Machine Learning and CFD: a new approach to simulate and optimize micromixers in different geometries

Luca Martin Ainstein

Advisors: D.Sc Fábio Pereira dos Santos and
D.Sc. Daniela de O. Maionchi

Rio de Janeiro
October 2022

Machine Learning and CFD: a new approach to simulate and optimize micromixers in different geometries

Luca Martin Ainstein

A bachelor thesis submitted to the Chemical Engineering department of the School of Chemistry at the Federal University of Rio de Janeiro in fulfillment of the requirements for the degree of Chemical Engineer

Author:

Luca Martin Ainstein

Advisor professor I:

Prof. D.Sc Fábio Pereira dos Santos

Advisor II:

Prof. D.Sc Daniela de O. Maionchi

Thesis Defence Committee:

Prof. PhD Tânia Suaiden Klein

Thesis Defence Committee:

Prof. D.Sc Gabriel Gonçalves da Silva Ferreira

Rio de Janeiro

August 2021

CIP - Catalogação na Publicação

A296m

Ainstein, Luca Martin
Machine Learning and CFD: a new approach to simulate and optimize micromixers in different geometries / Luca Martin Ainstein. -- Rio de Janeiro, 2023.
59 f.

Orientador: Fábio Pereira dos Santos.
Coorientador: Daniela de Oliveira Maionchi.
Trabalho de conclusão de curso (graduação) - Universidade Federal do Rio de Janeiro, Escola de Química, Bacharel em Engenharia Química, 2023.

1. Machine learning. 2. Neural Networks. 3. Optimization. I. dos Santos, Fábio Pereira, orient. II. Maionchi, Daniela de Oliveira, coorient. III. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/6283.

Esse trabalho é dedicado à minha amiga da alma Chiara Prochnik.

"All the stars must represent some single gigantic equation, to the mind of god as straight-forward as, say, the equation of a sphere... To us: unreadable, incalculable."

- Thomas Pynchon

ACKNOWLEDGEMENTS

Impossível não começar agradecendo aos meus pais, Monica Krekcza e Alejandro Ainstein, por todo o suporte, confiança, apoio e amor incondicional não somente durante a graduação, mas durante toda uma vida, bem como aos meus irmãos, Galo e Valentina Ainstein, que tiveram paciência e também sempre me apoiaram.

Não cabe nesse espaço o quanto sou grato pela minha companheira de vida, Ana Luísa Martins, que me deu o amor, carinho além de acolhimento nos últimos sete anos, sem os quais não seria hoje em dia quem eu sou. Obrigado por ter paciência quando precisei, e por sempre se preocupar comigo, te amo muito! Agradeço também aos meus sogros, Bernardete Ferreira e Alteclínio Martins, que me acolheram nesses anos, me daram apoio e ajuda em todos os momentos que precisei.

Sem o apoio de amigos também não seria possível realizar essa caminhada, e gostaria de destacar alguns. Os meus irmãos da vida: Antonia Vega, Malu Marques e João Lucas, que estão comigo há quase 20 anos, dividindo uma vida de histórias, memórias, planos e ideias. Malu Aguiar, amiga que fiz na faculdade e que levarei para o resto da vida, obrigado por me ouvir, opinar e me mostrar o caminho nos momentos em que eu sozinho não enxergava. Bruna Lins, por sempre ter uma porta da casa aberta pra mim, ensolarar meus dias além sempre me apoiar. Por fim, agradeço ao meu amigo João Carlos Martins, que, há 10 anos me apresenta um olhar original e inovador sobre como encarar a vida e tornou esses anos no Rio muito mais felizes e agradáveis.

Agradeço também a todos os professores que me acompanharam durante a gradução, em especial ao meu orientador Professor Dr. Fábio dos Santos por me indicar o caminho para a realização desse trabalho, tendo sempre paciência, compreensão e uma didática sem a qual a conclusão deste trabalho não seria possível. Agradeço também à Professora Dra. Daniela Maionchi, coorientadora deste trabalho, pelas dicas, orientações e correções durante o processo.

