em mim para superar os desafios impostos.

Em segundo lugar, gostaria de agradecer a todo o laboratório ATOMS, incontáveis pessoas que tornaram meu dias melhores e viveram comigo grande parte das emoções da jornada que é fazer uma pós graduação. Um agradecimento especial também a todos os meus amigos do PEQ, que fizeram parte dessa caminhada desde o primeiro dia.

Por fim, gostaria de agradecer a minha família, que me deu o suporte necessário para que eu pudesse chegar aqui, sem nunca deixar de acreditar em mim. Uma menção especial ao meu namorado, Bruno, que segurou muitas das barras comigo e não me deixou desistir em nenhum momento.

A todos, meu muito obrigado. Esse trabalho também é de vocês.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

DETERMINING PERMEABILITY THROUGH DEEP NEURAL NETWORKS TRAINED WITH LATTICE BOLTZMANN DATA

João Ricardo Friggo e Barros

Março/2024

Orientadores: Frederico Wanderley Tavares
Pedro Henrique Constantino

Programa: Engenharia Química

Nos últimos anos, os avanços tecnológicos têm melhorado significativamente nossa capacidade de entender materiais porosos, focando em uma propriedade crucial conhecida como permeabilidade, que dita o fluxo de fluidos através desses meios. Inovações como tomografias de rochas altamente detalhadas e modelos de inteligência artificial sofisticados revolucionaram a determinação da permeabilidade. Ao aproveitar essas ferramentas, os pesquisadores podem analisar geometrias porosas complexas com uma precisão sem precedentes. No entanto, há a necessidade de conjuntos de dados extensos e confiáveis para treinar esses modelos de inteligência artificial de forma eficaz. Tradicionalmente, a obtenção de dados de fluxo em meios porosos requer experimentos caros e demorados ou simulações de dinâmica dos fluidos computacional (CFD). No entanto, métodos CFD convencionais frequentemente dependem de modelos simplificados de redes de poros, limitando sua aplicabilidade a estruturas reais de rochas observadas em tomografias. O método lattice-Boltzmann (LBM), uma abordagem alternativa enraizada na mecânica estatística e na teoria cinética, opera no nível mesoscópico e simula diretamente o comportamento do fluido dentro de materiais porosos, contornando a necessidade de aproximações de redes de poros. Utilizando o operador de colisão MRT para melhorar a precisão e acurácia dos resultados, um esquema LBM é proposto, validado por meio de simulações do escoamento Poiseuille e observância à lei de Darcy, reproduzindo corretamente o comportamento previsto pela equação de Navier-Stokes. Estruturas de rochas artificialmente geradas são empregadas para as simulações diretas e assim gerar um banco de dados abrangente correlacionando permeabilidades, tortuosidades, porosidades e áreas superficiais com o objetivo de treinar modelos de inteligência artificial. Notavelmente, a rede supera equações tradicionais, como a equação de Kozeny-Carman, demonstrando seu potencial como uma ferramenta confiável para caracterizar meios porosos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

DETERMINING PERMEABILITY THROUGH DEEP NEURAL NETWORKS TRAINED WITH LATTICE BOLTZMANN DATA

João Ricardo Friggo e Barros

March/2024

Advisors: Frederico Wanderley Tavares
Pedro Henrique Constantino

Department: Chemical Engineering

In recent years, technological advances have significantly enhanced our ability to understand porous materials, focusing on a crucial property known as permeability, which dictates the flow of fluids through these mediums. Innovations such as highly detailed rock tomographies and sophisticated artificial intelligence models have revolutionized permeability determination. By harnessing these tools, researchers can analyze complex porous geometries with unprecedented precision. However, there is a need for extensive and reliable datasets to train these artificial intelligence models effectively. Traditionally, obtaining flow data in porous media requires expensive and time-consuming experiments or computational fluid dynamics (CFD) simulations. However, conventional CFD methods often rely on simplified models of pore networks, limiting their applicability to real rock structures observed in tomographies. The lattice-Boltzmann method (LBM), an alternative approach rooted in statistical mechanics and kinetic theory, operates at the mesoscopic level and directly simulates fluid behavior within porous materials, circumventing the need for pore network approximations. Using the multiple-relaxation-time (MRT) collision operator to improve the accuracy and precision of results, an LBM scheme is proposed, validated through Poiseuille flow simulations and adherence to Darcy's law, correctly reproducing the behavior predicted by the Navier-Stokes equation. Artificially generated rock structures are employed for direct simulations to generate a comprehensive database correlating permeabilities, tortuosities, porosities, and surface areas to train artificial intelligence models. Notably, the network outperforms traditional equations, such as the Kozeny-Carman equation, demonstrating its potential as a reliable tool for characterizing porous media.

Contents

List of Figures

List of Tables

List of Symbols

$\boldsymbol{\xi}$	Microscopic velocities of particles
$\mathcal{H}$	$\mathcal{H}$ quantity
$\mathbf{F}$	Force
$\mathbf{h}$	Hamiltonian in phase space
$\mathbf{H}(x)$	Hermite polynomials generating function
$\mathbf{M}$	MRT Matrix
$\mathbf{p}$	Momentum in phase space
$\mathbf{q}$	Position in phase space
$\mathbf{S}$	Relaxation times vector
$\mathbf{u}$	Macroscopic velocity
$\mathbf{V}$	Relative microscopic velocity
$\mathbf{x}$	System's coordinates in Cartesian space
μ	Dynamic viscosity
ν	Kinematic viscosity
Ω	Collision operator
ρ	Macroscopic density
τ	Relaxation time
$a^{(n)}$	Generalized series coefficient
c_i	Discrete velocities
c_s^2	Sound speed

E	Total energy
f	Distribution function
$f*$	Post collision distributions
f^{eq}	Distribution function in equilibrium
f_N	Probability distribution function
m	Mass
N	Number of particles in a system
P	Pressure
t	Time
u_x	Velocity along the x axis
u_y	Velocity along the y axis
$w(x)$	Hermite weight function
w_i	Discrete weights
x	X axis
y	Y axis
a	Distance between two plates in Poiseuille flow

Acronyms

BTE Boltzmann Transport Equation

CCUS Carbon Capture Usage and Storage

CFD Computational Fluid Dynamics

FHP Frisch Hasslacher and Pomeau

GRI Core analysis method by the Gas Research Institute

HPP Hardy Pomeau and Pazzis

LBE Lattice Boltzmann Equation

LBM Lattice Boltzmann Method

LGA Lattice Gas Automata

MICP Mercury Injection Capillarity Pressure

NMR Nuclear Magnetic Ressonance

REV Representative Element Volume

SEM Scanning Electron Microscopy

Chapter 1

Introduction

1.1 Motivation

Reducing carbon emissions and dealing with their consequences are one of the most significant challenges of the 21st century. Since 2005 the world has emitted more than 30 billion tons of CO_2 each year [1]. However, it is estimated that if most of the countries adopt net-zero targets, it is still possible to limit the warming of the planet in $2.4 - 2.9$ °C [2]. To reach that, many strategies must be implemented to reduce emissions without compromising industry and economic growth, besides dealing with the climate changes associated with the carbon already in the atmosphere.

A combination of strategies that have emerged over the last years as a tool to reduce the impact of the use of fossil fuels for power generation and industry processes that are carbon intensive is Carbon Capture Usage and Storage (CCUS). CCUS strategies are divided in CO_2 capture, CO_2 looping, CO_2 transport, CO_2 utilization and CO_2 storage [3].

Carbon storage can be divided into forest carbon storage [4], soil carbon storage [5], and geological carbon storage [6]. Geological carbon storage is a hot topic in engineering sciences for its capabilities of storing large amounts of CO_2 replicating the carbon cycle [7] [8] [9].

It is necessary to characterize the geology underneath to calculate the amount of carbon that can be stored at a specific reservoir. Reservoir rocks are necessarily porous, as they store fluids in rock voids. Using mathematical models for porous media provides the tools to address this problem and numerically describe different reservoirs.

Therefore, permeability prediction is essential in the context of CCUS to understand fluid flow under these conditions. This study aims to analyze and understand its measurement, explicitly focusing on the numerical estimation of permeability through the lattice Boltzmann method. Then, another approach is proposed, to predict permeability from rock images using neural networks and data correlation.

1.2 Porous media

Porous media are everywhere in nature. Adler [10] divides them into at least three categories: Artificial, Biological, and Geological porous media. Artificial media, such as a group of packed spheres or cylindrical tubes, are helpful for models and theoretical investigation. They also play a role in materials created by man, such as fabrics, catalysts, and cement. Biological media are often found due to transport processes needed to sustain life, relying on interfacial surfaces to exchange heat and mass. It is the case of the liver, the lungs, and the circulatory system. Geological media have been extensively studied for their importance to the oil and gas industry and groundwater flow applications. Most of the terminology used in studying porous media comes from geology, such as throats, fractures, and grains.

1.2.1 Porosity

Consider a medium composed of two phases, L and S (Generally associated with liquid and solid, but can also be seen as void and solid or any two phases in contact), it is possible to write a phase function as:

$$X_K(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \text{phase } K \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

Where $K = L, S$, in such a way that considering the interface negligible, in any point at position $\mathbf{x}$:

$$X_L(\mathbf{x}) + X_S(\mathbf{x}) = 1 \tag{1.2}$$

This expression is equivalent to saying that there is no point in this phase space that is in both phases, as for any $\mathbf{x}$ the sum must be 1. The porosity ϕ of such a medium can be calculated through a weighted integral in respect to the volume of the medium:

$$\phi = \frac{1}{V} \int_V X_L(\mathbf{x})\ d^3\mathbf{x} \tag{1.3}$$

Therefore, porosity may be seen as the fraction of the liquid phase inside of the domain, tending to zero when the solid phase is more abundant than the liquid phase, and to unity otherwise. To imagine a medium with $\phi = 1$ is simply a block of fluid, without any obstacle or resistance to flow.

Approximate porosity values for rocks are shown in table 1.1 [11].

Material	Porosity (%)
Limestone	15
Sandstone	20
Fritted glass	35
Semi-quartzitic sandstone	4
Chalk	33

Table 1.1: Typical porosity values for common rocks

Porosity may be the most critical property of porous media. The porosity of a material strongly affects it properties, both structural and functional, such as elasticity, ultimate strength, thermal and electric conductivities and magnetization. Many mathematical models have been proposed to study their relation with porosity, but they are restricted to some cases due to how the connectivity of the pores and their morphology affects the material [12].

As porosity is a multi-scale phenomenon, many methods are available to measure a medium's porosity. The most common method that covers the significant range of porosities is mercury intrusion, capable of measuring pores in the scale of $10nm$ to $0.1mm$. Other standard methods are gas adsorption, x-ray scattering, electron microscopy, thermoporosimetry, fluid flow, light microscope image analysis, and x-ray tomography, even though these methods work on a shorter range of scales than mercury intrusion [13].

1.2.2 Tortuosity

The hydraulic tortuosity of a path may be calculated as the squared ratio of the shortest path between two points and the actual path between them (Equation 1.4).

$$T_H = \left(\frac{L_e}{L}\right)^2 \tag{1.4}$$

Tortuosity directly impacts electric and hydraulic permeability [14]. However, due to the diversity of concepts regarding tortuosity, the literature has failed to unify its definition. Percolation theory has been vastly used to unify tortuosity, but it still lacks accuracy when near the limit of $\phi = 1$. However, Blunt (2017) [15] notes that is not precisely tortuosity that limits the flow, but dead ends in unconnected throats.

In this text, the word "Tortuosity" means the hydraulic tortuosity of a medium, as the simulations carried through the text are all single-phase fluid flow experiments. It is important to remark that tortuosity is an axis-dependent property, which means it is only defined with respect to the direction of flow. For a single porous media, it is possible to have different tortuosities related to the pair of boundaries chosen to perform the calculations.

1.2.3 Specific surface area

The specific surface area is the last of the main characterizing parameters treated in this text. It is defined as the surface area per unit of volume in a given porous domain. It is especially essential when the interface takes a critical role, as in the study of wettability, chemical reactions over solid catalysts and absorption.

Mathematically, the specific surface area is given as a ratio between the surface area (S_{area}) and the volume:

$$A = \frac{S_{area}}{V} \tag{1.5}$$

Generally, adsorption experiments characterize surfaces and find the specific surface area of a given material [16]. This is done through the deposition of an inert gas such as nitrogen over the surface of a material, then the desorption is done to measure the exact amount of gas adsorpted. Then, by estimating the radius of the gas molecules, the superficial area for a given volume can be calculated.

1.2.4 Permeability

Permeability (k) was first defined by the French engineer Henry Darcy in 1856 in his works over public fountains in Dijon [17]. He noticed that the relationship between the pressure gradient and the velocity at which fluids pass through a rock was proportional to a constant at laminar flow. The laminar regime for flow through porous media, known as the Darcy hypothesis, is generally valid for $Re < 1$ [18]. If it is valid, the relation known as Darcy's law is written initially as:

$$q = -k\frac{h_2 - h_1}{L} \tag{1.6}$$

Where q is the water flow rate, h_1 and h_2 are the heights above a reference level,L is the length of the medium in the direction of flow, and k is a constant named permeability. It is a purely geometrical property that denotes the ability of a medium to allow fluid flow through it. Even though Darcy discovered his law empirically, it may be directly derived from Navier-Stokes Equations [19] [20]. A more generic way of writing Darcy's Law for any fluid is given as:

$$u_x = k\frac{\rho}{\mu}\nabla P \tag{1.7}$$

Each material will have a different permeability value because it is a property of the medium in which the flow occurs. Permeability unit in the International System is m^2, but for most practical scenarios, it is in the order of $10^{-12}m^2$, or $1\mu m^2$, defined as one Darcy, or $1D$. As this study focuses only on the permeability of consolidated porous media, typical permeability values for consolidated rocks are presented in table 1.2 according to Woessner and Poeter [21].

Rock type	Permeability (Darcys)
Karst limestone	$10^5 - 10^{-1}$
Permeable basalt	$10^3 - 10^{-2}$
Fractured igneous and metamorphic rocks	$10^0 - 10^{-4}$
Limestone/Dolostone	$10^{-1} - 10^{-4}$
Sandstone	$10^{-1} - 10^{-5}$
Unfractured igneous and metamorphic rocks	$10^{-5} - 10^{-9}$

Table 1.2: Permeability values for common consolidated rocks

As permeability is a critical property to understand fluid flow, different methodologies for predicting it exist. A classical way of estimating the permeability of rocks is through the Kozeny-Carman equation [22]:

$$k = \frac{F\phi^3}{A^2(1-\phi^2)} \tag{1.8}$$

Where ϕ is the porosity of the medium, s is its specific surface area, and F is a fitting parameter. This equation has been extensively used in the determination of porosity, although it has been shown that it may overestimate permeability values as heterogeneity of the medium increases [23].

The multi-scale nature of permeability determination presents a significant challenge, as the porous morphology of a material can vary across different scales, including nano, micro, meso, and macroscales. Consequently, various characterization methods are appropriate for different scales, leading to differing values of porosity, tortuosity, and permeability depending on the level of magnification applied to the problem.

1.3 Experimental determination

The traditional method for determining permeability is the core-flooding experiment. The core-flooding experiment consists of passing a fluid through a porous rock sample trapped in a permeable membrane, generally with the aid of a Permeameter, precisely measuring the fluid velocity its the associated pressure drop at the stationary state [24]. These measurements are habitual in the oil and gas industry, as core-flooding can predict not only absolute but relative permeability of a medium concerning a particular component [25].

However, experimental challenges, such as fractures in the samples due to the high pressures and different flow regimes from laminar to turbulent, may cause inaccuracies in permeability values and limit the application of this technique [26]. Also, reservoir rocks might not be homogeneous and evenly distributed, so the samples analyzed may not always represent the whole medium. Finally, core-flooding is generally limited to a particular core scale, below which permeability cannot be predicted [25].

Other experimental methods for determining porosity, permeability, and network parameters include Mercury Injection Capillarity Pressure (MICP), Nuclear Magnetic Ressonance (NMR), Core analysis method by the Gas Research Institute (GRI), and pulse decay.

MICP involves injecting mercury and plotting a capillarity pressure curve that represents mercury wetting at the rock, and by integrating this curve it is possible to recover the permeability of the porous sample. It can recover permeability in an extensive range of scales from the micropores to the mesopores but fails at filling the nanopores [27].

NMR induces a spin polarization of proton in hydrogenous fluids located at the pores of a sample, using numerical models to estimate permeability based on the spin relaxation time [28]. It can estimate in situ formation permeability much faster than other methods, making it useful for measurements in offshore reservoirs [29].

GRI is mainly used in tight rocks and consists in passing a high-pressure gas through a reference cell, which is more convenient for the removal of liquids inside the sample and avoids the problem of induced fractures in the samples as the core-flooding experiment [30]. However, it is susceptible to significant variations in permeability due to micro-fractures in the geometry, which may produce inaccurate results.

These drawbacks show that no perfect experimental analysis can predict permeability for any rock sample. The intrinsic laboratory difficulties and the time needed for this analysis make them cost-intensive, which creates the urge for different methodologies for predicting permeability [31].

1.4 Numerical simulation

Different computational approaches to the problem of measuring permeability have been used throughout the literature. Major strategies revolve around approximating the macroscopic medium by Representative Element Volume (REV), using a pore network to approximate the media, or performing direct simulation over the porous geometries obtained through tomographies and image scanning.

In these three scenarios, traditional CFD methods have been deployed to permeability calculation, such as the finite difference method [23], the finite volume method [32], and the finite element method [33]. Also, especially for pore-scale simulation, the lattice Boltzmann method has gained much attention over the years [34].

Besides CFD, there are numerical strategies for estimating permeability directly from the parameters obtained from the characterization of the rocks [35]. The potential of Particle Swarm Optimization, Support Vector Regression, and restricted Boltzmann Machine has been shown in the prediction of permeability correlating data from oil fields [36]. It is also possible to estimate permeability with the aid of neural networks directly from images of porous media [37]. This will be the main scope of this work, which will be analyzed in detail in the further chapters.

The main advantages of numerical determination of permeability versus experimental determination are the non-destructive nature of numerical experiments, allowing multiple tests in multiple conditions to be performed over a single sample, the reduced costs, as laboratory analysis may be time and resource consuming, the accessibility to extreme conditions of pressure, usually unavailable in laboratories due to safety issues, the possibility of precisely controlling boundary conditions and the inherent flexibility of testing different samples directly and fast [38]. These reasons corroborate using only numerical simulation for permeability determination in this study. Different approaches for this task will be explored in the following topics.

1.4.1 REV approach and classical CFD

REV is a critical property of macroscopic porous media as it signifies the critical volume where the properties of the fractured rock mass will remain stable with the increase in volume [39]. These properties include porosity, permeability, and pore-connectivity, meaning that a well-defined REV can fully characterize a whole porous media through a sample.

The permeability of an entire given location, such as an aquifer or reservoir, can be measured through the definition of a REV if the porous domain can be considered periodic [40]. Macroscopic CFD methods may be used to perform the simulation over a mesh representing the REV of a medium [41] [42]. These traditional approaches have been extensively studied, and plenty of commercial solvers are available that make their implementation direct for the final user. However, the complexity of porous geometries offers challenges to these solvers.

The main problems related to deploying classical CFD in porous media regard the trade-off between oversimplificating the porous domains, which may not capture the complex flow patterns that occur in natural domains, or to overdetailing the media, which may lead to a prohibitive computational requirement [43].

1.4.2 Pore network approach

Another strategy is simplifying the porous domain through a network formed by pores and throats. Pore network models have been extensively used in problems involving flow in porous media at the pore scale[44]. These models are advantageous as they allow multiphase flow simulation when integrated with CFD, and they allow the simplification of actual rock samples with spherical pores and cylindrical throats, approximately capturing the topology and geometry of the sample [45].

One broadly used choice for creating pore network models is the software OpenPNM [46]. In this software, it is possible to create the geometries from scratch and simplify authentic rock images as pore-network models. Additionally, it is possible to perform simulations on the networks created using the macroscopic conservation equations directly inside the software, without further codes or software for the transport equations implementation.

However, as with the REV approach, pore network models rely on oversimplifying geometries and cannot correctly simulate the microscopic details of natural porous media, relying on simplifying macro structures. These simplifications lead to inaccuracies due to the lack of resolution based on the pore-throat geometry, unable to represent the complexities of natural rock structures in microscale [47]. Therefore, these models may be deployed to macroscale reservoirs, but fail to capture the microstructure diversities that cause fluctuation in permeability values. Then, other strategies must be applied when the necessity of detail arises.

1.4.3 Direct numerical simulation

In recent years, the power of X-ray computed tomography [48] and Scanning Electron Microscopy (SEM) [49] have made it possible to obtain almost exact 3D images of porous media, limited by scanning resolution. These data allow direct fluid flow simulations over these geometries to determine permeability as a relation between the calculated velocity and pressure gradient. The lattice Boltzmann method is generally a core choice to perform these simulations due to its mesoscopic characteristics and natural ability to deal with fluid flow at the pore scale.

The main advantages of direct numerical simulation over the pore network approach include the high fidelity and accurate flow resolution over the microscopic rock structure, allowing valuable insights into the pore-scale physics such as pore filling processes [50]. The direct simulation allows precise results without simplification of the structures, even though it may be more computationally expensive, leaving the choice between pore-networks or direct numerical simulation as a trade-off between the availability of computational resources and the accuracy needed of the simulations carried [51].

The choice of using direct numerical simulation in this study justifies joining its strengths of high-fidelity simulations while bypassing its disadvantages of computational intensity. This is mainly done through the aid of artificial intelligence, trying to correlate the rock images and their characterizing parameters in a network that should be able to predict permeability several orders of magnitudes faster. Direct simulation data is required to train this network, but once the training is complete, high-fidelity results can be achieved without costly simulations.

1.5 Objectives

1.5.1 General objective

Considering the challenges exposed in the previous topics, it is clear that permeability prediction is a complicated task and that many approaches can be used to deal with this problem, with the advantages and disadvantages of each method. Then, the general objective of this work is to predict permeability using a deep neural network trained with direct numerical simulation data obtained via the lattice Boltzmann method. This is expected to minimize the problems with the lack of detail while being computationally efficient once the training is complete.

1.5.2 Specific objectives

To achieve the general objective of this work, some specific objectives are proposed:

- First, to better understand, gain experience, and construct theoretical knowledge with the mathematical tools of the kinetic theory of gases and the lattice Boltzmann method.

- Second, to code a lattice Boltzmann algorithm capable of reproducing a system's correct hydro-dynamical behavior and validating it according to literature.

- Third, to generate and characterize hundreds of artificial rock geometry samples for further use in direct numerical simulation.