Obrigado UFRJ, por cada momento que passei aqui. O poder de transformação que a Universidade Pública possui é inquestionável e algo único, é uma pena que ainda seja uma experiência tão restringida.

A América Latina é, como relembra a famosa música do grupo porto riquenho

Calle 13, *un Pueblo sin piernas pero que camina.* Nesse aspecto, seria injusto para mim, que cheguei ao Brasil em 2002, não agradecer em todas as minhas conquistas a este país que, apesar de todas as suas dificuldades, sempre tem um braço estendido para todos os que precisam. *Gracias* Brasil por me acolher e sempre me tratar como se daqui fosse.

Resumo da Monografia apresentada à Escola de Química como parte dos requisitos necessários para obtenção do grau de Bacharel em Engenharia Química

Machine Learning and CFD: a new approach to simulate and optimize micromixers in different geometries

Luca Martin Ainstein

October, 2022

Orientadores: Prof. Fábio Pereira dos Santos, D.Sc

Prof. Daniela de O. Maionchi, D.Sc.

Este trabalho busca explorar uma nova abordagem de otimização na área de microfluidodinâmica, utilizando a combinação de técnicas de Dinâmica de Fluidos Computacional (CFD) e Inteligência Artificial. O objetivo desta combinação é possibilitar a realização de uma otimização global com menor custo computacional. Essa otimização ocorre através da possibilidade de construir um banco de dados de simulações em um menor tempo, utilizando uma rede neural densa treinada com os dados obtidos utilizando CFD. Neste trabalho foi possível testar, para micromisturadores de geometria-Y, 265 simulações, variando parâmetros como diâmetro da obstrução (OD) e offset da obstrução (OF), permitindo obter uma rede neural que apresenta erros de cálculo menores do que 1% para o processo de escoamento e menores do que 4% para a perda de carga. Por fim, a utilização do algoritmo genético possibilitou, levando 3 parâmetros como base (custo da energia da mistura, perda de carga e porcentagem de mistura), otimizar os valores de OD e OF da geometria globalmente.

Palavras-chave: 1. CFD. 2. Micromixers. 3. Artificial Inteligence. 4. Geometry Optimization.

ABSTRACT

Machine Learning and CFD: a new approach to simulate and optimize micromixers in different geometries

Luca Martin Ainstein

Agosto, 2021

Supervisors: Prof. Fábio Pereira dos Santos, D.Sc

Prof. Daniela de O. Maionchi, D.Sc.

This work seeks to explore a new optimization approach in microfluid dynamics, using the combination of Computational Fluid Dynamics (CFD) and Artificial Intelligence techniques. The objective of this combination is to enable the realization of a global optimization with a lower computational cost. This optimization occurs with the possibility of building a database of simulations in a shorter time, using a dense neural network trained with data obtained using CFD. In this work, we tested 265 Y-geometry micromixers, varying parameters such as obstruction diameter (OD) and obstruction offset (OF). Allowing to obtain a neural network that presents calculation errors smaller than 1% for the flow process and less than 4% for pressure drop. Finally, the use of the genetic algorithm made it possible to consider 3 parameters as a basis (cost of mixing energy, head loss and mixing percentage), to optimize the OD and OF values of the geometry globally.

Key-words: 1. CFD. 2. Micromixers. 3. Artificial Inteligence. 4. Geometry Optimization.

Contents

Nomenclature

$\mathbf{u}$	Flow velocity
μ	Viscosity
ρ	Density
φ	Blend Percentage
p	Pressure
ABC	Artificial bee colony
API	Application programming interface
CFD	Computational fluid dynamics
DE	Differential evolution
DNN	Deep neural networks
GA	Genetic algorithm
LNP	Lipid nanoparticles
LOC	Lab-on-chip
mec	Mixing energy cost
OD	Obstruction diameter
OF	Obstruction offset
PSO	Particle swarm optimization
Re	Reynolds number

RMS Response surface model

SA Simalated annealing

TS Tabu search

List of Figures

List of Tables

Chapter 1

Introduction

The need to develop chemical engineering technologies to solve various of problems have led to the miniaturization of equipment (Capretto et al., 2011). In addition to enabling the integration of several functionalities on a single chip, called Lab-on-chip (LOC), it was possible to focus on reducing energy consumption, equipment/chemical plant sizes, production capacity ratio, and waste generation (Stankiewicz et al., 2000). The use of LOC has been reported in several areas, such as nanoparticle crystallization (Ståhl et al., 2001), extraction (Sprogies et al., 2008), polymerization (Nagaki et al., 2004), organic synthesis (Haswell et al., 2001), enzyme assay (Miller and Wheeler, 2008), protein folding (Bilsel et al., 2005), bioprocess optimization (Micheletti and Lye, 2006), and drug delivery studies (Razzacki et al., 2004).

The use of Computational Fluid Dynamics (CFD), an area that is concerned with combining physical knowledge about fluid mechanics with mathematical and computational resources and tools to predict, model, and optimize flow parameters, occupies a prominent role in several areas due to its versatility (Khan et al., 2018). Among these areas, it is possible to mention its use in the study of atmospheric movements (Mirzaei, 2021) and (Kim et al., 2021), biomedicine (Wüstenhagen et al., 2021), aerospace (Wang et al., 2013), oil and gas (Martínez et al., 2020), design of industrial equipment (Negi and Subhash, 2021), ocean engineering (Foroushani and Sabzpooshani, 2021) and microfluidics (Chen et al., 2020; Ortega-Casanova, 2017b).

The understanding of relevant physical parameters in the analysis of micromixers occurs through the use of CFD, since it is not possible to treat a mi-

cromixer as an ordinary reduced-size mixer, given that this change in dimensions totally changes the physics involved, because it creates a flow preferably laminar (Capretto et al., 2011). This paradigm shift causes the preferentially turbulent mixing model on regular mixers to give way to a molecular diffusion model (Beebe et al., 2002). The challenge that arises with this physical model is that, to improve the mixing process, caps (Wang et al., 2012) or obstructions (Alam et al., 2014) are added and geometries in T (Nimafar et al., 2012a; Zhendong et al., 2012), H (Nimafar et al., 2012a), O (Nimafar et al., 2012b), Y (Wang et al., 2012) or fractals (Chen et al., 2020) are adopted, which end up increasing the pressure drop, which directly influences the mixing energy cost (mec), a variable that will be revisited throughout the text, as it presents great value for analysis (Rahmannezhad and Mirbozorgi, 2019).

In this sense, the optimization of equipment involves the determination of geometry, flow rates, and pressure drop analysis (Wang et al., 2012). The difficulty found in the optimization process is the generation of data since, if an experimental procedure is used, it is necessary to build several geometries, and test in different configurations, which generates a high cost of experimentation, while a Computational calculation approach, using CFD, requires time to generate a relevant number of simulations, in addition to having a computational cost. This makes the optimization work, both experimental and simulation, have a limited number of cases being tested (Wang et al., 2012; Nimafar et al., 2012a; Rahmannezhad and Mirbozorgi, 2019). To avoid this difficulty, the use of Machine Learning emerges as a smart output, given that it is a type of technology that learns from the data generated, and allows predictions to be made in a time window much shorter than that required by existing CFD techniques.

Although several works combine CFD with machine learning, an extensive review has shown that few articles use machine learning in microfluidics. Of these few works, it is worth mentioning the work of Hadikhani et al. (2019), where many droplet images are recorded and used to train deep neural networks (DNN) to predict the flow rate or the concentration. This method quantifies the concentrations of each component with a 0.5% accuracy and the flow rate with a resolution of 0.05 ml/h. Arjun et al. (2020) detected and classified binary-coalesced droplets inside

microchannels based on the degree of mixing using a deep neural network. The use of Machine Learning to reduce order in optimization problems using CFD in micromixers is an unprecedented approach.