- Fourth, to guarantee that permeability may be calculated through Darcy's law by securing that the simulations are carried in the Stokes regime, and then calculating the permeability for all rock geometries.

- Then, finally fifth, to calculate permeability directly from the generated geometry's images with the aid of artificial intelligence.

1.6 Text structure

This work is divided into five main chapters.

In chapter two, the fundamentals of Boltzmann's kinetic theory are presented as they form the basis of LBM theory. It introduces the concept of the particle distribution function, the fundamental variable in LBM. Then its evolution in space and time is analyzed through the Boltzmann Transport Equation, resulting in the macroscopic conservation laws due to its moments. Finally, the discretization procedure is discussed to obtain the lattice Boltzmann equation.

In chapter three, a brief explanation of the lattice Boltzmann method is presented to familiarize the reader with the main algorithm used in the simulations presented in this work. It introduces the algorithm cycle: initialization, applying boundary conditions, equilibrium calculation, collision, streaming, and convergence check. Then, the proposed code is validated through a laminar flow between parallel plates, as an analytical solution for this problem is available through Navier-Stokes assumptions.

In chapter four, permeability calculation with LBM is discussed. A bibliography review shows the chronology of the attempts to predict permeability with LBM. Then, the methodology for generating the geometries used in this study is presented, explaining the software used and the main assumptions made in the process. The characterization of such geometries is also explained, focusing on its main parameters and how they are computed. Finally, the results of a flow through porous media guarantee that the flow is in Stokes regime, showing the validity of Darcy's law. This is necessary to validate permeability as a linear relation between the pressure gradient and the mean velocity along the flow axis.

In chapter five, the deep neural network architecture is presented. It starts with a brief review of Neural Network architectures, reviewing standard models present in the literature. Then, the mathematical foundations of the model used are presented, creating the path for the implementation methodology. Then, the final results are shown, and the sensitivity analysis is discussed, analyzing the importance of each parameter input in the model.

Finally, chapter six summarizes the discussion and the main results presented. The difficulties found in the study are also reviewed, to advise the readers for further implementation. Then, research topics and suggestions are presented for future works.

Chapter 2

The lattice Boltzmann method

2.1 Mathematical background

The Lattice Boltzmann Method (LBM) arose in the 1980s as a new suitable option for fluid flow simulation [52]. It is a reliable numerical tool for simulating transfer phenomena such as heat [53] [54] and mass transfer [55] [56] along multiphase fluid dynamics [57] [58] [59]. Due to its particle-based nature differs from traditional Computational Fluid Dynamics (CFD) methods that describe a fluid macroscopically, in LBM the microscopic behavior of particles is collected to calculate the macroscopic properties of fluids. It is then considered a mesoscopic method, as it is mathematically founded upon Boltzmann's statistical mechanics, and can naturally handle the adversities of simulating micro and nano flows.

For this reason, LBM has gained attention due to its capability of simulating flow in porous media [60] [61], being capable of coupling multiphase flow [62] [63] and transfer phenomena [64] [65] [66] in these domains. Its squared-grid nature makes it possible to accurately represent complex domains, as it treats any image as a collection of squares (or cubes, in 3D). The interaction between the fluid and the solid interface is also facilitated regarding traditional CFD, without significant concerns with the resulting velocity direction due to the discrete velocity space, which only allows specific movement directions within the grid. Finally, it is naturally suitable for parallelization as it only needs local information for calculating the following timestep, allowing faster simulations in larger domains.

This chapter aims at introducing the basic concepts of the method and show classical hydrodynamic results that validate its ability to simulate fluid flow, especially in porous domains. It introduces Boltzmann's Kinetic Theory of Gases, its fundamental variables, and the Boltzmann Transport Equation (BTE). Then, BTE's discretization procedure is discussed in detail, enabling its computational implementation in a regular lattice as the Lattice Boltzmann Equation (LBE). On the following topic, the Lattice-Boltzmann algorithm is presented detailing each timestep cycle. Finally, the Poiseuille flow is solved as a reference problem, to guarantee the hydrodynamical behavior of the flow and show its capabilities of recovering NSE results.

In this chapter, the notation used by Kruger et al [67] was followed, as his book has become the standard reference to the Lattice-Boltzmann Method. Also, the works of Guo and Shu [68] and Succi [69] were vastly used throughout the text.

2.2 Kinetic Theory of Gases

The Kinetic theory of gases consists of understanding fluids as a collection of particles that fly randomly through space and change their velocity by colliding with each other. The kinetic theory was developed mainly in the XIX century as a scientific discussion between two prominent scientists of that time, James Clark Maxwell and Ludwig Boltzmann [70]. It led to prominent achievements in theoretical physics as the statistical understanding of entropy, rarefied gases, and surface phenomena, explicit expressions for viscosity, heat conduction, and diffusion, and ultimately led to the foundation of quantum mechanics [71].

One of the main equations in kinetic theory is the Boltzmann Transport Equation, a transport equation from which it is possible to recover the macroscopic conservation equations with a microscopic approach, and used to solve fluid flow problems at the mesoscale. As this equation holds the theoretical basis of the Lattice Boltzmann Method, the Kinetic Theory of Gases is briefly presented in this section.

2.2.1 The particle distribution function

The idea of the distribution function is a powerful concept, and it is the fundamental variable of the lattice Boltzmann method as it enables complex elaboration on the density of particles. $f(\mathbf{x}, \boldsymbol{\xi}, t)$ regards the evolution in time of both the distribution of mass in space ($\mathbf{x} = x, y, z$) and the distribution of mass with specific velocities, creating a so called velocity space ($\boldsymbol{\xi} = \xi_x, \xi_y, \xi_z$).

It means that in each coordinate, the macroscopic density of particles will be the sum of f over all its possible velocities $\boldsymbol{\xi}$, or that the collection of particles with all possible velocities in a given control volume will be equivalent to the density of particles in that control volume. Mathematically, this can be written as:

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \boldsymbol{\xi}, t)\ d\boldsymbol{\xi} \tag{2.1}$$

This is known as the zeroth moment of f, defined as an integral weighted by the variable (In this case, $\boldsymbol{\xi}$). This result may be interpreted as visualizing density in a given point $\mathbf{x}$ and time t as the sum of the mass of a distribution of particles with different speeds at that point: there will be particles in high speeds, particles in low speeds, and most particles have an average speed. Density is found when all of these particles are collected. Besides, f allows mathematically obtaining more physical information than simply from density. For example, it is possible to discover the density of the particles flying in one specific direction: it would be the integral on velocity space $\boldsymbol{\xi}$ over that direction. It is also possible to find the density of particles moving in a specific range of speeds.

Analogously to density, it is possible to consider two other moments with physical meanings. The first moment of f is related to the momentum density of each particle $f\boldsymbol{\xi}$, and its second moment to the individual kinetic energy $\frac{1}{2}f\boldsymbol{\xi}^2$. They are written as:

$$\mathbf{u}(\mathbf{x},t)\ \rho(\mathbf{x},t) = \int f(\mathbf{x},\boldsymbol{\xi},t)\ \boldsymbol{\xi}\ d\boldsymbol{\xi} \tag{2.2}$$

$$E(\mathbf{x},t)\ \rho(\mathbf{x},t) = \frac{1}{2}\int f(\mathbf{x},\boldsymbol{\xi},t)\ |\xi^2| d\boldsymbol{\xi} \tag{2.3}$$

This set of equations represents one of the significant advances of Kinetic theory: the link between the microscopic variables and their macroscopic counterparts. Through the concept of f, the individual nature of particles is abandoned for a collective description of ensembles of particles, and the dynamic behavior of the system may be predicted using statistical techniques.

2.3 The Boltzmann Transport Equation

It is now possible to write a balance for f, accounting for its change over time under three main assumptions: First, the particles are considered pointlike, hard spheres with a much smaller radius than the distance between them, so their volumes can be neglected. It is also assumed that particles only have one degree of freedom, i.e. translational energy, so changes in f happen only due to elastic binary collisions. Finally, the potentials of each particle are considered to be short-ranged, not affecting each other due to long-term interactions. Then, BTE may be written as:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi_1} \cdot \frac{\partial f}{\partial \mathbf{x_1}} = \frac{1}{m}\int \mathbf{V}\left[f(\mathbf{x_1},\boldsymbol{\xi_1}',t)\ f(\mathbf{x_1},\boldsymbol{\xi_2}',t) - f(\mathbf{x_1},\boldsymbol{\xi_1},t)\ f(\mathbf{x_1},\boldsymbol{\xi_2},t)\right] d\omega d\boldsymbol{\xi_2} \tag{2.4}$$

The right side of the equation is known as the collision integral and represents the momentum exchange between particles 1 and 2, with respective microscopic velocities $\boldsymbol{\xi_1}$ and $\boldsymbol{\xi_2}$ before and $\boldsymbol{\xi_1'}$ and $\boldsymbol{\xi_2'}$ after collision. A new term $\mathbf{V} = \boldsymbol{\xi_2} - \boldsymbol{\xi_1}$ is introduced representing the relative microscopic velocity. The integration happens over $d\boldsymbol{\omega}$ and $d\boldsymbol{\xi_2'}$, representing the geometric space in polar coordinates and the velocity space of the second distribution, respectively.

For simplicity, it is common to omit the collision integral in its full form as a collision operator $\Omega(t)$. It is also possible to add a force term $\mathbf{F}$ that represents the influence of potentials upon the distribution. With these two terms, the most common way of writing BTE is:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{\mathbf{F}}{\rho} \cdot \frac{\partial f}{\partial \boldsymbol{\xi}} = \Omega(t) \tag{2.5}$$

This can be seen as an advection equation for the property f. Even though its solution might be challenging, the consequences of Boltzmann's Transport Equation offer useful information on the analysis of fluid systems.

It is expected that in an elastic collision, mass, momentum, and energy are conserved. The Boltzmann Equation proposes a balance for f to respect these constraints. Then any function may be used as a collision operator substituting the collision integral if they respect the constraints:

$$\int \Omega(t) \begin{bmatrix} 1 \\ \boldsymbol{\xi} \\ |\xi^2| \end{bmatrix} d\boldsymbol{\xi} = 0 \tag{2.6}$$

This set of equations represents the mass, momentum, and total energy conservation equations regarding the collision $\Omega(t)$ that transports quantity f, and the terms inside the brackets are called collisional invariants.

It was not until 1954 when Bhatnagar, Gross, and Krook proposed an alternate solution [72] considering collision simply as a deviation from the Maxwellian equilibrium distribution function.

$$\Omega(t) = -\frac{1}{\tau}(f - f^{eq}) \tag{2.7}$$

Where τ is the relaxation time, a constant with magnitude with order of the mean free path time of molecules between collisions, and f^{eq} is the distribution function in equilibrium. The relaxation time τ is related to viscosity due to the non-slip boundary condition, as the mean time between collisions will be the mechanism that relates the collisions with viscosity. This is expressed through:

$$\nu = c_s^2 \left(\tau - \frac{1}{2} \right) \tag{2.8}$$

2.3.1 The equilibrium distribution

Maxwell's description of the equilibrium distribution was presented almost one hundred years before BGK presented their solution to BTE [73], following a work on swarms of particles that he proposed as a model for the motion of Saturn rings [70].

$$f^{eq}(\mathbf{x}, \boldsymbol{\xi}, t) = \frac{\rho}{(2\pi RT)^{\frac{3}{2}}}\ exp\left[-\frac{(\boldsymbol{\xi} - \mathbf{u})^2}{2RT}\right] \tag{2.9}$$

Essentially, this equation represents the normal distribution of f with three degrees of freedom related to the three components of the velocity vector. This description has found prestige by accurately describing viscosity, allowing Maxwell to calculate the viscosity of air correctly and show that viscosity was independent of density.

Curves with these characteristics are known as Maxwellians, even though the distribution itself is known as the Maxwell-Boltzmann distribution, accounting for Boltzmann's improvements in Maxwell's works [74].

2.4 The Lattice-Boltzmann Equation

The Lattice Boltzmann Method initially evolved from the Lattice Gas Automata (LGA). Hardy, Pomeau, and Pazzis first conceived LGA as a particle-based scheme for fluid flow called Hardy Pomeau and Pazzis (HPP) Model [75]. The idea is to simulate the fluid as particles confined in the corners of a 2-dimensional grid. These particles have associated velocities pointing to the four possible directions of the grid (Up, down, left, right). At each iteration, particles move one grid space toward their velocity. Each node can only contain one particle. If two particles are to occupy the same node coming from opposite directions, they collide and are deflected perpendicularly. If not, they pass through each other. As simple as this model may seem, it conserves linear momentum and therefore runs indefinitely without ever losing energy, it also lacked rotational invariance due to its squared nature.

The anisotropy problem was solved in 1986 when Frisch, Hasslacher, and Pomeau proposed a hexagonal lattice with six possible velocities as the Frisch Hasslacher and Pomeau (FHP) model [76]. Another fundamental difference between HPP and FHP models is that collisions are not deterministic in the latter, and may produce two possible outcomes due to a single collision. This outcome requires a pseudorandom process to decide the directions, adding a layer of empiricism to the method. Finally, it is still a boolean process where the particles have a 1 or 0 probability of occupying a cell in the lattice, so it suffers from statistical noise. Also, this model is not easily scalable for three dimensions due to symmetry problems caused by the hexagonal grid.

With the advances in hardware, LBM naturally evolves from LGA as a floating point algorithm instead of a boolean algorithm, where the particles follow a density distribution function when occupying a lattice cell. McNamara and Zanetti could use the Boltzmann Equation to eliminate statistical noise, with a resulting algorithm that demanded less computational time and was more efficient for low Reynolds number [52].

However, although LBM shares many similarities with LGA and historically has been developed as an extension of the model, LBE can be directly derived from BTE. Thus, it can be seen as an independent method. It was first proved by He and Luo [77] by deriving the Lattice Boltzmann Equation directly from the Boltzmann Transport Equation using a Mach series expansion.

In the mid-2000s, another strategy was presented by Shan and Yuan [78], which consisted of applying a Hermite series expansion on BTE. This procedure is known as being more mathematical rigorous as it takes advantage of the similarities between the Hermite weight function and the Equilibrium distribution function.

Guo and Shu [68] remark that LBE is originated from LGA, but was shown later to be an independent method found through the discretization of BTE with the right mathematical tools. This characterizes LBM as a quite different method than other CFD algorithms that are essentially solvers of the Navier-Stokes Equations.

In the following section, the text presents a simplified version of the Hermite Series approach, as it is as mathematically rigorous as it is heavy. The complete procedure may be found in the original article [78] or explained fluidly in the third chapter of Kruger's book [67].

2.4.1 Discretization in velocity space: Hermite polynomials

The discretization procedure begins in velocity space and compares the Equilibrium distribution function with the Hermite Polynomials generator function. The Hermite Polynomials of order n are created following the rule generalized for d dimensions as:

$$\mathbf{H}^{(n)}(\mathbf{x}) = (-1)^n \; \frac{1}{w(\mathbf{x})} \; \nabla^{(n)} w(\mathbf{x}) \tag{2.10}$$

Where $w(\mathbf{x})$ is the generator function, that reads as:

$$w(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}} \; e^{-\frac{\mathbf{x}^2}{2}} \tag{2.11}$$

As the Hermite Polynomials are a class of Orthogonal Polynomials, they form a basis for $\mathbb{R}$ and can represent any continuous real function as a series of Hermite

Polynomials. Thus, any generic continuous function $g(\mathbf{x})$ can be written as:

$$g(\mathbf{x}) = w(\mathbf{x}) \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{a}^{(n)} \mathbf{H}^{(n)}(\mathbf{x}) \tag{2.12}$$

Where $\mathbf{a}^{(n)}$ are the generalized series coefficients:

$$\mathbf{a}^{(n)} = \int g(\mathbf{x}) \mathbf{H}^{(n)}(\mathbf{x})\ d^d\mathbf{x} \tag{2.13}$$

Taking a closer look at equation 2.11, it is possible to see that it resembles equation 2.9. By writing the adimensional temperature $\theta = \frac{RT}{V^2}$, it is easy to see that the form of the weight function naturally resembles the equilibrium distribution function, so it is possible to write the equilibrium distribution through the idea of the weight function as:

$$f^{eq} = \frac{\rho}{\theta^{\frac{d}{2}}}\ w\left(\frac{\boldsymbol{\xi} - \mathbf{u}}{\sqrt{\theta}}\right) \tag{2.14}$$

Through this definition, it is possible to write the series coefficients as:

$$\mathbf{a}^{(n)\ eq} = \frac{\rho}{\theta^{\frac{d}{2}}} \int w\left(\frac{\boldsymbol{\xi} - \mathbf{u}}{\sqrt{\theta}}\right) \mathbf{H}^{(n)}(\mathbf{u})\ d^d\mathbf{u} \tag{2.15}$$

This equation may be solved with the aid of mathematical software to give the results for the first three coefficients of the series:

$$\mathbf{a}^{(0)\ eq} = \rho \tag{2.16a}$$

$$\mathbf{a}^{(1)\ eq} = \rho u_\alpha \tag{2.16b}$$

$$\mathbf{a}^{(2)\ eq} = \rho(u_\alpha u_\beta + (\theta - 1)\ \delta_{\alpha\beta}) \tag{2.16c}$$

This result shows that the conserved quantities ρ, $m\mathbf{u}$ and E are directly related to the coefficients of the Hermite series expansion, so it is also possible to write the distribution function with the aid of the Hermite polynomials:

$$\mathbf{a}^{(0)\ eq} = \int f^{eq}\ d\boldsymbol{\xi} = \rho = \int f d\boldsymbol{\xi} = \mathbf{a}^{(0)} \tag{2.17a}$$

$$\mathbf{a}^{(1)\ eq} = \int f^{eq}\ \boldsymbol{\xi} d\boldsymbol{\xi} = \rho\mathbf{u} = \int f\ \boldsymbol{\xi} d\boldsymbol{\xi} = \mathbf{a}^{(1)} \tag{2.17b}$$

$$\frac{\mathbf{a}^{(2)\ eq} + \rho d}{2} = \int f^{eq}\ \frac{|\boldsymbol{\xi}^2|}{2} d\boldsymbol{\xi} = \rho E = \int f\ \frac{|\boldsymbol{\xi}^2|}{2} d\boldsymbol{\xi} = \frac{\mathbf{a}^{(2)} + \rho d}{2} \tag{2.17c}$$

These equations establish the link between the Hermite polynomials and the distribution function. The advantage of this approach is that it makes it possible to express the integrals over f (its momenta) as summations in a finite set of points, called abscissae. If one is to choose these points as the roots of $\mathbf{H}(\mathbf{x})$, any polynomial $P^{(N)}(\mathbf{x})$ with $N = 2n - 1$ will be exactly:

$$\int_{-\infty}^{\infty} w(x)\ f(x)\ dx = \sum_{i=1}^{q} w_i\ f(x_i) \tag{2.18}$$

This is called the Gauss-Hermite quadrature, Where q represents the number of abcissae x_i and w_i equals to:

$$w_i = \frac{n!}{(nH^{(n-1)}(x_i))^2} \tag{2.19}$$

Equation 2.18 resembles the moment equations for f, thus opening a path to represent these moments in a discrete grid with the right sets of weights and abscissae. The procedure is to apply the Gauss-Hermite quadrature in equations 2.17a, and after some mathematical manipulation, the resulting discrete equilibrium function for the isothermal case can be read as:

$$f_i^{eq} = w_i\rho\left(1 + \frac{c_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{u} \cdot c_i)^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{c_s^2}\right) \tag{2.20}$$

This equation introduces new concepts as the discrete velocity sets c_i, the weight of each velocity direction w_i, and the sound speed c_s. The discrete velocities need further explanations as they define how the distributions evolve in space and time. Here, a finite set of movement directions is chosen to guarantee mass, momentum, and energy conservation, as a simplified equilibrium f^{eq} and a discrete velocity set are sufficient to obtain the correct macroscopic conservation laws [67]. Additionally, rotational isotropy must be respected. The weights of each direction are then calculated to respect the conservation and isotropic constraints.

Then, using equations 2.17, it is possible to recover the macroscopic momenta as a summation over the distribution function, as the macroscopic conservation laws are respected in the finite set of velocities c_i.

$$\rho = \sum_i f_i^{eq} = \sum_i f_i \tag{2.21a}$$

$$\rho\mathbf{u} = \sum_i c_i\ f_i^{eq} = \sum_i c_i\ f_i \tag{2.21b}$$

These equations represent the end of the discretization in velocity space. However, it is still necessary to discretize f in $\mathbf{x}$ and t.

2.4.2 Discretization in time: Explicit Euler scheme

The LBM algorithm is classically discretized using an explicit forward Euler scheme for time and a squared regular grid for space. This guarantees that the distribution passes its values to the neighboring nodes following the velocity set directions in each time step.

Following these rules, the main equation of the algorithm, the Lattice-Boltzmann Equation, may be finally obtained:

$$f_i(\mathbf{x} + c_i\Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \Omega_i(\mathbf{x}, t) \tag{2.22}$$

Where simply the value of the distribution at a given time step $t+\Delta t$ and position $\mathbf{x} + c_i\Delta t$ will be the value of f at the previous time step t and position $\mathbf{x}$ plus a collision term $\Omega_i(\mathbf{x}, t)$. The collision is generally modeled using the BGK collision operator as a deviation from equation 2.20, originating the LBGK equation.

It is important to remark that all the simulations in this text are two dimensional. The most common velocity set used for 2D hydrodynamics is the D2Q9 velocity set (Figure 2.1), where D stands for 2 dimensions and Q for the 9 possible movement directions for a given point in a squared lattice (2 horizontal, 2 vertical, 4 diagonal,s and 1 rest position). The main parameters of the D2Q9 are summarized below in table 2.4.2.

D2Q9 velocity set			
Velocities (c_i)	Number	Length ($\lvert c_i \rvert$)	Weight (w_i)
(0,0)	1	0	4/9
(±1, 0) (0,±1)	4	1	1/9
(±1,±1)	4	$\sqrt{2}$	1/36

Table 2.1: D2Q9 velocity set parameters

2.5 The algorithm

The lattice Boltzmann method is a numerical algorithm that can simulate fluid flow in a regular squared grid. This is possible through a closed loop of equations that are solved at each timestep in each grid node. By imposing a boundary condition, a deviation from the equilibrium is induced and the system loops until it reaches a steady state. Possessing equations 2.20, 2.21 and 2.22, it is possible to write this closed loop that can be used to simulate fluid systems.