1.1 Motivation

One of the main challenges concerning the optimization of geometries using CFD is the high computational cost and processing time for performing the simulations. Because of this, this work's motivation was to think of a way to reduce this simulation time without significantly sacrificing the accuracy of the solution obtained.

This work proposes the use of a set of numerical flow simulations in different geometries for the training of a neural network. This neural network could perform the second set of simulations in a shorter time interval than the numerical solution, allowing the global optimization of geometry based on the parameters of interest.

In addition to the apparent relevance of this type of optimization, given that there is a reduction in processing time without an equivalent loss of assertiveness, this work paves the way for the use of the same possibility of resolution (the almost unprecedented combination of CFD and deep neural networks) in other fluid dynamics problems.

1.2 Objectives

1.2.1 General Objectives

Originally, one of the goals of this work was to perform simulations using CFD to allow global optimization of the geometry of this micromixer. However, this set of simulations required about 45 days of full computing power and was not sufficient to perform such an optimization process.

Because of that, the general objective of this work is to propose a new methodology for flow simulation in micromixers and geometry optimization through the combined use of CFD and machine learning.

1.2.2 Specific Objectives

The specific objectives of this study are as follows:

- Reproduce, through the use of numerical methods, the experimental results presented by Wang et al. (2012) for a simple micromixer geometry.
- Reproduce, through the use of numerical methods, the experimental results presented by Rahmannezhad and Mirbozorgi (2019) for a grooved micromixer geometry.
- Simulate, using OpenFOAM, a new set of geometries derived from data presented by Rahmannezhad and Mirbozorgi (2019).
- Use the results obtained in the simulations to train a neural network, which can calculate the flow parameters (pressure drop, mec and percentage of mixing) by selecting the geometry (type of obstruction, obstruction diameter (OD) and obstruction offset (OF)).
- Use the dataset generated by a Neural Network to reach the global optimization, using a Genetic Algorithm, of the system geometry.

1.3 Organization of this Work

An introduction to the main concepts used, as well as the application of micromixers in various fields of science is presented in Chapter 1, along with the general and specific objectives of this work. Chapter 2 presents and discusses the main technical concepts covered in this work, such as microfluidics, micromixers, the Navier-Stokes equation, numerical solution methods, and use of neural networks. In Chapter 3 it is possible to follow the entire methodology, starting with the construction of geometries, their simulations using OpenFOAM, training the neural network and geometry optimization. The results obtained are presented and discussed in Chapter 4. Finally, in Chapter 5 the final discussion is made, where some points are briefly presented and the conclusion is presented.

Chapter 2

Fundamental Concepts

Chapter 2 is focused on explaining the theoretical basis that guided the construction of this work. It reading begins with the definition and literature review related to microfluidics, passing through the main points of study of micromixers. Next, the Navier-Stokes equations, which model the flow of fluids, are discussed. In addition to explaining the functioning and application of such equations, this section aims to illustrate the historical context of its proposal and the difficulties and implications of the process of obtaining an analytical solution. This step is only for the didactic purpose of demonstrating to readers who are beginning the study of fluid mechanics the beauty behind an equation of such a simple nature. Finally, a review of numerical solution methods is made, deepening the concept of CFD. The following discussion is where the main concepts of Machine Learning are presented.

2.1 Computational Fluid Dynamics

The analytical solution of fluid dynamics problems is possible only in systems with simple geometries and specific pressure and flow velocity conditions (Fox et al., 2020).

There is a need for a solution model that can address more complex problems. The strategy adopted to solve this problem is to overcome the difficulty of working in complex geometries with different conditions by adopting a discretized mesh. That allow, through extensive calculation work, to obtain a convergent numerical solution (Fox et al., 2020).

The main problem when adopting this approach is that as the number of nodes increases, the computational cost increases. For very fine meshes, it is necessary to use supercomputers. The field of study that deals with this type of solution is known as Computational Fluid Dynamics (CFD).

The use of CFD can be analyzed as a technique composed of three sequential steps: pre-processing, solver, and post-processing (Versteeg and Malalasekera, 2007), as presented in Figure 2.1.

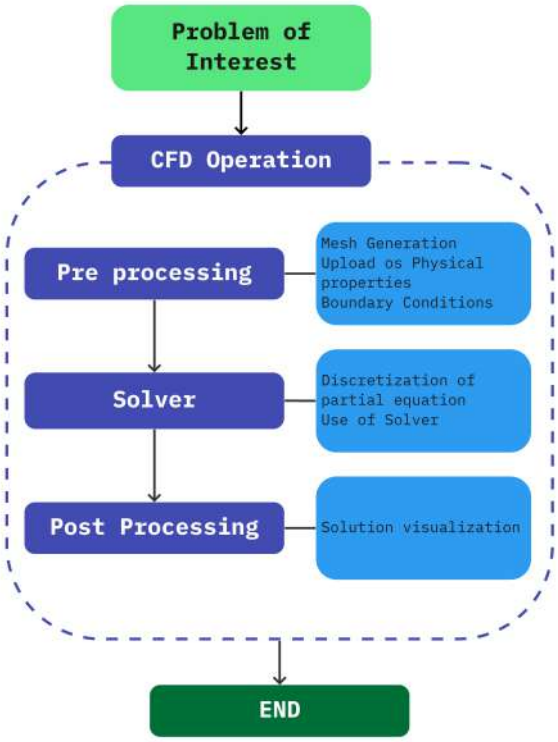


Figure 2.1: Schematic flowchart of steps to solve a problem using CFD inspired by Anderson and Wendt (1995).

In the pre-processing stage, the mesh that will be used is defined, in addition to describing the physical-chemical properties of the fluids represented in the problem of interest, as well as all relevant phenomenological models and their respective boundary conditions (Versteeg and Malalasekera, 2007).

The process of dividing the analysis domain into cells is called meshing or discretization (Silva and Cardoso, 2020). There is an extensive debate about the ideal size of each cell in a mesh, and this definition is made to reach the mesh that achieves the mesh independent solution. At the same time, one should seek

to achieve the mesh that minimizes the computational solution time required. The procedure usually used is shown in Figure 2.2 and consists of starting the process with a coarse mesh, solving the problem, refining the mesh (usually doubling the number of cells), and redoing the solution. Once you have both solutions, evaluate the convergence criterion and decide if another iteration is necessary (Silva and Cardoso, 2020).