Figure 2.2 shows the main steps of the algorithm. It starts by uploading a geometry containing fluid and solid nodes and initializing the density and velocity fields. The first step of the loop is to calculate the macroscopic properties using

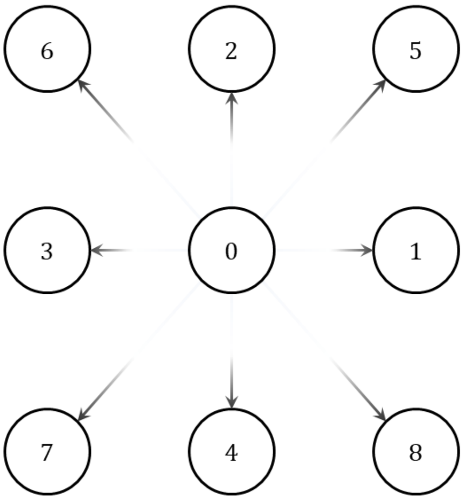

Figure 2.1: The D2Q9 velocity set is presented with the common notation for each possible velocity direction.

the moment equations. The boundary conditions are then applied to move the system out of rest, and the equilibrium condition at each node is calculated to create a difference between the actual values of f and the values of f^{eq}. The LBE is applied and distributes the post-collision values of f through the grid, in two separate processes of Collision and Streaming. If the post-collision distributions meet the convergence criteria, the simulation finishes and outputs the results.

This section describes in detail the main algorithm used for the simulations present in this work.

2.5.1 Initialization

The algorithm starts by choosing a geometry on which to perform simulations. The geometry is drawn in a simple boolean-squared lattice where 0 represents the solid nodes, and 1 represents the fluid nodes. The density, velocity, and distribution function matrices are created with dimensions equal to the image input. The equations presented throughout this chapter are only defined in fluid nodes, leaving the solid nodes as obstacles that the fluid dodges. In the remaining nodes, we define the initial conditions for density, velocity, and the distribution functions, using $\rho = 1$, $\mathbf{u} = \mathbf{0}$, and the equilibrium assumption to calculate the distribution functions values.

In initialization, the flow parameters are also set. The Reynolds number will dictate the flow regime, that acts as a bridge from the lattice domain to the physical domain. In this algorithm, Re and u_{max} are inputs, leaving the adimensional kinematic viscosity ν as a free parameter, as the density and the characteristical length

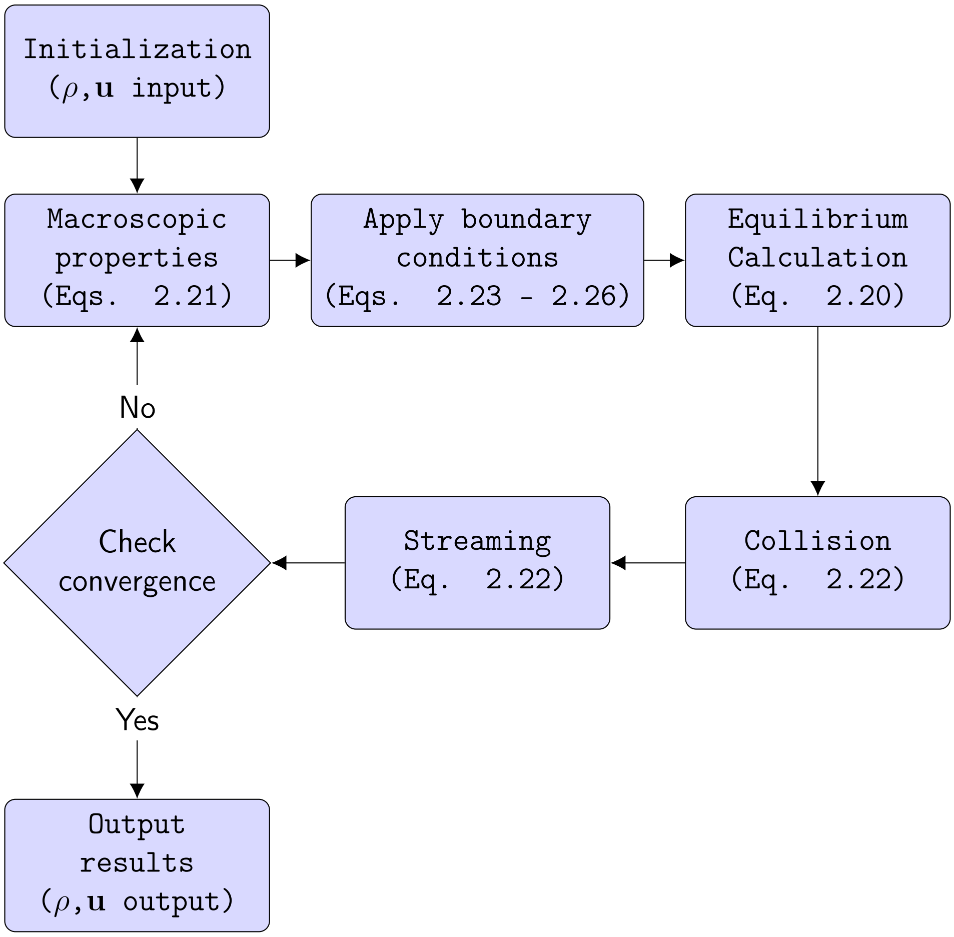
Initialization
(ρ,$\mathbf{u}$ input)
Macroscopic properties (Eqs. 2.21)
Apply boundary conditions (Eqs. 2.23 - 2.26)
Equilibrium Calculation (Eq. 2.20)
No
Check convergence
Streaming (Eq. 2.22)
Collision (Eq. 2.22)
Yes
Output results (ρ,$\mathbf{u}$ output)

Figure 2.2: Flowchart of the main algorithm used in this work's simulations.

are also fixed.

As viscosity remains as a free parameter, so remains the relaxation time. This choice of parameters has the objective of improving stability, as the process of manually choosing τ values is inefficient. Not optimal values may cause slow simulation times, and wrong values may even crash the simulation.

With fluid and flow parameters set, the simulation loop starts. The first step of the loop is to calculate the macroscopic properties, applying the momentum equation to calculate ρ and $\mathbf{u}$ values based on the equilibrium distributions.

2.5.2 Boundary Conditions

At this point, the system remains in equilibrium. The boundary conditions are set to move the system out of rest, creating the gradient that will be the driving force of the flow. Here the entrance velocity is set, which may be seen as the most critical parameter of the simulation, as the algorithm goal is to calculate the pressure gradient generated by the imposition of a specified velocity in a given geometry.

Two boundary conditions are applied: First, the bounce-back boundary condition, which is equivalent to the non-slip condition [79]. Second, the Zou-He boundary conditions are Dirichlet boundary conditions where a fixed value of ρ and $\mathbf{u}$ are set [80]. These boundary conditions are known as Open Boundaries, where the fluid is created on the inlet boundary and is destroyed at the outlet boundary, they differ from Periodic Boundaries, where the fluid that escapes on the outlet returns to the inlet at each iteration, maintaining its density and velocities values.

As the Bounce-back scheme is applied intricately with streaming, it will be discussed in the following topics of this section. Now, the Zou-He boundary conditions will be addressed.

2.5.2.1 Open Boundaries

Differing from Periodic Boundaries, where symmetry considerations must be made to guarantee that the fluid behavior corresponds to the actual physics of the problem, the Open Boundaries may be applied in any geometry without further concerns about symmetry. The Open Boundaries are, therefore, more versatile than the Periodic Boundaries, but are trickier to implement due to the mathematical development behind it.

One of the suitable choices for the implementation of open boundaries is the Zou-He boundary conditions. The idea is to use the bounce-back of the non-equilibrium distributions at the boundaries and then formulate a fixed equation to the distributions that will continuously create fluid at the entrance of the domain and destroy fluid at the exit. The equations applied in the inlet and outlet are written so that they always conserve mass, momentum, and energy.

The Zou-He Boundary Conditions specify the macroscopic variable value in a node and then calculate all the following quantities through that specified value. The complete formalism may be found in the original article [80]. Here, a brief version of the main equations will be presented.

Generally, velocity is specified on the inlet, and density is specified on the outlet. Considering specifying macroscopic velocities u_x and u_y at the inlet, and that the inlet is positioned at the left border of the domain following the standard D2Q9 orientation, the macroscopic equation for density is written as:

$$\rho = \frac{1}{1-u_y}\left(f_0 + f_2 + f_4 + 2\cdot(f_3 + f_6 + f_7)\right) \tag{2.23}$$

Now in the outlet, velocity is calculated as:

$$u_x = -1 + \frac{(f_0 + f_2 + f_4 + 2\cdot(f_1 + f_5 + f_8))}{\rho} \tag{2.24}$$

The macroscopic variables have been set, but some distributions are still missing in this numerical scheme, specifically those that enter the system at the inlet (1, 5, and 8) and those that leave the system at the outlet (3, 6, and 7). For the inlet, the distributions may be written as:

$$f_1 = f_3 + \frac{2}{3}\rho u_x \tag{2.25a}$$
$$f_5 = f_7 + \frac{1}{2}(f_4 - f_2) + \frac{1}{6}\rho u_x + \frac{1}{2}\rho u_y \tag{2.25b}$$
$$f_8 = f_6 + \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho u_x - \frac{1}{2}\rho u_y \tag{2.25c}$$

Then, for the outlet, a similar set of equations is applied:

$$f_3 = f_1 - \frac{2}{3}\rho u_x \tag{2.26a}$$
$$f_7 = f_5 + \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho u_x - \frac{1}{2}\rho u_y \tag{2.26b}$$
$$f_6 = f_8 + \frac{1}{2}(f_4 - f_2) - \frac{1}{6}\rho u_x + \frac{1}{2}\rho u_y \tag{2.26c}$$

These equations may vary for each geometry and must be recalculated for each problem. In all simulations presented in this work, fluid inlets are defined at the left border, and fluid outlets are defined at the right border of the domain. But this is not always the case, as if a geometry has inlets or outlets in the top or bottom borders, a new set of equations needs to be formulated to account for the entrance and exit of fluid through f_i with $i = (2, 4, 5, 6, 7, 8)$ according to D2Q9.

2.5.3 Equilibrium calculation

The new values of ρ and $\mathbf{u}$ must be associated with a new value of f^{eq} to accurately reproduce the particle equilibrium distribution at these new conditions in phase space. At the initialization the particles in each node were at rest, so the equilibrium initially set was equal in each node. As velocity and density have changed at the borders, f^{eq} values must be recalculated through equation 2.20.

2.5.4 Collision

Having f corrected by the Zou-He scheme and recalculated f^{eq}, it is time to apply the LBE. This is computationally done in two main steps: Collision and Streaming. First, in collision, the post-collision values f^* are obtained with the difference between f and f^{eq}. Then, in streaming, these values are distributed to their neighboring nodes.

In this work, collision is performed using the MRT Collision Operator, as it has been proved as a better choice for porous media simulations than BGK [81]. The main advantages of the MRT over the BGK collision operator are that the LBM-MRT scheme does not violate the Galilean invariance and that it is less susceptible to accuracy deviations due to τ values [82]. The idea is to perform collision in moment space rather than in population space, allowing more than one relaxation time, which is the main parameter that impacts the accuracy and stability of the simulations.

There are different strategies for obtaining the MRT matrix that mathematically performs the transformation of f and f^{eq} from the population space to the moment space. One of the first approaches to this problem is to use the steady recurrence equations obtained through linear combinations of the evolution of the LBE, which leads to a closure problem that may be solved with a proper matrix of "magic numbers" that conserves mass, momentum, and energy while allowing multiple relaxation times [83]. On the other hand, the algorithm presented in this work follows the description by Kruger [67] in the 10th chapter of his book, using the Gram-Schmidt procedure.

2.5.4.1 MRT Collision Operator

The LBE-MRT equation is read as:

$$f_i(\mathbf{x} + c_i\Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = -\mathbf{M}^{-1}\mathbf{S}\mathbf{M}[f - f^{eq}]\Delta t \tag{2.27}$$

Where **M** is the MRT Matrix, that performs algebraically transformation to the moment space, **S** is the relaxation times matrix and $\mathbf{M}^{-1}$ is the inverse of **M** that takes back the populations from the moment to the population space. Accordingly to the Gram-Schmidt procedure, **M** takes the form:

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{2.28}$$

This perhaps obscure set of numbers derives from a strong mathematical background, as they form a set of orthogonal vectors that can map the populations to a new set of orthogonal momenta. Subsequently, **S** carries the multiple relaxation times as:

$$\mathbf{S} = \begin{bmatrix} 0 & \omega_1 & \omega_2 & 0 & \omega_3 & 0 & \omega_3 & \omega_4 & \omega_4 \end{bmatrix}^T \tag{2.29}$$

Where $\omega_i = \frac{1}{\tau_i}$ represents the four different relaxation times with different physical meanings, with ω_1 related to bulk viscosity, ω_4 related to shear viscosity and ω_2 and ω_3 are free parameters to adjust according to simulation's stability.

2.5.5 Streaming

Finally, the last step of the algorithm is to redistribute the values of f^* to the correct places according to $\mathbf{x}^* = \mathbf{x} + c_i \Delta t$. These values will occupy the previous values of f in each node to close the cycle and start a new iteration. The streaming process is illustrated in Figure 2.3.

It is also in streaming that the bounce-back boundary conditions are naturally implemented, as they are set at the moment that a fluid node is to pass its f^* values to a neighboring solid node, the solid "deflects" the populations back to their specular counterparts 2.4. This effect is equivalent to the no-slip boundary condition to a second-order accuracy with a halfway wall scheme [84], as proposed in this work.

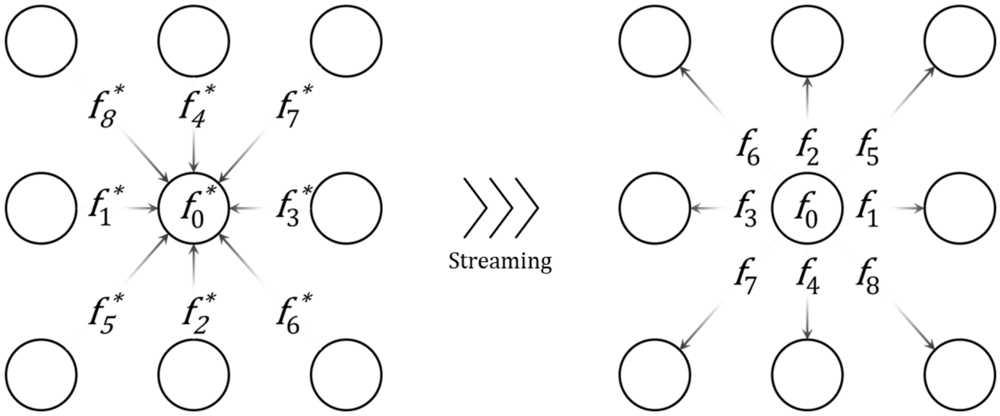

Figure 2.3: The nine components of f^* are shown before and after the streaming process. From a given node point of view, the left distribution shows how its neighbors pass their f^* values. After streaming, the right distribution shows f values in the following time step, closing the algorithm's loop.

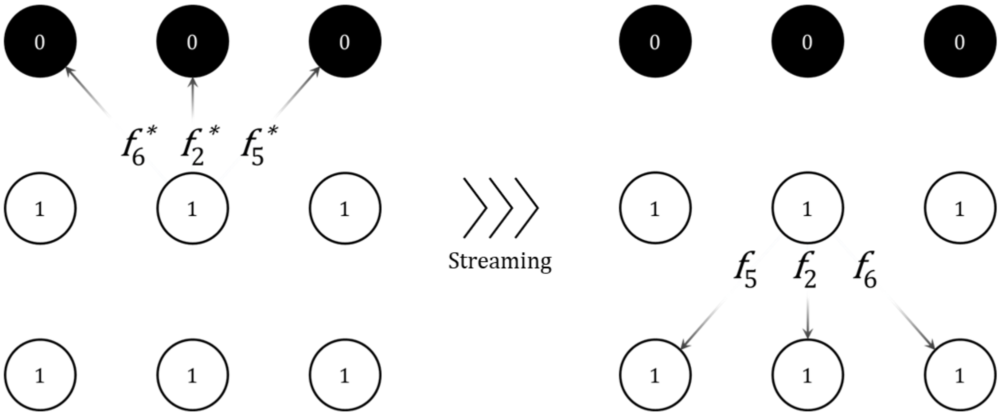

Figure 2.4: Three post-collision distributions collide with a solid wall, redistributing their f values in the respective specular reflection of each direction.

2.5.6 Check convergence

Now that the entire mathematical loop is detailed, explaining how the simulation stops is necessary. To end this loop, the flow must have reached a steady state, which is done through a convergence analysis. In this work, convergence is checked for density and velocity at each $\Delta t_{check} = 1000$ iteration, comparing its actual values with their values at a previous Δt_{check}. If the deviation is under a certain tolerance for both density and velocity, it is said that the system has reached its steady state concerning the imposed boundary conditions. Then, the simulation stops and outputs the results.

2.5.7 Application I: Poiseuille Flow

The Poiseuille Flow is a classic fluid flow problem. As it is an analytic solution of the Navier-Stokes Equation for parallel plates, it can be used as a benchmark to verify the validity of 2D fluid flow simulations. Also, it is appealing due to its simple geometry, making it a natural choice for computational implementation. With the no-slip boundary condition at the top and bottom walls, a laminar flow is induced with Dirichlet boundary conditions at the inflow and outflow, at a moderate Reynolds number.

The Poiseuille flow has consistently been used as a benchmark for the hydrodynamical consistency of an LBM scheme [85] [86] [87] [67]. In the lack of experimental data, the relatively simple analytic solution for the Poiseuille profile is generally the first goal to be achieved by a hydrodynamics simulation model. This problem is also helpful for τ values tuning, as the non-slip boundary condition is induced by bounce-back, different τ values may produce different results.

Starting with the Navier-Stokes Equation in its complete form, assuming that we have zero velocity at the walls, that the fluid has only velocity along the x-axis, that the plates are separated by a distance a, and that the flow is fully developed at the steady state, the governing equation for the velocity as a function of the position along the y axis is given by:

$$u_x = \frac{a^2}{\mu}\frac{\partial p}{\partial x}\left[\frac{y^2}{a^2} - \frac{y}{a}\right] \tag{2.30}$$

y represents the position along the y direction, μ represents the fluid's dynamic viscosity, and $\frac{\partial p}{\partial x}$ is its pressure gradient along the x-axis. Through this equation, it is possible to see the parabolic profile expected for the fluid to assume a steady state.

Regarding the Lattice-Boltzmann simulation, a domain composed of nothing but top and bottom walls is created. At the inflow, velocity is specified using

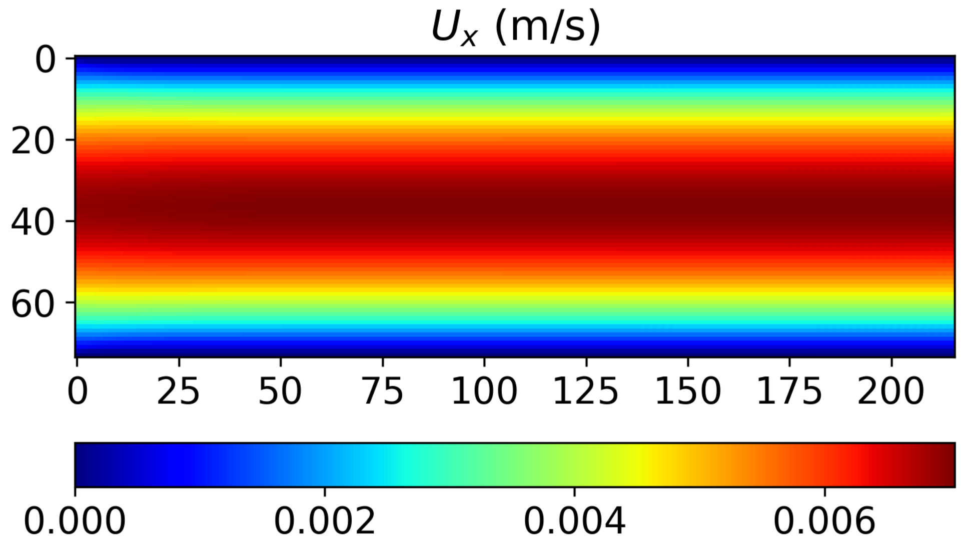

Figure 2.5: 2D Lattice Boltzmann simulation of a fluid flow between parallel plates. The plates are separated by a distance of $a = 0.001m$, and the x-axis has a length of $3a$.

Zou-He boundary conditions, equivalent to the Dirichlet boundary conditions. At the outflow, density is specified, as it is directly related to the pressure via LB's standard EoS. The flow direction is from left to right, and it is possible to see the laminar pattern created in Figure 2.5, where the rainbow profile denotes the parabolic profile with the zero velocity at the walls and the u_{max} at the center along the whole geometry.

It is possible then to compare these results with the analytic solution given by equation 2.30 to verify the validity of our model. First, determining which grid refinement will accurately match the Poiseuille solution is necessary. Eight grid sizes are chosen, starting with a Poiseuille channel of 9 + 2 pixels in width (counting two extra pixels for the channel walls) and 27 pixels in length, multiplying these numbers by factors counting from 1 to 8. Then we reach the configurations of 11x27, 20x54, 29x81, 38x108, 47x135, 56x162, 65x189 and 74x216 pixels. The results are shown in lattice units in Figure 2.6.

All grids could correctly match the analytical solution given by the Navier-Stokes equation. However, the coarser grids displayed more errors than the refined ones, almost matching the analytical solution. It is essential to notice the slight deviation near the walls present in all simulations, being less prominent as the grid becomes finer.

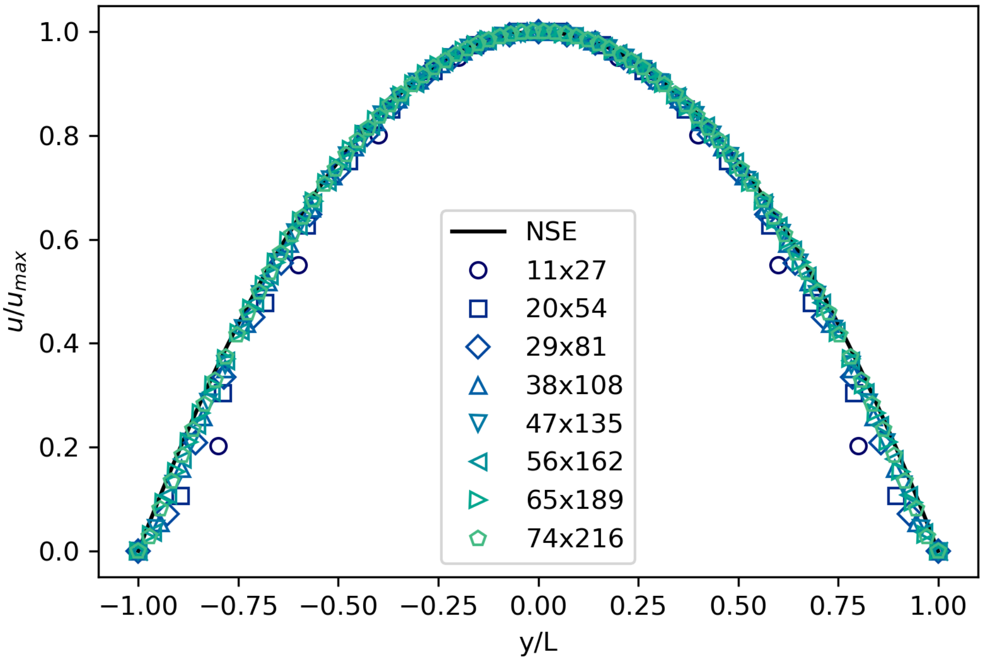

Figure 2.6: Grid convergence test resulting of a Poiseuille flow between parallel plates. Eight grid sizes solutions are compared with the analytical solution (In black) given by the Navier-Stokes Equation.

Therefore, it is possible to conclude that the proposed lattice Boltzmann scheme is an effective tool to solve hydrodynamic problems, with results equivalent to the traditional Navier-Stokes approach. Besides, it can solve it at the pore scale, which accounts for an advantage over NSE in the context of the problem it aims to solve.

Chapter 3

Permeability with LBM

This chapter presents a strategy to determine permeability through direct simulations over porous geometries with the lattice Boltzmann method, using the algorithm described in Chapter 3 to perform direct numerical simulation over artificial geometries. It starts with a bibliography review of experimental and numerical methods to determine permeability, showing why LBM is a suitable alternative to solve this problem in the pore scale. Then, the methodology for generating the porous geometries that will be used for direct lattice Boltzmann simulations is shown, focusing on the characterizing parameters such as tortuosity and porosity, among others. Finally Darcy's law results are presented to guarantee that the systems simulated respect the Darcy regime, so the permeability values calculated through Darcy's law are valid.

3.1 LBM in porous media

In recent years, the lattice Boltzmann method has been consolidated as the leading choice for flow in porous media applications [69]. This is due to its particle-like nature, efficiently dealing with irregular geometries regarding mechanical events, such as bounce-backs and mirror reflections. Also, its field-like nature carries all smooth hydrodynamic information, accessing tiny interstitial regions that macroscopic models could not achieve.

The first attempts to use LBM to estimate the permeability of porous geometries go back to the early 90s when Cancelliere et al [88] used a 64^3 voxels three-dimensional medium composed of randomly positioned spheres to emulate a porous media. At that time, the BGK operator was used to carry the simulations, and the movement of fluid particles was driven by an external force imposed upon the geometry. They concluded that lattice gas models effectively predict permeability and could also be extended for studies of hydrodynamic dispersion and multiphase flows.

In the first decade of the 2000s, this topic was further explored by directly simulating over 400^3 voxels tomographies of Fontainebleau sandstone comparing LBGK and Finite Differences [85]. It was shown that the two methods apply to the problem, with comparable computing time. However, LBM needed special attention with τ dependency on the non-slip boundary condition and the compressibility effects noticed. They compared the results with experimental values and concluded that direct numerical simulation was becoming a possible reality to determine permeability due to the increasing computational power available. Also in 2002, Kang et al [60] proposed a multiscale unified lattice Boltzmann model capable of recovering Darcy's law in an extensive range of scales. The method was very satisfactory in predicting the permeability of fractured systems where different scales coexist in a single medium.

In 2010, the MRT Collision Operator started being used to extend the LBGK model for porous media. Narvaez et al [87] concluded that, in general, BGK and MRT produced comparable results for quadratic, circular, and triangular pipes, however, when simulating tomographies of Fontainebleau sandstones, the increase in geometries complexity caused more significant errors for permeability in the BGK model, as it creates a dependency between τ and permeability. A few years later, Eshghinejadfard et al [89] tested different collision operators and force schemes to show that the permeability dependency with τ happens with all models, even though it may be primarily reduced using MRT coupled with the Guo forcing scheme. It also shows the dependency with proper grid size, with the coarser grids having more significant errors than the refined grids.

3.2 Methodology

This section aims to explain how the geometries used in this study were created, and characterized and how the simulations were performed.

3.2.1 Artificial geometries models

To perform simulations on porous media, choosing a proper model to generate the geometries upon which the simulations will take place is necessary. Porous media are naturally random, but some structures may be perceived and recreated with mathematical tools. It is then possible to make approximations of real media using computational algorithms for artificial porous media generation.

A model represents a simplification of the complex reality. Identifying significant structures makes it possible to simplify porous media as a gathering of these structures. Generally, porous media are recreated by two significant models: Capillarity networks and packing of particles [10]. These spatially periodic structures can be studied as deviations of more perfect and ordered structures such as FCC crystal. The randomness components added to the initial significant structures create good artificial representations of natural porous media.

Also, it is necessary to notice that porous media generally occurs in an extensive range of scales, and the significant structures common in a certain length may or may not be repeated throughout these scales. These kinds of objects repeating themselves in various lengths are called fractals, complex geometric structures often used to depict porous media. They were first described in the works of Sierpinski, Koch, Cantor, and Haussdorf, but Mandelbrot (1982) [90] created a systematic approach to unify all of those works.

All the models mentioned above may be used to create artificial porous media models. Ordered-placed spheres (or tubes, or squares), randomly placed spheres (or tubes, or spheres), fractal noises, the Sierpinski carpet and Gasket, the random Cantor dust, and the Voronoi Edges are some models that are commonly used for the generation of porous-like structures. None of these models is inherently better than the others, so choosing one model over another is purely a user's choice for the more accurate description of the system they want to study.

This study has used the Voronoi algorithm for porous media generation, as this pattern is typical, especially when dealing with rock fractures. This will be further explained in the following topic.

3.2.2 Voronoi algorithm

The Voronoi Algorithm (also called Voronoi Decomposition or Dirichlet tessellation [91]) represents a way of subdividing space around a given set of points, also called seeds, that are randomly distributed. The algorithm guarantees that each subdivided region contains all the points that are closer to a specific seed in relation to any other. This is exemplified in figure 3.1 with a randomly generated diagram around 30 seeds [92].

The mathematician Georgy Voronoi proposed the algorithm in a series of two papers in 1908 [93] [94]. Its formal definition to an n dimensional space is given by equation 3.1, where m Voronoi seeds creates V_i Voronoi cells based on the euclidean distance d between the position vector $\mathbf{x}$ and the position of the seeds $\mathbf{s}_i$ and $\mathbf{s}_j$.

$$V_i = \{\mathbf{x} : \forall j \neq i, d(\mathbf{x}, \mathbf{s}_i) \leq d(\mathbf{x}, \mathbf{s}_j)\},\ \textit{with}\ i, j \in 1, 2, ..., m \tag{3.1}$$

The Voronoi pattern naturally occurs when the randomly placed set of seeds grows at a fixed rate in all directions. This is the interaction between cells, where its contact arcs' overall circular shape and size distribution can be modeled [95].

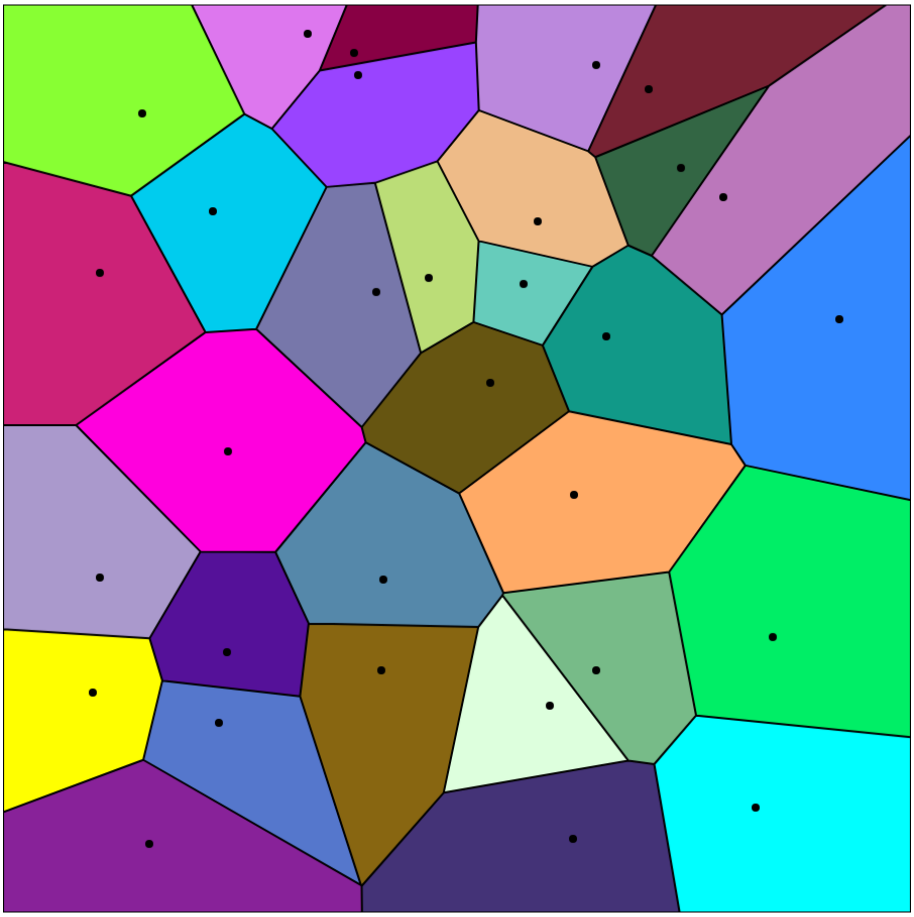

Figure 3.1: An example of a randomly generated Voronoi Diagram with 30 points.

The algorithm has also been shown to take part in animal fur pattern generation by studying the melanin producer cells randomly placed in the giraffe's body [96]. Finally, it can simulate geological processes that generate reservoir soils, critical to reservoir engineering [97].

One of the main applications of the Voronoi algorithm is the generation and characterization of porous media. Tesselations are useful in pore network models, where porous media are simplified through pores (randomly placed spheres) and throats (cylinders that connect the spheres). The Voronoi algorithm rules the distribution of throats, defining the connectivity between the pores [98]. It can also characterize porous media at the microscopic level, identifying pore sizes and shapes [99]. Also, the tesselations lead to direct representations of porous media by defining the edges between two adjacent sub-divisions as the percolating region. Therefore, creating 2D and 3D porous media for fluid flow simulation is possible by directly applying the Voronoi algorithm [100] [101].

The choice of using the Voronoi algorithm for generating the geometries of this work is due to the direct granular representation that it creates with ease. With the need to create hundreds of different geometries for direct numerical simulation, the algorithm emerges as a natural candidate for fast and accurate representation of reservoir rocks. The software used and the process of creating geometries is explained in the following topic.

3.2.3 Porespy

There are different options on the internet of software or free code to generate porous media through Voronoi tesselations. This work chose Porespy as it is a reliable, fast, and easy-to-use tool available in Python [102]. It has gained much attention from the porous media community in the last couple of years, leading to significant applications [103] [104] [105].

The *ps.voronoi_ edges* module is found in the "Generators" section of Porespy's website. It can generate Voronoi matrices in Python, allowing the user to manipulate the parameters to obtain a perfect geometry for its needs. The main inputs are the size of the system in pixels in each direction (Two or three-dimensional), the number of Voronoi seeds, and the radius of the throats (channels). It is also possible to define the location of the seeds (if not, they are randomly placed) and if the Voronoi edges will lie on the boundary of the image. The function returns a boolean matrix of the desired geometry based on the parameters provided.

The relation between porosity and the number of Voronoi seeds is shown in figure 3.2. One hundred figures are plotted for each radius, varying the number of Voronoi seeds from 2 to 200 to analyze its effect on porosity.

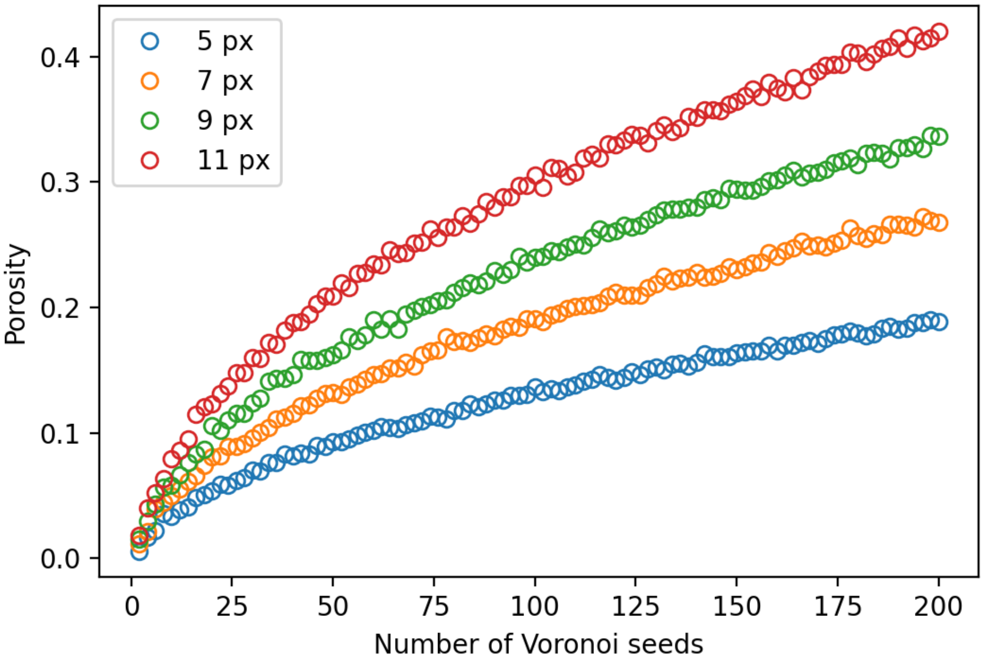

Figure 3.2: The effect of the number of Voronoi seeds in Porosity.

One thousand geometries were generated with 200x200 pixels and a radius set to 4, leading to a diameter of 9 pixels. This channel radius was chosen to be the same size as the coarser grid validated with the Poiseuille flow in Chapter 3, to guarantee hydrodynamical validity in our simulations. Then, the number of Voronoi seeds was manipulated to create a porosity range from 0.096 (5 Voronoi seeds) to 0.448 (45 Voronoi seeds).

A sample of 100 geometries randomly chosen out of the 1000 total set is shown in figure 3.3. The black portions represent the solid phase and the white represents the fluid phase, and geometries are stacked in a 10x10 grid in ascending order of porosity. It is then visually apparent that the increase in Voronoi seeds with a constant channel diameter causes porosity to rise, as the sub-divisions become increasingly small, and the white color starts to be more present than the black color.

A suitable strategy for increasing the complexity of the Voronoi diagrams is to create voids in the geometry by removing small regions. Porespy function *ps.filters.trim_small_clusters* extracts regions that have less area than a certain user-defined level. The strategy was to vary the cut area, creating void regions inside the Voronoi diagrams resembling geological failures common to rock tomographies, so an area of cut filter ranging from $150 to 300 px^2$ was applied. A sample of 100 of the 220 geometries created is stacked in Figure 3.4 to show the imperfections generated through this method visually.

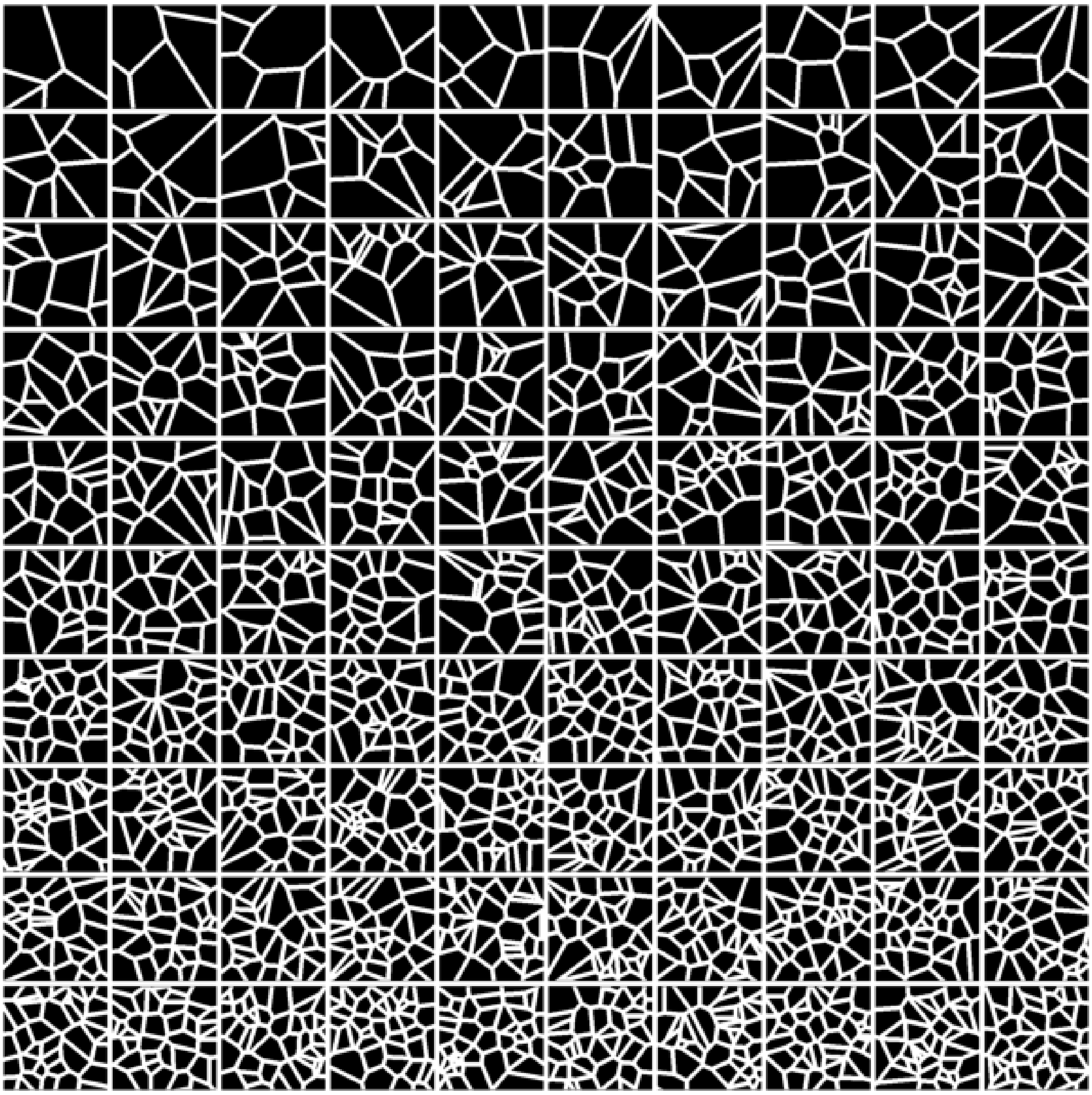

Figure 3.3: A visual representation of 100 sample geometries ordered by porosity.

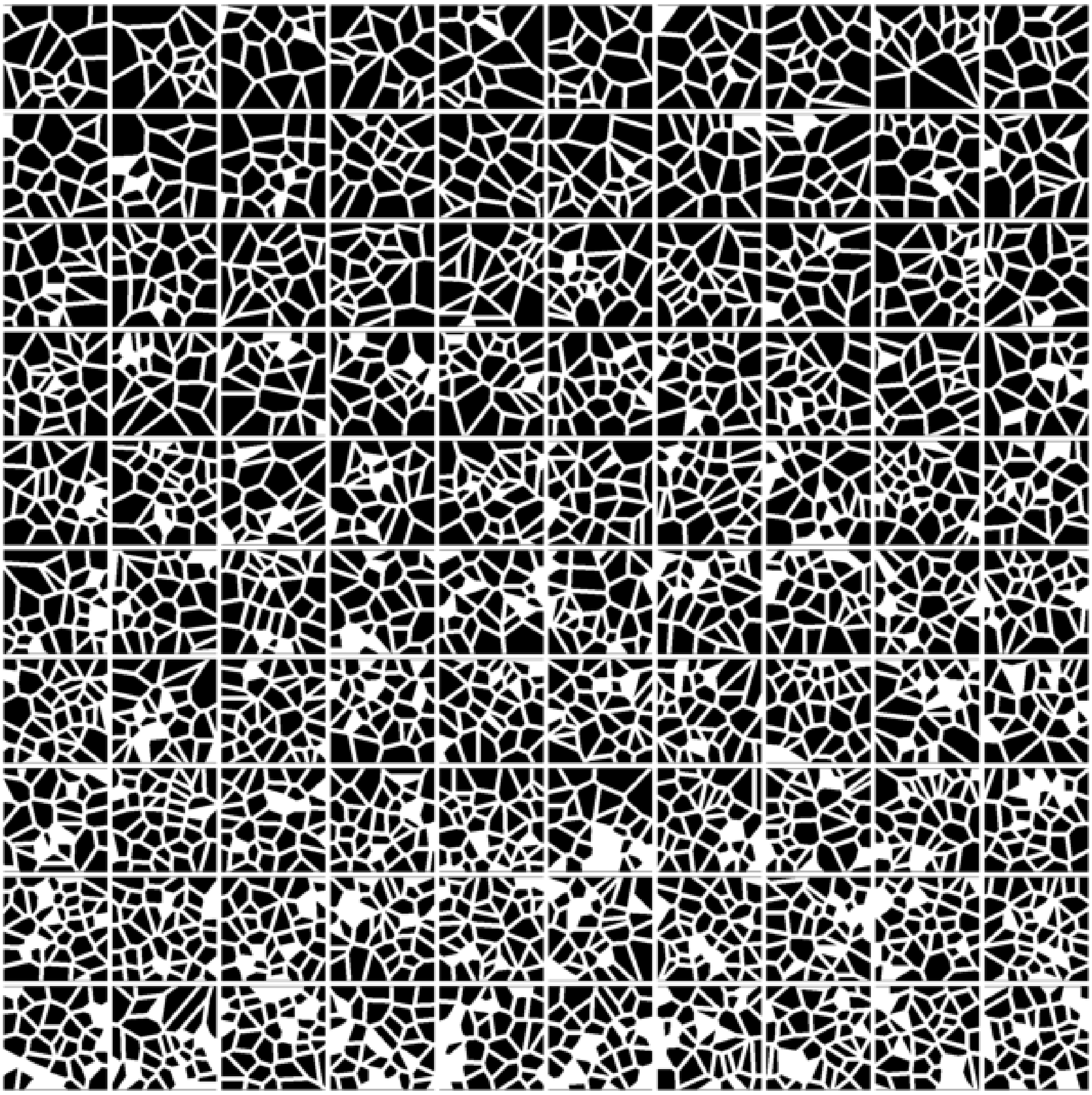

Figure 3.4: A visual representation of 100 sample geometries with voids ordered by porosity.