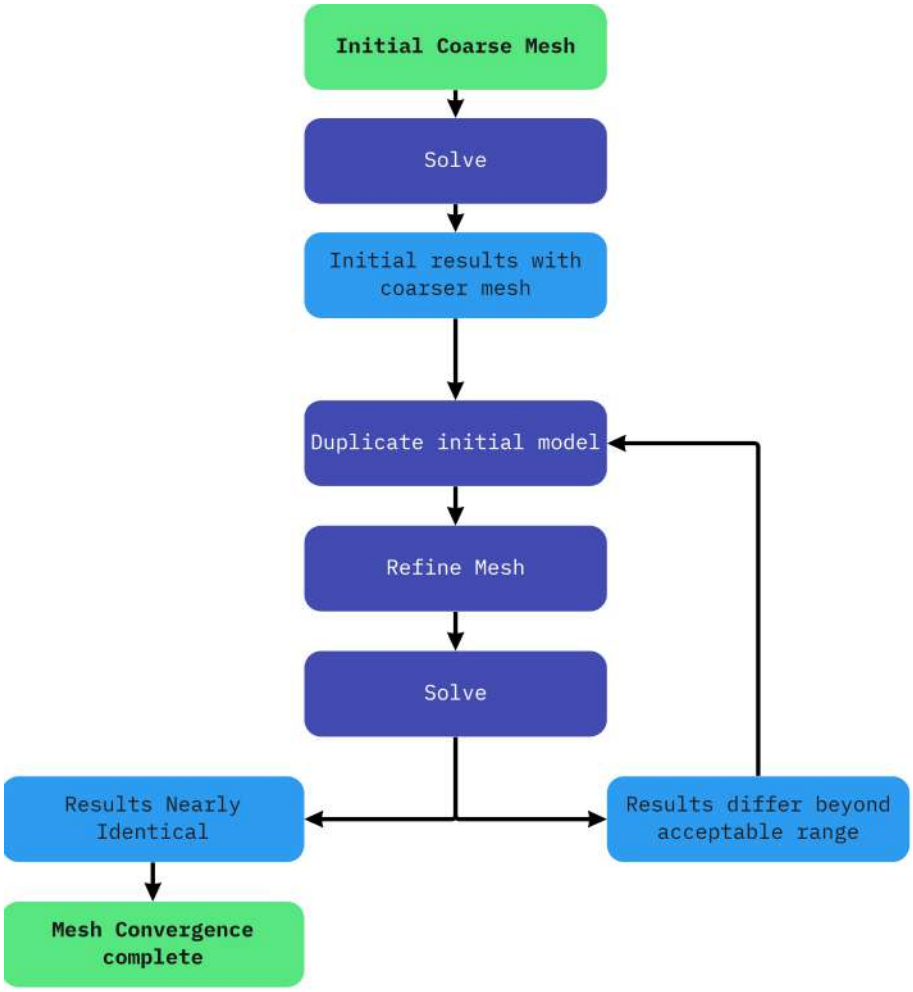


Figure 2.2: Iterative process of mesh selecting flowchart inspired by Silva and Cardoso (2020).

Silva and Cardoso (2020) studies the solution mesh construction process for a fluidized bed reactor. There is the possibility of choosing different cell sizes for the same geometry, obtained and optimized during the mesh convergence process.

In the solver step, the equations that govern the problem are discretized. Discretization involves transforming partial equations into a system of non-linear equations, which are solved using a solver. There is a deeper understanding of this

solution method in the section 2.3 (Versteeg and Malalasekera, 2007).

Finally, the post-processing step refers to a visualization step (which can be in 2D or 3D) of the generated data, the flow vectors, and other relevant generated results (Versteeg and Malalasekera, 2007).

In this section, the concept of numerical solution was constantly adopted. This concept was not developed in this section as it is discussed in section 2.3. Therefore, once the mesh convergence is defined, it is possible to proceed to the most appropriate numerical solution method.

It is also worth mentioning that the use of CFD comprises several areas, such as its application in the food industry (Norton and Sun, 2007; Xia and Sun, 2002), in oil and gas (Jafari et al., 2020), in the biochemical industry, as in the study of optimization of the process of pyrolysis (Rostami et al., 2004), in medicine (Morris et al., 2016), in the agro-enviromental industry (Lee et al., 2013), in the aeronautics (Langtry and Menter, 2005) and automotive (Dhaubhadel, 1996) industries, over heat exchangers and other equipaments design (Bhutta et al., 2012), among others applications.

2.2 Microfluidics

Microfluidics refers to the behavior, control or operation of fluids on the sub-millimeter scale, which causes surface forces to become the main forces of interaction between fluids and the medium. In fact, the understanding of relevant physical parameters in the analysis of micromixers occurs through the use of CFD, since it is not possible to treat a micromixer as an ordinary reduced-size mixer, given that this change in dimensions totally changes the physics involved, because it creates a flow preferably laminar (Ducree et al., 2006; Wu and Nguyen, 2005). When referring to equipment that involves the flow of more than one fluid, as is the case in this work, it is observed that, since it is not a turbulent flow, the mixing between the fluids shifts from convection to a diffusion mechanism (Pethig and Smith, 2012), which requires a long channel length and a high retention time, resulting in high pressure drop and energy cost (Alam et al., 2014).