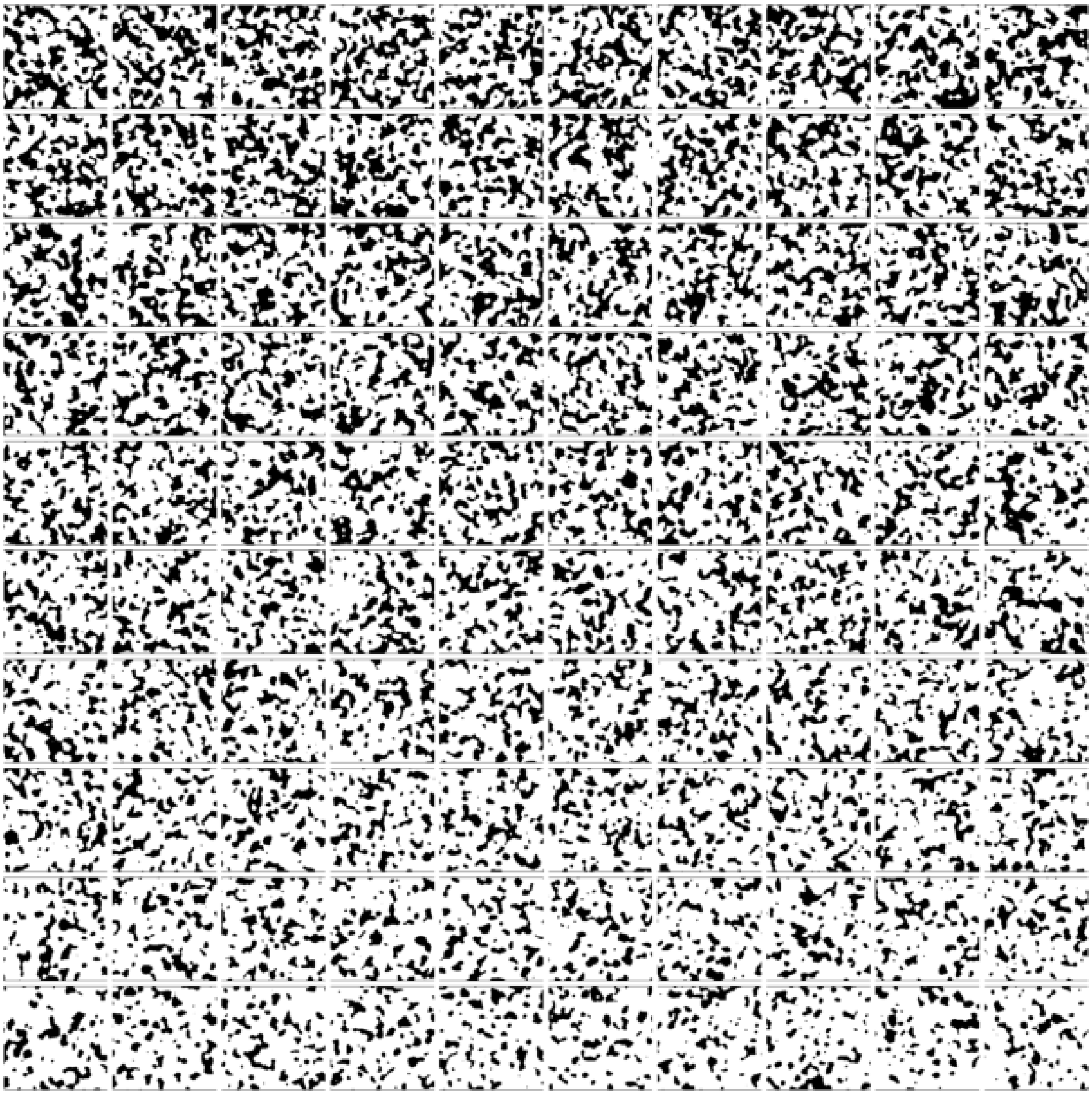

Figure 3.5: A visual representation of 100 sample geometries generated with blobs ordered by porosity.

Finally, to generate even more complex geometries, the *ps.generators.blobs* algorithm is used to create geometries completely different from the Voronoi algorithm. It takes as input the dimension of the domain, the porosity needed, and the "blobiness", a parameter that characterizes if there will be more or fewer bubbles in the final geometry. The geometries were created using $200x200$ pixels, a porosity range from $\phi = 0.6$ to 0.8, and the blobiness was set to 3. The idea is to create a set that is radically different from the previous one to increase the complexity of the Neural Networks that will close this work, as the two previous geometry generation models were too simple, which could lead to overly simplification of natural geometries. The blobs show that porosity is greater than the previous geometries, and they are much more complex than simple Voronoi diagrams (Figure 3.5).

3.2.4 Geometries characterization

Porespy has many intrinsic modules for calculating geometrical parameters. These will be briefly summarized and explained to familiarize the reader with the metrics used in the following chapter to feed the neural network.

3.2.4.1 Porosity

The *ps.porosity* module on Porespy works simply using a 2D or 3D matrix as its only argument to calculate porosity as the fraction between the liquid cells C_L and the total number of cells C_T (Equation 3.2).

$$\phi = \frac{C_L}{C_T} \tag{3.2}$$

This is equivalent to equation 1.3 in a discrete grid, as C_L represents the summation of the phase function $X_L(\mathbf{x})$ over all the porous domain. For this function to work correctly, the solids cells must be marked as 0, and the liquid cells must be marked as 1.

3.2.4.2 Tortuosity

The *ps.tortuosity_fd* function uses the binary geometry matrix and the axis concerning which tortuosity will be calculated as inputs. It applies a fickean diffusion algorithm based on the finite difference method.

The algorithm of the function starts by calculating the porosity of the medium. Then, the function *ps.trim_nonpercolating_paths* is applied to remove the paths that start in the inlet and are not connected to the outlet. Then, the effective porosity is calculated based on the new geometry without the non-percolating paths. Now, it invokes OpenPNM's function *op.algorithms.FickianDiffusion*, setting the inlet concentration as 1 and the outlet concentration as 0. The effective diffusion coefficient D_{eff} is computed as:

$$D_{eff} = \frac{\dot{N}(L-1)}{A\Delta C} \tag{3.3}$$

Where $\dot{N}$ represents the mass rate inlet, L the size of the domain in the direction of flow, A the area of the domain (Volume for a 3D domain), and ΔC the concentration delta, in this case, set to 1. Then, the hydraulic tortuosity T_H may be calculated as:

$$T_H = \frac{D_{AB}}{D_{eff}} \phi_{eff} \tag{3.4}$$

The ratio $\frac{D_{AB}}{D_{eff}}$ is also known as Formation Factor.

3.2.4.3 Specific surface area

The specific surface area in a discrete grid is calculated as surface area cells over the liquid volume cells:

$$A = \frac{C_{surface}}{C_L} \tag{3.5}$$

Where $C_{surface}$ represents the surface cells in contact with the solid. However, the determination of the surface is not simple. As Porespy does not have a specific function to calculate A, we have implemented a code that counts the bulk cells in contact only with fluid cells in the x and y-axis. It's a loop on every cell, seeing if the upper, lower, right, and left cells are fluid. The cell is marked as bulk cell C_{bulk} if all conditions are satisfied. Then, the surface cells are counted as $C_{surface} = C_L - C_{bulk}$. Note that the diagonal cells do not need liquid to mark the primary cell as the bulk.

3.3 Application II: Darcy Law

The last characterizing parameter left for these geometries is permeability, as it is the main task of this text. The lattice Boltzmann method is chosen to perform direct numeric simulation over the geometries created, and then permeability is given through Darcy's law as the relationship between the imposed velocity and the resulting pressure gradient. However, to ensure the correct application of Darcy's law and the permeability values calculated, it is necessary to guarantee that the simulations are carried in laminar flow.

One random porous geometry was chosen to perform 10 simulations at different imposed velocities, calculated according to the inputted Reynolds number to show that the Darcy hypothesis is valid. The range of $Re = 0.1$ to 1 is simulated with a geometry with $0.001m$ of channel diameter, as these values are closer to the limit of laminar flow in porous domains as described by Arora (1989) [18]. The results are shown in Figure 3.6.

The linear relationship found correctly emulates the linear behavior expected in Darcy's law: Permeability behaves as a constant at these Re, calculated as the angular coefficient of a $R^2 = 0.9998$ line. This result is crucial as it shows that for this range of Reynolds numbers (Or lower), the flow regime is guaranteed to be laminar at this geometry [60].

This is not necessarily valid for all geometries. It would be necessary to perform the same experiment with different Reynolds numbers with all 1.000 geometries to guarantee that they all correctly reproduce laminar flow and the Darcy hypothesis. However, as this procedure would increase the time consumed by the simulations at least by a factor of 10, for the scope of this work, this result is expected to

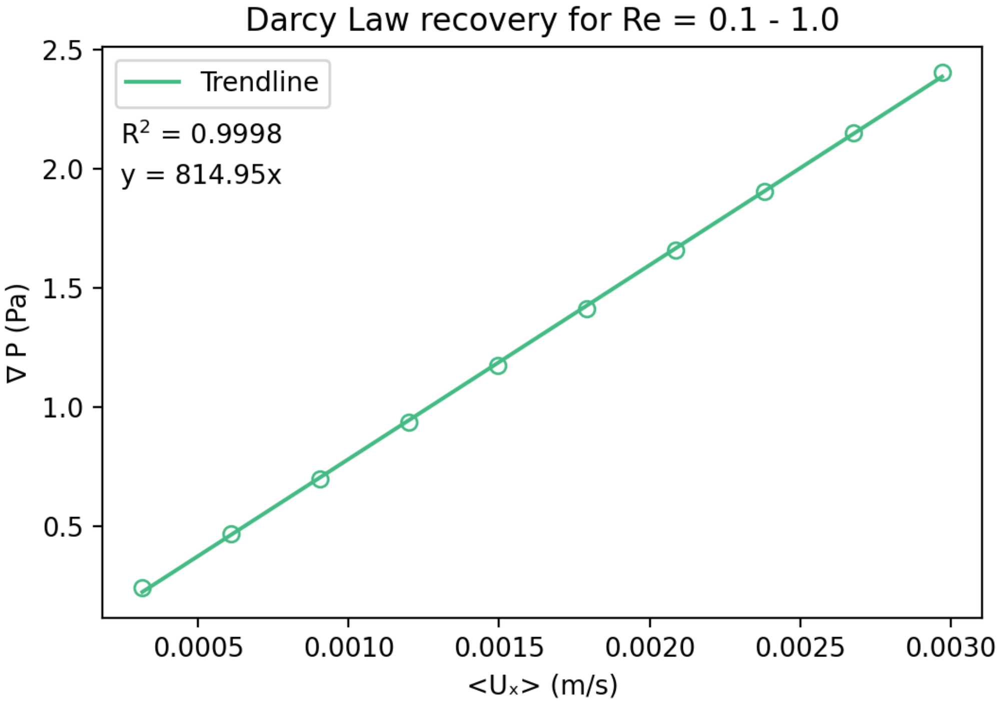

Figure 3.6: The resulting pressure gradient is plotted as a function of the average velocity in the flow direction.

be extrapolated for all artificially generated geometries. Then, from this point on, all simulations were performed only once with Re = 0.003, and permeability is a constant given by the ratio of μu_x and $\rho \nabla P$.

Considering that the simulations are carried in the Stokes regime, permeability results for 771 geometries are shown in Figure 3.7.

Figure 3.7a shows permeability as a function of porosity. These two properties exhibit a linear relationship, which follows the expected physical behavior as the flow should be facilitated as void increases. Eshghinejadfard (2016) [89] remarks that permeability is very sensitive to porosity fluctuations, explicating this relation using FCC and BCC-packed spheres.

Figures 3.7b and 3.7c show permeability as a function of tortuosity and the formation factor, respectively. In both cases, permeability is inversely proportional to the quantities, even though the values show different behavior for tortuosity and the formation factor. In respect to tortuosity, the values are more scattered, while for the formation factor, the fall is smooth. This may be explained through the relation between the formation factor and tortuosity, the effective porosity will affect the relation between tortuosity and permeability, making its relationship more mathematically complex than the ratio of D_{AB} and D_{eff}.

Figure 3.7d relates specific surface area and permeability, exhibiting a linear correlation. This may seem strange initially, as the increase in surface area could

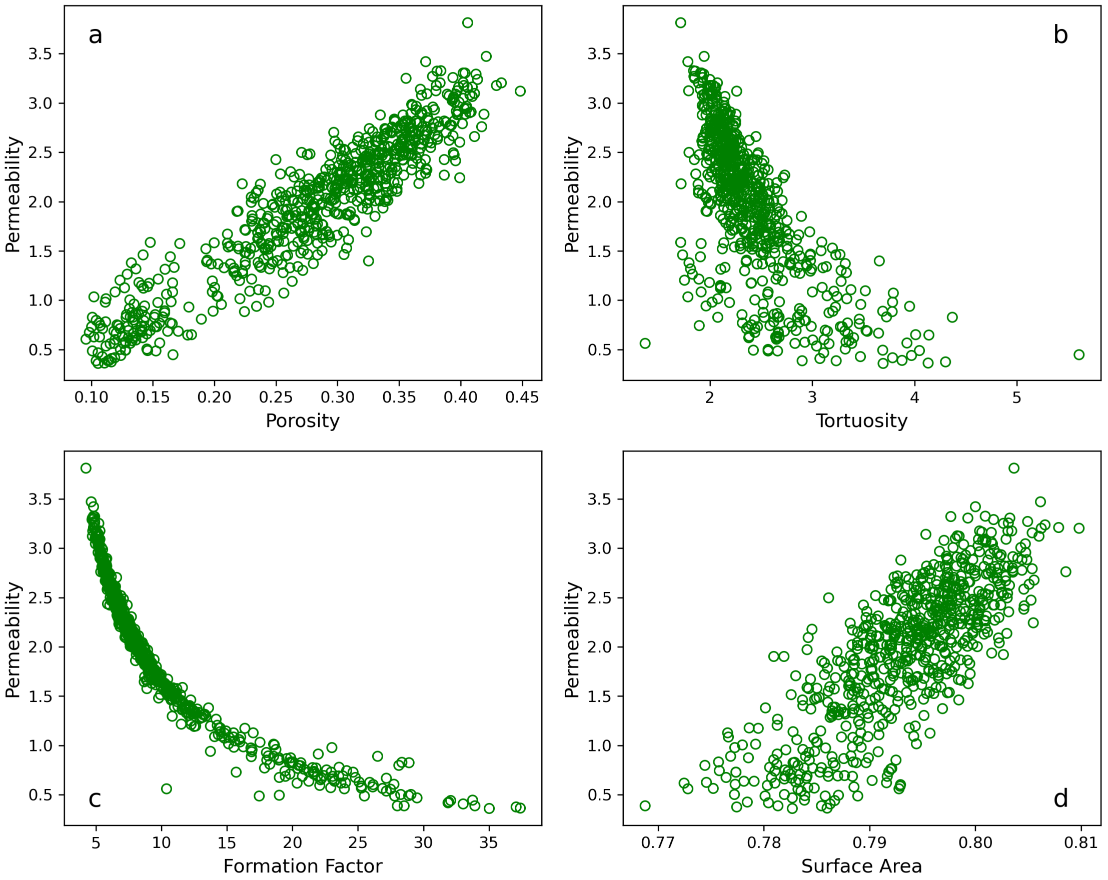

Figure 3.7: Permeability is plotted against the characterizing parameters for simple Voronoi. a) Porosity, b) Tortuosity, c) Formation factor and d) Surface area.

diminish fluid flow through viscous forces, but it seems logical if the relation with porosity is observed. Here a disclaimer is advised, due to the nature of the Voronoi algorithm as implemented, as throat radius remains constant in all geometries, the surface area will be directly proportional to porosity. This may not be true in all scenarios for more complex or even for 3D geometries.

Then, the simulations are performed with the set of geometries that contains voids. It is possible to see in Figure 3.8 that we had a general increase in permeability values as the porosities were higher and that the general tendencies for the four parameters are the same even if more scattered. The increase in complexity had a remarkable effect on porosity, diminishing its linear behavior and creating an exponential tendency similar to the Kozeny-Carman relation's predictions for high porosity values. Tortuosity and the formation factor remained with their tendencies, even if associated with higher permeability values. Finally, the specific surface area showed a significant increase in range, leading to higher permeability values than the simple Voronoi diagram samples.

Finally, the blobs are used as input for the simulations, and the results are

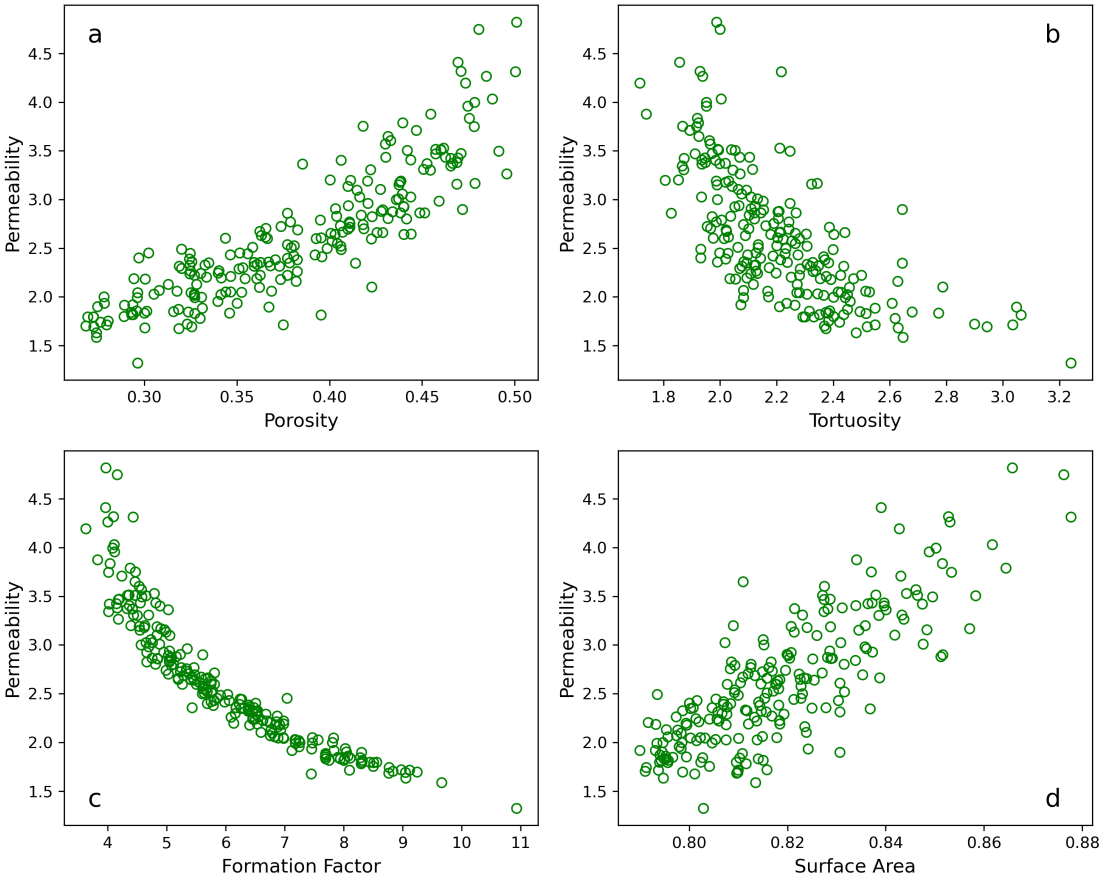

Figure 3.8: Permeability is plotted against the characterizing parameters for Voronoi with voids. a) Porosity, b) Tortuosity, c) Formation factor and d) Surface area.

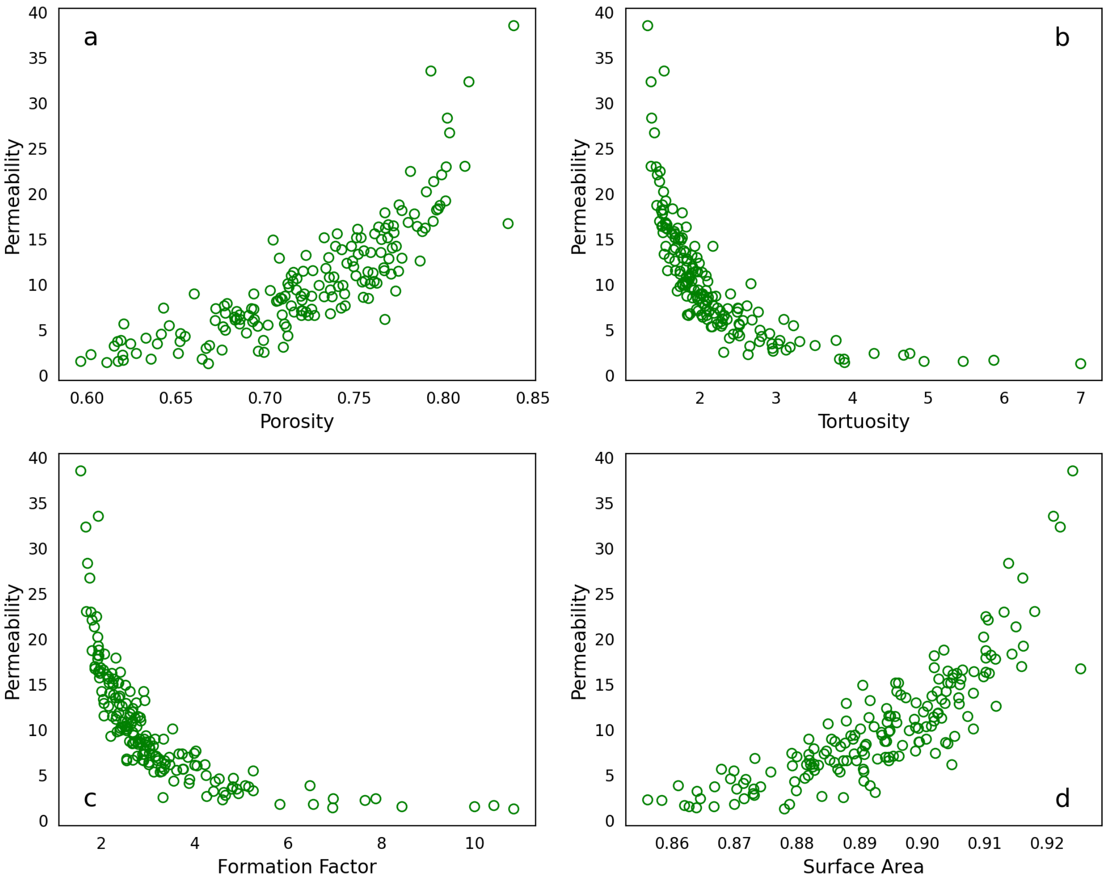

Figure 3.9: Permeability is plotted against the characterizing parameters for blobs. a) Porosity, b) Tortuosity, c) Formation factor and d) Surface area.

summarized in Figure 3.9. It is possible to see that the parameters follow the same tendency as before. with the permeabilities being, in general higher than for the previous geometries. This is expected as the porosity-permeability relation has been successfully established so that more porous geometries will lead to higher permeability values. The significant differences lie in the effect of tortuosity and formation factor data, which are more prominent as they increase, exhibiting lower values for permeability. This may be explained as high tortuosity values at these porosities leading to slower permeability values since they are related to tighter packings and less porous geometries.

These results combined show the capacity of the lattice Boltzmann method in dealing with complex geometries and calculating permeability values according to Darcy's law for low Reynolds numbers. These data will be used as input in the next chapter for the Neural Networks analysis, aiming to predict permeability directly from the characterizing parameters along the images.

Chapter 4

Artificial Neural Networks

4.1 Introduction to machine learning

To meet the last specific objective, analyzing the data generated through the LBM simulations is necessary. The data obtained until this point are the artificially generated rock geometries and their characterizing parameters, porosity, tortuosity, specific surface area, and permeability. To predict permeability, this chapter aims to correlate these data types (Images and numbers) and create a tool to predict permeability from the images, porosities, tortuosities, and specific surface areas. It is then necessary to choose an adequate artificial intelligence model that can solve the problem of handling the different types of data available.

The terms artificial intelligence, machine learning, neural networks, deep learning, and Convolutional Neural Network correlate. Still, despite being misused as synonyms, they are subsets of each other (Figure 4.1). Machine learning is a class of models within the significant field of artificial intelligence. Neural networks are a class of architectures for machine learning, among many others, and deep learning refers to neural networks with multiple hidden layers. Finally, convolutional neural networks are a subset of deep learning algorithms capable of interpreting image data and identifying patterns in the images. This makes this model suitable for analyzing the simulation data composed of images and parameters.

Artificial intelligence was first proposed by John McCarthy in 1955 as an independent field of study at the Dartmouth Summer Research Project on Artificial Intelligence, held in the summer of 1956 in Dartmouth College [106]. Notably, a program called *Logic theorist* was presented, capable of proving mathematical theorems upon theorizing upon mathematical basic operations. Today, it is a significant area of computer science that aims to use mathematical tools and algorithms to perform tasks that would otherwise need human thinking.

Machine learning was a term first used by Arthur Samuel in 1959, when he

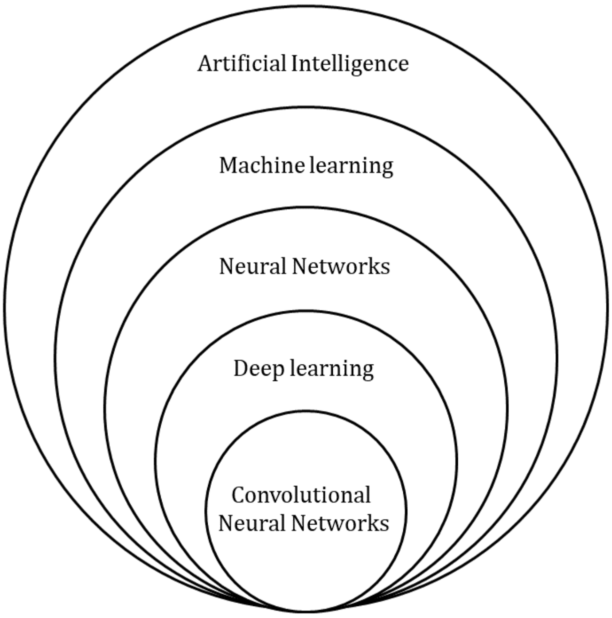

Figure 4.1: Different terms in the Artificial Intelligence terminology are shown as subsets from one another.

studied computer models to play checkers that could learn from past games [107]. He created a model that, in 1962, defeated Robert Nealey, the so-called "checkers king" at the time. In a general sense, machine learning is the process of teaching machines how to think. It is considered a significant field within artificial intelligence in which algorithms can learn and generalize from data, performing tasks without being explicitly programmed.

The idea of machine learning is to use an algorithm to reproduce the human learning process, beyond all, being able to generalize knowledge. Machine learning algorithms are instrumental in data science to handle and understand big datasets, as the human mind is often incapable of dealing with them [108]. Inside the machine learning domain are supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, ensemble learning, instance-based learning, and neural networks. Each of them is suitable for a different task, and only the latter lies in the scope of this text. Other works should be consulted for the interested reader, such as the book by Aurélien Geron (2022) [109].

Neural networks are a subset of applications within machine learning that aim to reproduce the thinking process in the human brain. A feedforward neural network is a complex graph structure [110]. A simple representation of a general neural network architecture is shown in Figure 4.2. It is organized in layers containing multiple neurons, the neural processing units that perform mathematical transformations upon the information from the previous layer. Each neuron of a layer is connected to the neurons of the following layers through weights, which define how much of

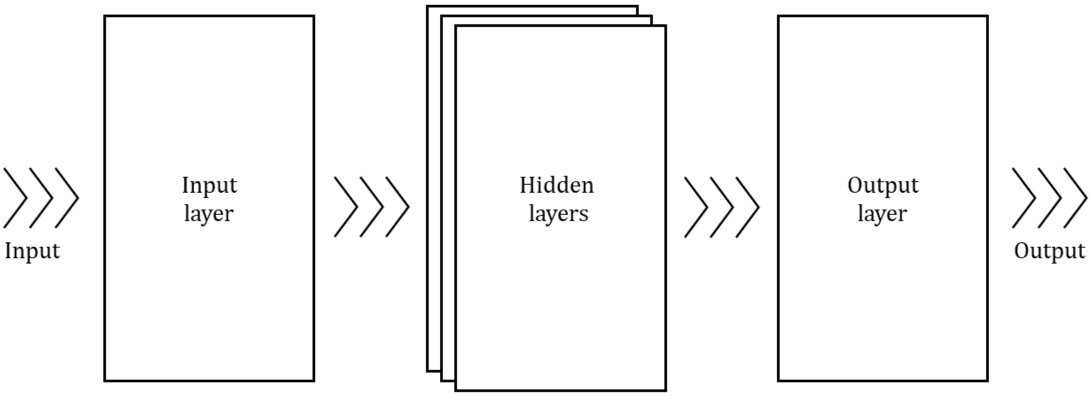

Figure 4.2: A general architecture for neural networks.

the calculated value will be passed forward. All neurons converge to results in the last layer (output layer), and the error between the calculated and expected results is calculated. Then, it starts a feedback process that recalculates the weights to minimize error. Through the loop of this feedforward-feedback process, the weights are recalculated multiple times until the output error is reduced below a user-defined threshold.

Deep learning is a class of neural network architectures that takes advantage of multiple hidden layers to extract information through different levels of representation, decomposing an input into higher forms of abstraction and allowing feature recognition [111]. Deep learning can deal with unlabeled datasets and understand data in its raw form, eliminating human intervention and enabling more significant amounts of data of different types. This feature is handy in classification tasks, as higher layers of representation may generally offer details hidden at first sight. It has consolidated deep learning as a significant image and speech recognition tool.

Convolutional Neural Networks is the architecture category within deep learning that has been more extensively studied [112]. It is inspired by the natural image perception of living organisms, so it is especially suitable for computer vision, speech, and natural language processing. CNN's modern architecture was first proposed by LeCun (1998) [113], obtaining great success in an attempt to identify handwritten digits (Usually referred as MNIST dataset).

A general CNN architecture contains convolutional layers, pooling layers, an activation function, and a fully connected layer (Figure 4.3). Each layer has a different objective and a different position in the architecture. Generally, pooling

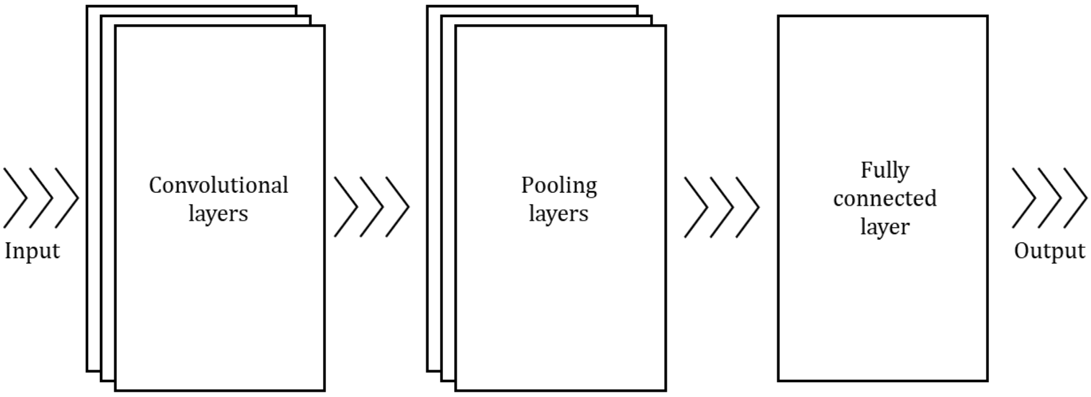

Figure 4.3: A general architecture for convolutional neural networks.

layers follow convolutional layers in pairs, and only after a sequence of convolutions and poolings, do the values pass through a fully connected layer that outputs the final value calculated. These layers are connected by activation functions that dictate the dynamics of information flow between the neurons.

The convolutional layers are the most essential part of the CNN architecture, giving its name to the method. At this layer, a convolution operation is performed over the input with randomly initialized kernels, which can be seen as design filters over the image. At each new image input, these kernels are recalculated, being able to extract specific features of the images inputted after adequate training, for example, a filter that blurs the image everywhere except in the extracted feature region. The kernel performs operations in a high-dimensional implicit feature space, capable of turning a linear model into a non-linear model [114]

The activation functions add another layer of non-linearity to the model. It allows the convolution outputs to be saturated or terminated based on their values, which enables CNN to perceive complex and sophisticated patterns. These functions can be essentially any function going from -1 to 1, but some choices have been consolidated as the most effective, such as ReLU, Sigmoid, and Tanh.

Then fully connected layer gathers all information previously passed and calculates the final output values. It is crucial, as is in this layer, that the algorithm calculates the error between the solution and the output calculated. This is the moment of backpropagation, recalculating the weights until the training can obtain a satisfactory result.

The training step is followed by a validation step, where the model uses a separate subset of the data initially fed to the model to evaluate its results and adjust the parameters. Once validation is complete, a separate test set, not yet fed, is used to verify if the model can be used for real world applications.

4.1.1 Applications in industry

Two of the most notable conquers of CNN are speech recognition, natural language processing, and computer vision [115]. CNN properties of locality, weight sharing, and pooling make it sound at handling audio. It is less affected by non-white noise, reduces over fitting due to multiple repeated frequency bands, and can deal with natural frequency shifts of the human voice [116].

Advances in image processing due to CNNs led to computer vision, which has brought advances in multiple areas. In medical sciences, computer vision has found use in identifying body parts, lung texture and patterns analysis, thyroid nodule diagnosis, breast cancer, and diabetes identification, among others [117]. It has also linked image data and pattern identification that led to the almost instantaneous response times needed for robot vision applications, such as identifying cracks in pipes [118]. CNN has also shown great potential in modern facial recognition, that roots in image recognition associated with the semantics perception of human expression [119].

Computer vision has also found use in porous media applications. One of the most immediate applications is the determination of properties such as porosity, coordination number, and average pore size from 2D greyscale micro-CT imaging of actual rock samples [120]. CNNs can also be applied directly to transport problems, such as the determination of the dispersion coefficient, which was able to establish the link between the coefficient and the morphology of the pore space [121]. Beyond that, the area of physics-informed neural networks had used CNNs to perform direct numerical simulation of fluid flow on porous media, performing the task of predicting permeability in a matter of seconds versus the traditional simulation times of DNS, which can take hours or even days in supercomputing facilities [122]. Besides, after the model is trained, it can be used for any different porous geometry, instead of having to perform another costly simulation, it can output an almost instantaneous result based on image recognition. Finally, it is also capable of predicting correct permeability values for geometries more complex than those used in the training set, showing the capability of the model to generalize knowledge and make connections between various cases.

4.1.2 LBM applications

The rock images dataset must be labeled with the permeability values to perform the training of recognizing permeability values from rock images. Thus, before the training, the permeability values for a representative sample of images must be calculated. As the lattice Boltzmann method has already been shown as the ideal tool for these calculations, some studies have been done to couple LBM with CNNs.

Wu (2018) [37] has successfully predicted permeabilities with a CNN trained with lattice Boltzmann data and porous geometry characterizing parameters. It was shown that permeability is a geometric-dependant parameter and that CNN achieved better predictions than the Kozeny-Carman relation. It was also shown that the characterizing parameters of porosity and specific surface area, when inputted together in the fully connected layers scheme, can achieve more accurate predicted values, even if the increase in geometrical complexity leads to worse predictions. The study's main idea was to show that the potential of CNN informed with geometric parameters of the images could obtain better permeability results than a well-consolidated equation in the literature.

Wang (2021) [123] used a different approach, not feeding the model only with raw geometry images but with the resulting steady-state velocity field images, then asking the model to reproduce new velocity fields in raw geometrical images previously not seen. The study shows that it can accurately predict permeability values as domain averages for simple geometries, but the pressure and velocity fields predicted by the model do not necessarily match the calculated through LBM, showing that the fluid flow prediction's accuracy was increasingly dependent on the complexity of the geometry. The advection-dominated solution at high Peclet numbers, even in the simplest geometries, lacked voxel-by-voxel accuracy showing the steady state velocity fields predicted had no further use.

Yalamanshi (2024) [124] used SEM images of geological basins in India as inputs. These images were carefully characterized in relation to the pore size distribution, the grain size distribution, the throat radius, and the 2D coordination number. Then, permeability is calculated through Darcy's law after the steady state is reached with a BGK-LBM scheme. It is shown that the porosity, the grain size distribution, and the throat radius are the main parameters affecting permeability, while pore size distribution and the coordination number showed almost no impact on permeability calculation. The study showed that CNNs may be applied to natural reservoir rocks, which contain much more intrinsic complexities than the artificially generated geometries, obtaining a good agreement between the model and the predicted permeability values and those calculated with LBM.

4.2 Theoretical foundations

4.2.1 Mathematical description

The basic mathematical description of a neural network may be seen through the function of a single neuron. It may be described as a functional mapping between the input matrix $\mathbf{q}$ and the output matrix $\mathbf{y}$:

$$\mathbf{y} = W^{(2)}\sigma(W^{(1)}\mathbf{q} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)} \tag{4.1}$$

Where coefficient vectors $W^{(1)}$ and $W^{(2)}$ represent the functional mappings, $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ are their associated biases and σ represents the activation function.

The activation function is the mathematical entity that calculates the output based on the inputs and the weights. Cybenko (1989) was able to show that a finite linear combination of Sigmoid functions was capable of approximating any given function [125]. Neural Networks use this result to perform complex generalizations, as the multiple neurons correspond to the superposition of different activation functions. Common activation functions are Sigmoid (or logistic), ReLU, GELU, hyperbolic tangent, Softplus, and exponential.

The validation of the model was performed following the Mean Squared Error criteria:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(k_i^{CNN} - k_i^{LBM})^2 \tag{4.2}$$

The k_is are the permeabilities predicted by the CNN and the direct simulation via LBM, respectively.

4.3 Methodology

4.3.1 Architecture used

The architecture used mixed classic Convolutional Neural Networks for image analysis and Deep learning for numerical data analysis, joining them through densely connected layers at the network's end. This approach is advantageous as it allows the user to manipulate image and numerical data separately, facilitating the importance analysis without losing the information the images give.

The number of layers and neurons was optimized regarding MSE to create an architecture that efficiently solved the problem minimizing computational cost.

4.3.2 Implementation tools

The framework chosen for the numerical procedure was the *tensorflow*, *keras*, and *scikitlearn* packages.

Tensorflow was first proposed by Abadi, et al (2016) [126] in the 12th USENIX Symposium on Operating Systems Design and Implementation as a toolkit for solving matrix operations for machine learning. As TensorFlow is inspired by high-level dataflow systems and the low-level efficiency of parameter servers, it can represent both the operation and the state in which the algorithm operates. By unifying state management, experimentation with parallelization schemes becomes more accessible for the final user. Due to that, Tensorflow has experienced massive success as a machine learning framework currently used by most state-of-the-art companies and research groups.

Scikit-learn is a Python interface with trustful implementations of standard machine learning algorithms [127]. Its main advantages include code efficiency, as it relies solely on Numpy and Scipy for calculations and is focused on imperative programming, differing from the dataflow framework.

Keras is a famous Python machine-learning package used as the primary implementation tool in this work. It is direct and clean to code and has been extensively studied, being the topic of various books [128] [129] [130]. Keras was also reviewed against other commercial Python frameworks for machine learning, such as PyTorch and MXnet, obtaining a better overall performance [131].

The popularity of Keras led it to be used in different research areas. Keras has been used for real-time sentiment analysis on Twitter, reading and classifying tweets based on positive or negative feelings [132]. Keras also easily handles massive datasets, used on the transfer learning of cloud street image data due to tuning its hyperparameters to avoid excessive calculations [133]. Keras has also outshined other software and showed its strengths in satellite image analysis for traffic detection, detecting vehicles with almost no misclassification [134]. Then, it has been consolidated as a good choice for machine-learning implementation, especially for convolutional neural network architectures, so it is naturally chosen as the main library for neural networks used in this work.

4.3.3 Preparation of data

To upload the data in the model, the *keras.preprocessing.image.load_ img* function loads the labeled *.png* images of the artificially generated geometries and the *pandas.read_ csv* function reads a *.csv* file that contains the parameters associated with each label. Then, the model can associate each geometry with a previously calculated value of porosity, tortuosity, specific surface area, and permeability.

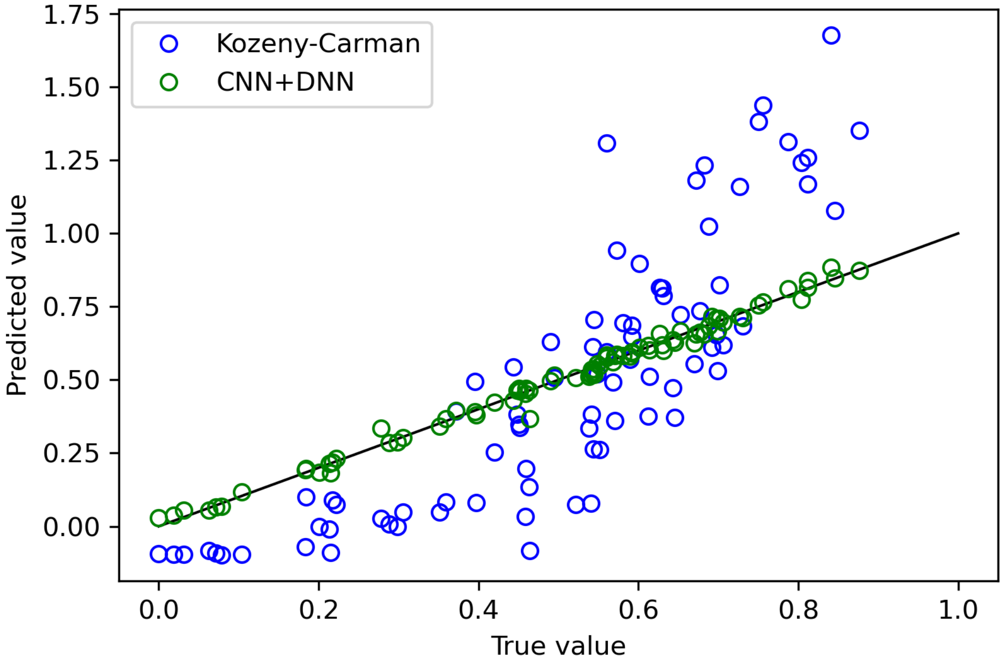

Figure 4.4: The main permeability prediction results are shown versus the Kozeny-Carman equation.

4.3.4 Validation and sensitivity analysis

After validation, a sensibility analysis is generally carried out to better understand the influence of each parameter inputted. It consists of inducing an error in each of the parameters and tracking how its variation influences the final output, with the most sensible parameters being those with the most importance.

4.4 Results

After the training is complete, it is possible to summarize the permeability prediction results for a model trained only with simple Voronoi geometries in Figure 4.4. The permeabilities calculated with the CNN+DNN model were much closer to the actual value than those predicted by the Kozeny-Carman equation, showing that the model is generally better than the correlation.

Then, a sensibility analysis is performed to evaluate the model's performance (Figure 4.5). As expected, porosity is the most critical parameter, leading to significant errors when perturbed. Tortuosity and formation factor plays a minor role, but are also responsible for correctly calculating the permeability values. The specific surface area and the image are parameters not important at all, as their influence on the perturbation error is shown to be very small. This is interesting, as the main

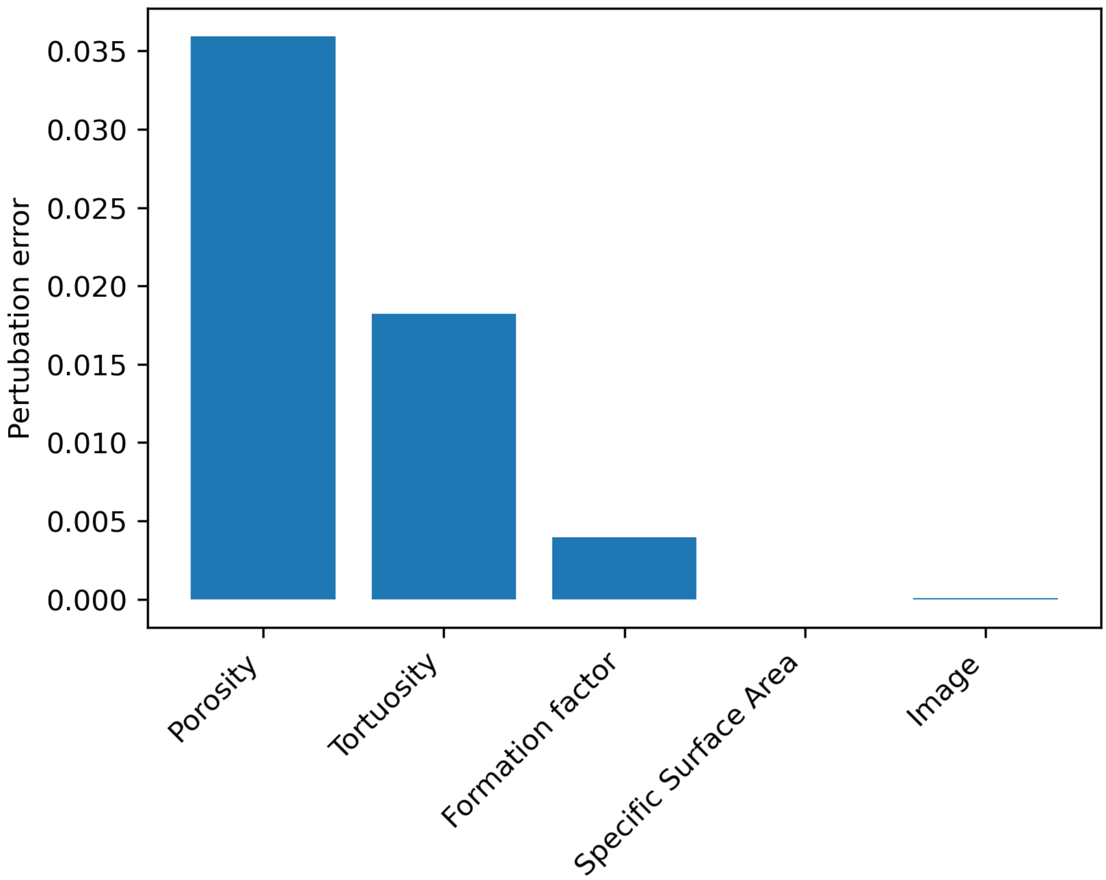

Figure 4.5: The main characterizing parameters are shown in order of importance to permeability prediction.

idea of the work consisted of predicting permeability from images, but this test shows that to predict permeability values in a simple Voronoi diagram scheme, the geometry characterization parameters of Porosity, Tortuosity, and Formation factor are sufficient. It is possible to conclude that simple Voronoi diagrams can calculate permeability without the proper images, using only a less costly Deep Neural Network.

It is interesting to see how the trained network responds when fed with data that differs from what was previously trained. The Voronoi with voids and the Blobs datasets are fed to the model as inputs. The results for the permeabilities of these geometries are shown in Figure 4.6. The model can predict permeability with a certain degree of accuracy, even though the values are slightly higher than the reference LB values. However, the model could not predict permeability values accurately for the blob geometries. This might be explained as the blobs are severely different from the Voronoi geometries, presenting curved shapes that the Network had not spotted before. Also, blob porosities were generally higher than Voronoi's, leading to increased permeability values.

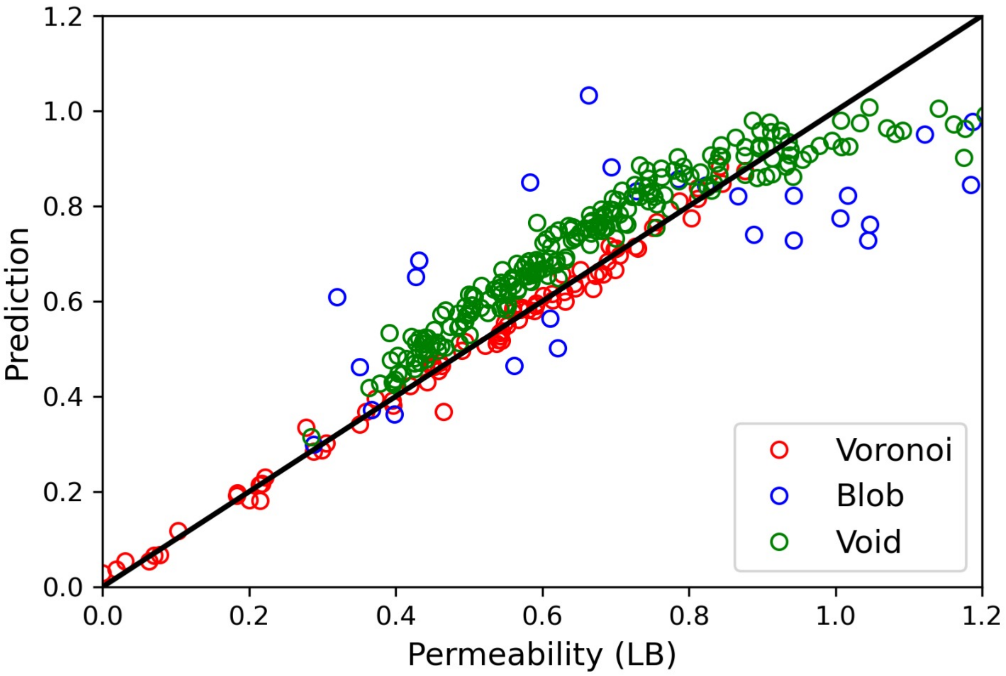

Figure 4.6: The permeability values predicted by a network trained only with simple Voronoi data for three different sets of geometries.

Then, the strategy is to retrain the network using the Voronoi with voids and Blobs as input, so the network is now presented to a different range of porosities, permeabilities, and geometries. This is fundamentally important as the model is generally only able to predict values regarding what it has been previously shown, so it was expected that it could not predict Voids and Blobs when only trained with simple Voronoi geometries. The prediction results for the new network trained with all the geometries are shown in figure 4.7.

The $R^2 = 0.949$ indicates a good agreement between the predictions and the actual permeability results. Notably, the values are much more scattered in this prediction as the range of permeabilities increases with the addition of the Blobs geometries. Also, as there were fewer Blobs than Voronoi geometries, there is an accumulation of results of lower permeability and a lack of higher permeability values, as shown in the parity plot. A sensitivity analysis is conducted to verify the influence of the characterizing parameters for the new network (Figure 4.8)

Interestingly, in this case, all of the parameters exhibit some importance, differing from the case of the pure Voronoi geometries where the specific surface area and the image were almost negligible. Particularly in the case of specific surface area, it has become the most critical parameter for the new network, followed by tortuosity and only then porosity. This may be explained since, in simple Voronoi geometries, the relation of porosity and permeability is somewhat direct, while it is not perfectly

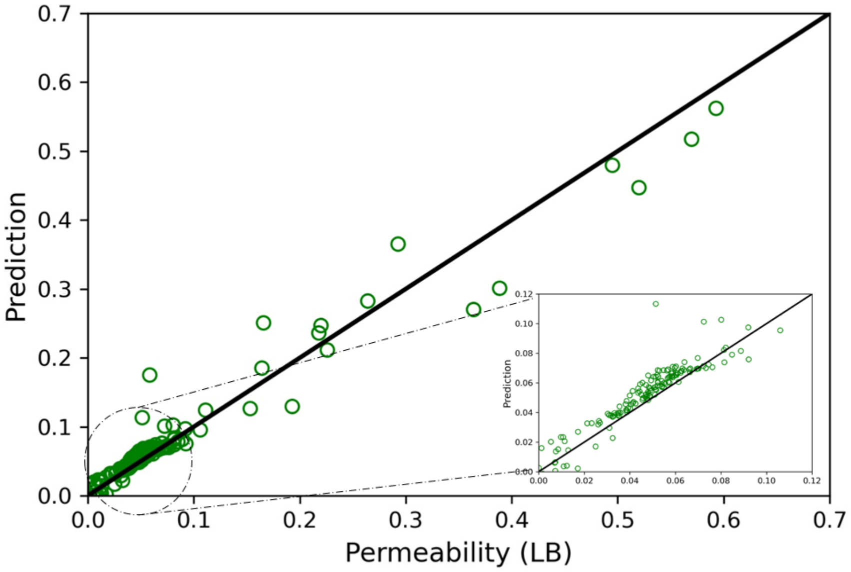

Figure 4.7: The permeability values predicted by a network trained only with simple Voronoi data for three different sets of geometries.

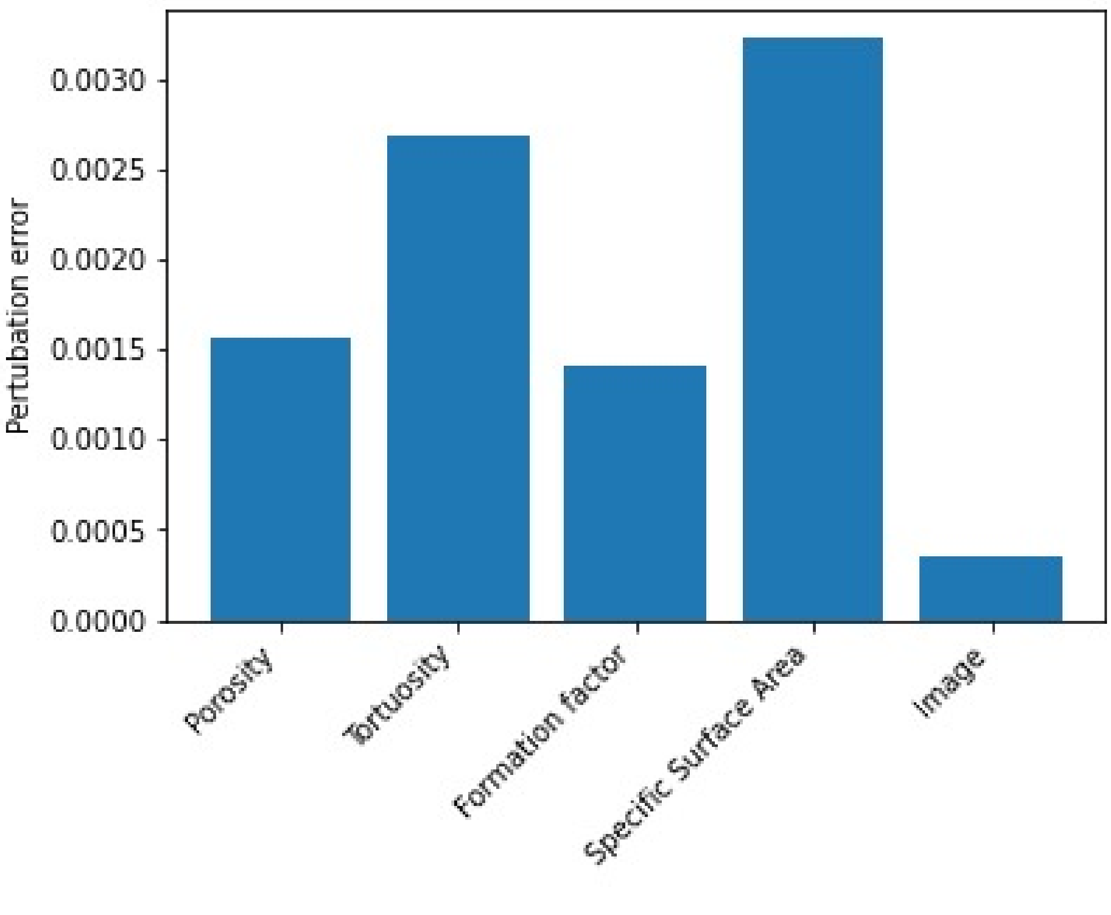

Figure 4.8: The permeability values predicted by a network trained only with simple Voronoi data for three different sets of geometries.

linear for the other geometries. Also, the increase in complexity due to voids and blobs may explain why the specific surface area became so crucial concerning the other parameters, as in simple Voronoi, it is directly proportional to the number of seeds, with a constant channel radius, which is not valid as complexity increase.

Chapter 5

Conclusions

5.1 Results overview

The results of the previous chapters can answer this work's fundamental question: Is it possible to train neural networks to obtain high-fidelity permeability results in direct numerical simulation without the high computational cost of this technique? The answer is yes, given the assumptions made, the neural networks were successful in the task of predicting permeabilities while offering valuable insights about the relative importance of each characterizing parameter.

To corroborate this answer, it is possible to review the specific objectives proposed on the first chapter. First, theoretical knowledge should be constructed using the mathematical tools of the kinetic theory of gases and the lattice Boltzmann method, which was carefully reviewed in chapter two. The mathematical basis of the kinetic theory of gases, defining the primary variables, explaining the construction of the Boltzmann Transport Equation, and the discretization process to obtain the Lattice Boltzmann Equation. The solid theoretical basis gained has proven to be a fundamental tool during all phases of this research, aiding the interpretation of results concerning correct physics.

Then, a lattice Boltzmann algorithm is proposed to solve fluid flow problems,validated to guarantee hydrodynamical consistency. The algorithm used was detailed in chapter three, focusing on implementation and applications, and the Poiseuille flow is chosen as a benchmark problem for hydrodynamical accuracy as it has been recommended by the specific literature. A grid convergence test confirms that the model correctly approximates the analytical result for grids until a nine-pixel channel radius, which will then be used as the channel radius for the porous media artificially generated. The model is then validated for hydrodynamical simulations and shall then be exposed to different geometries to calculate permeability through Darcy's law.

The algorithm objective is to predict permeability through direct numerical simulation, so a set of artificially generated rock geometry samples are generated and characterized for this further use. The python libraries *Porespy* were tested for creating and characterizing the geometries, as through its simple interface, it is possible to create hundreds of Voronoi diagrams with varying porosity and to extract cells creating void spaces that add complexity to the geometries. This leads to an increase in their characterizing parameter diversity, specifically the specific surface area. Later, blob geometries were also created to increase geometric complexity, leading to a more diverse pattern of permeabilities for the Neural Network to analyze.

To guarantee that permeability may be calculated through Darcy's law, it is necessary to ensure that the simulations are carried out in the Stokes regime, and only then calculate the permeability for all rock geometries. The laminar regime is proven through experimentations with the Reynolds number, showing that the pressure gradient followed linearly an increase in velocity. Then, assuming that the Darcy hypothesis is valid for all simulations, direct simulations are carried out upon 771 traditional Voronoi geometries, 220 Voronoi geometries with voids and 173 blob geometries. The results have shown that the complexity increase due to the insertion of voids did not significantly affect the general properties, showing a slighter increase in porosity and a significant increase in the specific surface area. Finally, the change in the geometry model raised the values for porosity, as the blobs were more porous than the previous Voronoi geometries. However, the general tendency of permeability concerning the characterizing parameters is maintained, being proportional to porosity and the specific surface area and inversely proportional to tortuosity and the formation factor.

Finally, permeabilities are predicted directly with a joint CNN+DNN model trained to predict permeability based on image input and the geometrical characterization inputs. The first set of images to train the model was the simple Voronoi geometries, where permeability is directly related to porosity. The sensibility analysis confirms this relationship between porosity and permeability. Also, it shows that the Specific Surface Area and the images were not necessary to calculate permeability, evidencing the potential of the DNNs without the need for costly convolution operations. The model could predict better results than the Kozeny-Carman equation, showing its capabilities of outperforming traditional methods for determining permeability based on porous media characterizing parameters.

The Voronoi with voids and Blob sets are then used as input on the previously trained network to test its capabilities of predicting permeabilities of geometries that differ from the ones on the training set. It was found that the model could still accurately calculate permeability for the Voronoi with voids input, but could not correctly predict permeabilities for the Blob set. This is due to the significant irregularities in blobs, such as curves and unconnected regions, besides their higher porosity and permeability values. The network is then retrained with the Voronoi with voids and Blobs, able to predict permeability values for all images with significant accuracy correctly. However, the importance of each parameter changes drastically, now with the specific surface area being classified as the most critical parameter on permeability prediction, and the images are also more important, invalidating the hypothesis of predicting permeability without them.

The main conclusion of this work is that artificial intelligence trained with LBM simulation data is a powerful tool for predicting permeability values for rock geometries. It has direct field applications and, if appropriately trained, can calculate permeability faster than direct numerical simulation, maintaining accuracy while saving resources.

5.2 Limitations, suggestions and future works

It is essential to mention some intrinsic difficulties found during the development of this work, serving as a word of caution for others who try to pursue similar research.

First, a heavy mathematical background is necessary to understand kinetic theory and the lattice Boltzmann method. It may frighten readers not used to this kind of mathematical sophistication and microscopic physics abstraction, and even for those used to fluid mechanics, it may offer new challenges as kinetic theory is not generally taught in standard engineering courses. Specifically, the section regarding the discretization in velocity space through the Hermite Polynomials approach is hard to explain, understand, and visualize, leaving the velocity sets a mystery for the LBM users.

Second, implementing Zou-He boundary conditions is not trivial though it might seem direct, and it's not well established in the literature, with few works using it. The divergences between the standard boundary condition implementation and wet node approaches confuse those aiming to implement it. In this work, the Zou-He boundary conditions were implemented directly over the nodes, not using the wet-node approach, producing satisfactory results without extra care with interpolation schemes.

The geometries created were not necessarily perfect for direct simulation, leading to lots of time deployed in nonphysical simulations. As the geometries were created randomly, some were unsuitable for fluid flow simulations because they did not have percolating channels or a tiny channel radius (as 1px, for example). These simulations generally took a long computing time as they could not reach equilibrium, and some of them even crashed after hours of iterating as the values became too high for the computer to process. As the LBM simulations generally took at least one hour to run, the extensive database of generated geometries took weeks of computing time to run in a single computer using 10 CPU of its cores.

Further research topics can extend and improve this work to deal with 3D geometries, a choice made in chapter two when working with a bi-dimensional LBM scheme instead of a 3D approach due to the limited time and computational resources available. Also, training the Neural Network with 3D image data could significantly improve the range of applications of the network, as 3D porous media can exhibit phenomena that are not easily seen in 2D. This could make the model eager for rock tomography analysis, even substituting direct numerical simulation after correct training.

Also, this work is limited to the geometry sets used and the time necessary to perform the simulations and the training. Ideally, several different geometry models could be used to train a robust network that could predict the permeability of any given geometry without large error margins. Images of actual rock geometries could also have been used, making the model better for real-world applications. The time consumed by the simulations to predict the permeability of the geometries was also a problem, as this work did not use an efficient parallelization scheme based on GPUs, one of the main advantages of LBM. Instead, we remained limited to a single computer with 16 CPU cores. The training also consumed much time, even using less than 2.000 images and points to feed the model, showing that parallelization could also have been employed here to increase efficiency.

References

[1] RITCHIE, H., ROSER, M. “CO emissions”, *Our World in Data*, 2020. https://ourworldindata.org/co2-emissions.

[2] HÖHNE, N., GIDDEN, M. J., DEN ELZEN, M., et al. “Wave of net zero emission targets opens window to meeting the Paris Agreement”, *Nature Climate Change*, v. 11, n. 10, pp. 820–822, 2021.

[3] YAN, J., ZHANG, Z. “Carbon capture, utilization and storage (CCUS)”, *Applied Energy*, v. 235, pp. 1289–1299, 2019.

[4] FAHEY, T. J., WOODBURY, P. B., BATTLES, J. J., et al. “Forest carbon storage: ecology, management, and policy”, *Frontiers in Ecology and the Environment*, v. 8, n. 5, pp. 245–252, 2010.

[5] ONTL, T. A., SCHULTE, L. A. “Soil carbon storage”, *Nature Education Knowledge*, v. 3, n. 10, 2012.

[6] BICKLE, M. J. “Geological carbon storage”, *Nature Geoscience*, v. 2, n. 12, pp. 815–818, 2009.

[7] GISLASON, S. R., OELKERS, E. H. “Carbon storage in basalt”, *Science*, v. 344, n. 6182, pp. 373–374, 2014.

[8] ZHANG, D., SONG, J. “Mechanisms for geological carbon sequestration”, *Procedia IUTAm*, v. 10, pp. 319–327, 2014.

[9] NEWELL, P., ILGEN, A. G. “Overview of geological carbon storage (GCS)”. In: *Science of carbon storage in deep saline formations*, Elsevier, pp. 1–13, 2019.

[10] ADLER, P. *Porous media: geometry and transports.* Elsevier, 2013.

[11] DOTSON, B., SLOBOD, R., MCCREERY, P., et al. “Porosity-measurement comparisons by five laboratories”, *Journal of Petroleum Technology*, v. 3, n. 12, pp. 341–346, 1951.

[12] TERNERO, F., ROSA, L. G., URBAN, P., et al. "Influence of the total porosity on the properties of sintered materials—A review", *Metals*, v. 11, n. 5, pp. 730, 2021.

[13] ESPINAL, L. "Porosity and its measurement", *Characterization of Materials*, pp. 1–10, 2002.

[14] GHANBARIAN, B., HUNT, A. G., EWING, R. P., et al. "Tortuosity in porous media: a critical review", *Soil science society of America journal*, v. 77, n. 5, pp. 1461–1477, 2013.

[15] BLUNT, M. J. *Multiphase flow in permeable media: A pore-scale perspective.* Cambridge university press, 2017.

[16] SING, K. S. "Characterization of porous materials: past, present and future", *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, v. 241, n. 1-3, pp. 3–7, 2004.

[17] DARCY, H. *Les fontaines publiques de la ville de Dijon: exposition et application des principes à suivre et des formules à employer dans les questions de distribution d'eau*, v. 1. Victor Dalmont, 1856.

[18] ARORA, K. *Soil Mechanics and Foundation Engineering.* Standard Publishers, 1989.

[19] HUBBERT, M. K. "Darcy's law and the field equations of the flow of underground fluids", *Transactions of the AIME*, v. 207, n. 01, pp. 222–239, 1956.

[20] NEUMAN, S. P. "Theoretical derivation of Darcy's law", *Acta mechanica*, v. 25, n. 3-4, pp. 153–170, 1977.

[21] WOESSNER, W., POETER, E. *Hydrogeologic Properties of Earth Materials and Principles of Groundwater Flow.* 1 ed. Oxford, Oxford University Press, USA, 2018.

[22] CARMAN, P. C. "Flow of gases through porous media", *(No Title)*, 1956.

[23] MOSTAGHIMI, P., BLUNT, M., BIJELJIC, B. "Computations of Absolute Permeability on Micro-CT Images", *Mathematical Geosciences*, v. 45, pp. 103–125, 2013.

[24] DE SILVA, P., RANJITH, P. "Advanced core flooding apparatus to estimate permeability and storage dynamics of CO2 in large coal specimens", *Fuel*, v. 104, pp. 417–425, 2013.

[25] KRAUSE, M., PERRIN, J.-C., BENSON, S. “Recent progress in predicting permeability distributions for history matching core flooding experiments”, *Energy Procedia*, v. 4, pp. 4354–4361, 2011.

[26] KHOSHAW, F. M. *Petroleum and Mineral Resources (Wit Transactions on Engineering Sciences)*. 1 ed. Southampton, WIT Press, 2012.

[27] REZAE, R., SAEEDI, A., CLENNELL, B. “Tight gas sands permeability estimation from mercury injection capillary pressure and nuclear magnetic resonance data”, *Journal of Petroleum Science and Engineering*, v. 88-89, pp. 92–99, 2012.

[28] TIMUR, A. “Pulsed Nuclear Magnetic Resonance Studies of Porosity, Movable Fluid, and Permeability of Sandstones”, *Journal of Petroleum Technology*, v. 21, pp. 775–786, 1969.

[29] LALA, A., EL-SAYED, N. “Calculating absolute permeability using nuclear magnetic resonance models”, *Arabian Journal of Geosciences*, v. 8, pp. 7955–7960, 2015.

[30] JIA, B., JIN, L., SMITH, S., et al. “Extension of the Gas Research Institute (GRI) method to measure the permeability of tight rocks”, *Journal of Natural Gas Science and Engineering*, v. 91, pp. 103756, July 2021.

[31] SIAVASHI, J., NAJAFI, A., SHARIFI, M., et al. “An insight into core flooding experiment via NMR imaging and numerical simulation”, *Fuel*, v. 318, pp. 123589, 2022.

[32] PILLER, M., SCHENA, G., NOLICH, M., et al. “Analysis of Hydraulic Permeability in Porous Media: From High Resolution X-ray Tomography to Direct Numerical Simulation”, *Transport in porous media*, v. 80, pp. 57–78, 2009.

[33] SCHENA, G., FAVRETTO, S., PILLER, M., et al. “Pore Space Characterisation and Permeability Prediction Using Fast Network Extraction and Pore Throat Conductance Calculation”. Paper presented at the Europec/EAGE Conference and Exhibition, Rome, Italy, 2009.

[34] ISLAM, A., FAISAL, T., CHEVALIER, S., et al. “Multi-scale experimental and numerical simulation workflow of absolute permeability in heterogeneous carbonates”, *Journal of Petroleum Science and Engineering*, v. 173, pp. 326–338, 2019.

[35] BARR, D. “Coefficient of Permeability Determined By Measurable Parameters”, *Ground water*, v. 39, pp. 356–361, 2002.

[36] GU, Y., BAO, Z., CUI, G. “Permeability prediction using hybrid techniques of continuous restricted Boltzmann machine, particle swarm optimization and support vector regression”, *Journal of Natural Gas Science and Engineering*, v. 59, pp. 97–115, 2018.

[37] WU, J., YIN, X., XIAO, H. “Seeing Permeability From Images: Fast Prediction with Convolutional Neural Networks”, *Science Bulletin*, v. 63, pp. 1215–1222, 2018.

[38] LI, K., HORNE, R. N. “Numerical simulation without using experimental data of relative permeability”, *Journal of Petroleum Science and Engineering*, v. 61, n. 2-4, pp. 67–74, 2008.

[39] BEAR, J. *Dynamics of fluids in porous media.* Courier Corporation, 1972.

[40] WANG, Z., LI, W., BI, L., et al. “Estimation of the REV size and equivalent permeability coefficient of fractured rock masses with an emphasis on comparing the radial and unidirectional flow configurations”, *Rock Mechanics and Rock Engineering*, v. 51, pp. 1457–1471, 2018.

[41] NOBLE, R., ISLAM, S. Z., DROUBI, G., et al. “Fluid Flow through a Fractured Porous Reservoir Using CFD Modeling”, *Journal of Porous Media*, v. 22, n. 5, 2019.

[42] LAURE, P., PUAUX, G., SILVA, L., et al. “Permeability computation on a REV with an immersed finite element method”. In: *AIP Conference Proceedings*, v. 1353, pp. 978–983. American Institute of Physics, 2011.

[43] FEICHTNER, A., MACKAY, E., TABOR, G., et al. “Using a porous-media approach for CFD modelling of wave interaction with thin perforated structures”, *Journal of Ocean Engineering and Marine Energy*, v. 7, pp. 1–23, 2021.

[44] JOEKAR-NIASAR, V., HASSANIZADEH, S. “Analysis of fundamentals of two-phase flow in porous media using dynamic pore-network models: A review”, *Critical reviews in environmental science and technology*, v. 42, n. 18, pp. 1895–1976, 2012.

[45] YANG, Y., WANG, K., ZHANG, L., et al. “Pore-scale simulation of shale oil flow based on pore network model”, *Fuel*, v. 251, pp. 683–692, 2019.

[46] GOSTICK, J., AGHIGHI, M., HINEBAUGH, J., et al. "OpenPNM: a pore network modeling package", *Computing in Science & Engineering*, v. 18, n. 4, pp. 60–74, 2016.

[47] XIONG, Q., BAYCHEV, T. G., JIVKOV, A. P. "Review of pore network modelling of porous media: Experimental characterisations, network constructions and applications to reactive transport", *Journal of contaminant hydrology*, v. 192, pp. 101–117, 2016.

[48] AL-RAOUSH, R., PAPADOPOULOS, A. "Representative elementary volume analysis of porous media using X-ray computed tomography", *Powder Technology*, v. 200, pp. 69–77, 2010.

[49] TAHMASEBI, P., JAVADPOUR, F., SAHIMI, M. "Three-Dimensional Stochastic Characterization of Shale SEM Images", *Transport in porous media*, v. 110, pp. 521–531, 2015.

[50] KOHANPUR, A. H., CHEN, Y., VALOCCHI, A. J. "Using direct numerical simulation of pore-level events to improve pore-network models for prediction of residual trapping of CO2", *Frontiers in Water*, v. 3, pp. 710160, 2022.

[51] KOHANPUR, A. H., RAHROMOSTAQIM, M., VALOCCHI, A. J., et al. "Two-phase flow of CO2-brine in a heterogeneous sandstone: Characterization of the rock and comparison of the lattice-Boltzmann, pore-network, and direct numerical simulation methods", *Advances in Water Resources*, v. 135, pp. 103469, 2020.

[52] MCNAMARA, G., ZANETTI, G. "Use of the Boltzmann Equation to Simulate Lattice-Gas Automata", *Physical Review Letters*, v. 61, pp. 2332–2335, 1988.

[53] HE, Y., LIU, Q., TAO, W. "Lattice Boltzmann method for the heat conduction problem with phase change", *Numerical Heat Transfer, Part B*, v. 39, pp. 167–187, 2010.

[54] LI, L., LU, J., FANG, H., et al. "Lattice Boltzmann Method for Fluid-Thermal Systems: Status, Hotspots, Trends and Outlook", *IEEE Acess*, v. 8, pp. 27649–27675, 2020.

[55] LI, L., CHEN, C., MEI, R., et al. "Conjugate heat and mass transfer in the lattice Boltzmann equation method", *Physical Review*, v. 89, pp. 043308, 2014.

[56] JAHANSHALOO, L., SIDIK, N., FAZELI, A., et al. “An overview of boundary implementation in lattice Boltzmann method for computational heat and mass transfer”, *International Communications in Heat and Mass Transfer*, v. 78, pp. 1–12, 2016.

[57] CHEN, L., KANG, Q., MU, Y., et al. “A critical review of the pseudopotential multiphase lattice Boltzmann model: Methods and applications”, *International Journal of Heat and Mass Transfer*, v. 76, pp. 210–236, 2014.

[58] ZHANG, J., LI, B., KWOK, D. “Mean-field free-energy approach to the lattice Boltzmann method for liquid-vapor and solid-fluid interfaces”, *Physical review*, v. 69, pp. 032602, 2004.

[59] JOSHI, S., SUN, Y. “Multiphase lattice Boltzmann method for particle suspensions”, *Physical review*, v. 79, pp. 066703, 2009.

[60] KANG, Q., D., Z., CHEN, S. “Unified lattice Boltzmann method for flow in multiscale porous media”, *Physical Review*, v. 66, pp. 056307, 2002.

[61] SPAID, M., PHELAN, F. “Lattice Boltzmann methods for modeling microscale flow in fibrous porous media”, *Physics of fluids*, v. 9, pp. 2468–2474, 1997.

[62] MARTYS, N., CHEN, H. “Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice Boltzmann method”, *Physical review*, v. 53, pp. 743, 1996.

[63] WANG, S., CHEN, L., FENG, Q., et al. “Pore-scale simulation of gas displacement after water flooding using three-phase lattice Boltzmann method”, *Capilarity*, v. 6, pp. 19–30, 2023.

[64] HE, Y., LIU, Q., TAO, W. “Lattice Boltzmann methods for single-phase and solid-liquid phase-change heat transfer in porous media: A review”, *International Journal of Heat and Mass Transfer*, v. 129, pp. 160–197, 2019.

[65] XU, A., ZHAO, T., SHI, L. “Lattice Boltzmann Simulation of Mass Transfer Coefficients for Chemically Reactive Flows in Porous Media”, *Journal of Heat and Mass Transfer*, v. 140, pp. 052601, 2018.

[66] KANG, Q., D., Z., CHEN, S., et al. “Lattice Boltzmann simulation of chemical dissolution in porous media”, *Physical Review*, v. 65, pp. 036318, 2002.

[67] KRÜGER, T. *The Lattice Boltzmann Method: Principles and Practice*. 1 ed. New York, Springer, 2017.

[68] GUO, Z., SHU, C. *Lattice Boltzmann Method and its Applications in Engineering*. 1 ed. Singapore, World Scientific Publishing Co. Pte. Ltd, 2013.

[69] SUCCI, S., SUCCI, S. *The lattice Boltzmann equation: for complex states of flowing matter*. Oxford university press, 2018.

[70] MULLER, I. *A History of Thermodynamics: The Doctrine of Energy and Entropy*. 1 ed. New York, Springer, 2007.

[71] LOEB, L. *The kinetic theory of gases*. 3 ed. Mineola, New York, Dover publications inc, 2004.

[72] BHATNAGAR, P. L., GROSS, E. P., KROOK, M. "A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems", *Physical review*, v. 94, pp. 511–525, maio 1954.

[73] MAXWELL, J. C. "On the Motions and Collisions of Perfectly Elastic Spheres", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, v. 19, pp. 19–32, 1860.

[74] BOLTZMANN, L. "Weitere Studien über das Wärmegleichgewicht unter Gasmolekülen", *Sitzungsberichte der Akademie der Wissenschaften Wien (II)*, v. 66, pp. 275–370, 1872.

[75] HARDY, J., POMEAU, Y., PAZZIS, O. "Time Evolution of a Two-Dimensional Classical Lattice System", *Physical review*, v. 31, pp. 276–279, 1973.

[76] FRISCH, U., HASSLACHER, B., POMEAU, Y. "Lattice Gas Automata for the Navier-Stokes Equation", *Physical review*, v. 56, pp. 1505–1508, April 1986.

[77] HE, X., LUO, L. "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice boltzmann equation", *Physical review*, v. 56, pp. 6811–6817, December 1997.

[78] SHAN, X., YUAN, X., HUDONG, C. "Kinetic theory representation of hydrodynamics: a way beyond the Navier–Stokes equation", *Journal of Fluid Mechanics*, v. 550, pp. 413–441, 2006.

[79] ZIEGLER, D. "Boundary Conditions for Lattice Boltzmann simulations", *Journal of statistical physics*, v. 71, pp. 1171–1177, 1993.

[80] ZOU, Q., HE, X. "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model", *Physics of fluids*, v. 9, pp. 1591–1598, 1997.

[81] KHIREVICH, S., GINZBURG, I., TALLAREK, U. “Coarse- and fine-grid numerical behavior of MRT/TRT lattice-Boltzmann schemes in regular and random sphere packings”, *Journal of computational physics*, v. 281, pp. 708–742, 2014.

[82] LALLEMAND, P., LUO, ., TALLAREK, U. “Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability”, *Physical review*, v. 61, pp. 6546–6562, 2000.

[83] D’HUMIÈRES, D., GINZBERG, I. “Viscosity independent numerical errors for Lattice Boltzmann models: From recurrence equations to “magic” collision numbers”, *Computers Mathematics with Applications*, v. 58, pp. 823–840, 2009.

[84] HE, X., ZOU, Q. “Analytic Solutions of Simple Flows and Analysis of Nonslip Boundary Conditions for the Lattice Boltzmann BGK Model”, *Journal of statistichal physics*, v. 87, pp. 115–136, 1997.

[85] MANWART, C., AALTOSALMI, U., KOPONEN, A., et al. “Lattice-Boltzmann and finite-difference simulations for the permeability for three-dimensional porous media”, *Physical Review E*, v. 66, n. 1, pp. 016702, 2002.

[86] GUO, Z., ZHAO, T. “Lattice Boltzmann model for incompressible flows through porous media”, *Physical review E*, v. 66, n. 3, pp. 036304, 2002.

[87] NARVÁEZ, A., ZAUNER, T., RAISCHEL, F., et al. “Quantitative analysis of numerical estimates for the permeability of porous media from lattice-Boltzmann simulations”, *Journal of Statistical Mechanics: Theory and Experiment*, v. 2010, n. 11, pp. P11026, 2010.

[88] CANCELLIERE, A., CHANG, C., FOTI, E., et al. “The permeability of a random medium: comparison of simulation with theory”, *Physics of Fluids A: Fluid Dynamics*, v. 2, n. 12, pp. 2085–2088, 1990.

[89] ESHGHINEJADFARD, A., DARÓCZY, L., JANIGA, G., et al. “Calculation of the permeability in porous media using the lattice Boltzmann method”, *International Journal of Heat and Fluid Flow*, v. 62, pp. 93–103, 2016.

[90] MANDELBROT, B. B., MANDELBROT, B. B. *The fractal geometry of nature*, v. 1. WH freeman New York, 1982.

[91] POKOJSKI, W., POKOJSKA, P. “Voronoi diagrams–inventor, method, applications”, *Polish Cartographical Review*, v. 50, n. 3, pp. 141–150, 2018.

[92] HILL, R. "Voronoi Diagram Generator". 2024. Available in:<`https://cfbrasz.github.io/Voronoi.html`> Accessed in March 2nd, 2024.

[93] VORONOI, G. "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites." *Journal für die reine und angewandte Mathematik (Crelles Journal)*, v. 1908, n. 133, pp. 97–102, 1908.

[94] VORONOI, G. "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs." *Journal für die reine und angewandte Mathematik (Crelles Journal)*, v. 1908, n. 134, pp. 198–287, 1908.

[95] BOCK, M., TYAGI, A. K., KREFT, J.-U., et al. "Generalized voronoi tessellation as a model of two-dimensional cell tissue dynamics", *Bulletin of mathematical biology*, v. 72, pp. 1696–1731, 2010.

[96] WALTER, M., FOURNIER, A., MENEVAUX, D. "Integrating shape and pattern in mammalian models". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 317–326, 2001.

[97] LAFORCE, T., EBEIDA, M., JORDAN, S., et al. "Voronoi Meshing to Accurately Capture Geological Structure in Subsurface Simulations", *Mathematical Geosciences*, v. 55, n. 2, pp. 129–161, 2023.

[98] ENCALADA, Á., BARZOLA-MONTESES, J., ESPINOZA-ANDALUZ, M. "A Permeability–Throat Diameter Correlation for a Medium Generated with Delaunay Tessellation and Voronoi Algorithm", *Transport in Porous Media*, v. 132, n. 1, pp. 201–217, 2020.

[99] VRETTOS, N. A., IMAKOMA, H., OKAZAKI, M. "Characterization of porous media by means of the Voronoi-Delaunay tessellation", *Chemical Engineering and Processing: Process Intensification*, v. 25, n. 1, pp. 35–45, 1989.

[100] RAOOF, A., HASSANIZADEH, S. M. "A new method for generating pore-network models of porous media", *Transport in porous media*, v. 81, pp. 391–407, 2010.

[101] XIAO, F., YIN, X. "Geometry models of porous media based on Voronoi tessellations and their porosity–permeability relations", *Computers & Mathematics with Applications*, v. 72, n. 2, pp. 328–348, 2016.

[102] GOSTICK, J. T., KHAN, Z. A., TRANTER, T. G., et al. "PoreSpy: A python toolkit for quantitative analysis of porous media images", *Journal of Open Source Software*, v. 4, n. 37, pp. 1296, 2019.

[103] QIN, C.-Z., VAN BRUMMELEN, H., HEFNY, M., et al. "Image-based modeling of spontaneous imbibition in porous media by a dynamic pore network model", *Advances in Water Resources*, v. 152, pp. 103932, 2021.

[104] ESPINOZA-ANDALUZ, M., PAGALO, J., ÁVILA, J., et al. "An alternative methodology to compute the geometric tortuosity in 2D porous media using the A-Star pathfinding algorithm", *Computation*, v. 10, n. 4, pp. 59, 2022.

[105] AHMED-MALOUM, M., DAVID, T., GUETAZ, L., et al. "Computation of oxygen diffusion properties of the gas diffusion medium-microporous layer assembly from the combination of X-ray microtomography and focused ion beam three dimensional digital images", *Journal of Power Sources*, v. 561, pp. 232735, 2023.

[106] MCCARTHY, J., MINSKY, M. L., ROCHESTER, N., et al. "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955", *AI magazine*, v. 27, n. 4, pp. 12–12, 1956.

[107] SAMUEL, A. L. "Some studies in machine learning using the game of checkers", *IBM Journal of research and development*, v. 3, n. 3, pp. 210–229, 1959.

[108] MAHESH, B. "Machine learning algorithms-a review", *International Journal of Science and Research (IJSR).[Internet]*, v. 9, n. 1, pp. 381–386, 2020.

[109] GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* " O'Reilly Media, Inc.", 2022.

[110] KOZA, J. R., BENNETT, F. H., ANDRE, D., et al. "Automated design of both the topology and sizing of analog electrical circuits using genetic programming", *Artificial intelligence in design'96*, pp. 151–170, 1996.

[111] LECUN, Y., BENGIO, Y., HINTON, G. "Deep learning", *nature*, v. 521, n. 7553, pp. 436–444, 2015.

[112] GU, J., WANG, Z., KUEN, J., et al. "Recent advances in convolutional neural networks", *Pattern recognition*, v. 77, pp. 354–377, 2018.

[113] LECUN, Y., BOTTOU, L., BENGIO, Y., et al. "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, v. 86, n. 11, pp. 2278–2324, 1998.

[114] TAYE, M. M. "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions", *Computation*, v. 11, n. 3, pp. 52, 2023.

[115] SHAMSALDIN, A. S., FATTAH, P., RASHID, T. A., et al. "A study of the applications of convolutional neural networks", *J. Sci. Eng*, v. 3, pp. 31–39, 2019.

[116] ABDEL-HAMID, O., MOHAMED, A.-R., JIANG, H., et al. "Convolutional neural networks for speech recognition", *IEEE/ACM Transactions on audio, speech, and language processing*, v. 22, n. 10, pp. 1533–1545, 2014.

[117] ANWAR, S. M., MAJID, M., QAYYUM, A., et al. "Medical image analysis using convolutional neural networks: a review", *Journal of medical systems*, v. 42, pp. 1–13, 2018.

[118] BROWNE, M., GHIDARY, S. S. "Convolutional neural networks for image processing: an application in robot vision". In: *Australasian Joint Conference on Artificial Intelligence*, pp. 641–652. Springer, 2003.

[119] LOU, G., SHI, H. "Face image recognition based on convolutional neural network", *China communications*, v. 17, n. 2, pp. 117–124, 2020.

[120] ALQAHTANI, N., ARMSTRONG, R. T., MOSTAGHIMI, P. "Deep learning convolutional neural networks to predict porous media properties". In: *SPE Asia Pacific oil and gas conference and exhibition*, p. D012S032R010. SPE, 2018.

[121] KAMRAVA, S., IM, J., DE BARROS, F. P., et al. "Estimating dispersion coefficient in flow through heterogeneous porous media by a deep convolutional neural network", *Geophysical Research Letters*, v. 48, n. 18, pp. e2021GL094443, 2021.

[122] SANTOS, J. E., XU, D., JO, H., et al. "PoreFlow-Net: A 3D convolutional neural network to predict fluid flow through porous media", *Advances in Water Resources*, v. 138, pp. 103539, 2020.

[123] WANG, Y. D., CHUNG, T., ARMSTRONG, R. T., et al. "ML-LBM: predicting and accelerating steady state flow simulation in porous media with convolutional neural networks", *Transport in Porous Media*, v. 138, n. 1, pp. 49–75, 2021.

[124] YALAMANCHI, P., DATTA GUPTA, S. "Estimation of pore structure and permeability in tight carbonate reservoir based on machine learning (ML)

algorithm using SEM images of Jaisalmer sub-basin, India", *Scientific Reports*, v. 14, n. 1, pp. 930, 2024.

[125] CYBENKO, G. "Approximation by superpositions of a sigmoidal function", *Mathematics of control, signals and systems*, v. 2, n. 4, pp. 303–314, 1989.

[126] ABADI, M., BARHAM, P., CHEN, J., et al. "{TensorFlow}: a system for {Large-Scale} machine learning". In: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016.

[127] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., et al. "Scikit-learn: Machine learning in Python", *the Journal of machine Learning research*, v. 12, pp. 2825–2830, 2011.

[128] GULLI, A., PAL, S. *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[129] MOOLAYIL, J., MOOLAYIL, J., JOHN, S. *Learn Keras for deep neural networks*. Springer, 2019.

[130] PRAKASH, K. B., KANNAN, R., ALEXANDER, S. A., et al. *Advanced Deep Learning for Engineers and Scientists*. Springer, 2021.

[131] KIM, S., WIMMER, H., KIM, J. "Analysis of deep learning libraries: Keras, Pytorch, and MXnet". In: *2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 54–62. IEEE, 2022.

[132] KATHURIA, R. S., GAUTAM, S., SINGH, A., et al. "Real time sentiment analysis on twitter data using deep learning (Keras)". In: *2019 international conference on computing, communication, and intelligent systems (ICCCIS)*, pp. 69–73. IEEE, 2019.

[133] JOSHI, S., OWENS, J. A., SHAH, S., et al. "Analysis of preprocessing techniques, Keras tuner, and transfer learning on cloud street image data". In: *2021 IEEE International Conference on Big Data (Big Data)*, pp. 4165–4168. IEEE, 2021.

[134] GUILLERMO, M., FRANCISCO, K., CONCEPCION, R., et al. "A Comparative Study on Satellite Image Analysis for Road Traffic Detection using YOLOv3-SPP, Keras RetinaNet and Full Convolutional Network". In: *2023 8th International Conference on Business and Industrial Research (ICBIR)*, pp. 578–584. IEEE, 2023.