The analysis of size reduction can be performed through the use of scaling law,

which compares surface forces (such as surface tension and viscosity) with volumetric forces (such as gravity and inertia) when there is a variation in dimension ℓ, keeping other physical properties constant, such as time, pressure and temperature (Bruus, 2007). The most common way to make this comparison is:

$$\frac{surface}{volume} = \frac{\ell^2}{\ell^3} = \ell^{-1} \xrightarrow[\ell \to 0]{} \infty \tag{2.1}$$

This reinforces the above by demonstrating that when working on the micro scale, volumetric forces, which play an important role in most fluid mechanics applications, lose out to surface forces.

Preetam et al. (2022) studied the main applications of microfluidics for use in the fields of medicine and biotechnology. He lists the main reasons that make the application of microfluidics appropriate: low consumption of reagents (for geometric reasons, it is trivial to understand that in fact there is a low consumption of reagents, which is especially useful in handling expensive materials, as synthesis reagents or enzymes, difficult to purify, in addition to making operations involving hazardous materials safer), high surface to volume ratio (facilitating mass and heat transfer, and being particularly useful in applications involving surface reactions), high Spatio-temporal resolution (microfluidics allow to maintain small environments and do detailed monitoring), high-throughput applications (microfluidics is ideal for applications such as screening studies as they have small working volumes and parallel reactions sites), rapid prototyping (there are prototyping methods that allow, at low cost, to quickly develop new geometries and equipment) and portable system (this relates to the initial purpose of microfluidics: to build a LOC).

In this sense, Maeki et al. (2022) presents technologies and devices for lipid nanoparticle-based RNA delivery. The use of new RNA-based compounds gained ground in 2021, with the approval of vaccines against COVID-19. The study of RNA-delivery technology has been investigated by several research centers for approximately 30 years. One of the technologies that have recently emerged is the use of Lipid Nanoparticles (LNP) to stabilize and improve the delivery of this RNA to target organs. The production of RNA-loaded LNP production was developed using advantages provided by microfluidics, such as high reproducibility, high-throughput optimization of LNP formulation, and controllability.

Cheng et al. (2022) present the main advances in using microfluidics to de-

velop advanced polymer films. The main point here is that the use of laminar flow properties, which occurs at the micro-scale, as discussed earlier, leads to the possibility of controlling and manipulating the properties of the developed film (such as ultrathin, heterogeneous, gradient, and/or multilayer). These films have applications in several emerging fields, such as biotechnology, nanotechnology, other syntheses of high added-value components, among others. Despite this, there is still a big gap between the means to develop new films and the market's need for them.

2.2.1 Micromixers

Over the past few decades, microfluidics devices have occupied an important role in chemical, biochemical, and analytical applications. As cited before, there are examples of microfluidics being applied in COVID detection, chemical synthesis, or other operations that require high efficiency and precise control. Therefore, for all the applications that mix two or more fluid streams, the micromixer is one of the essential units (Tofteberg et al., 2010).

Active micromixers require an external power source to increase their mixing capability. Because of this, they have a higher cost of operation and installation as they need auxiliary equipment to supply energy, which demands investment and space (Ortega-Casanova, 2016).

Ortega-Casanova (2016) proposes using an active micromixer consisting of a straight channel with a squared obstruction, that requires energy to oscillate in a given frequency and amplitude. This oscillation promotes an increase in the system's surface area, favoring the molecular diffusion mechanism. The work is based on finding the best flow configuration for each selected amplitude, including for low Reynolds numbers.

Other active micromixers construction strategies are interesting and worth mentioning. Groisman and Steinberg (2001) used polymer additives at a concentration of only 0.001% to lead to instability and irregularity in the flow, which increased the mixing capacity in curved channels. Hellman et al. (2007) used nanosecond pulses from laser to generate cavitation bubbles within 100 and 200 micrometers. This bubbles expansion and later collapsed produced a local region of mixed fluid, improving the percentage of mixing. Finally, Jain and Nandakumar (2013) used

heterogeneous electrical charges to create an electromagnetic field, thus obtaining a pattern of electrical charges that led to improvements in the results obtained in T-shaped micromixers.

On the other hand, there are passive micromixers, which use geometry changes to promote fluid mixing. These can be further divided into two subclasses: laminar, where the mixture is based on increasing diffusion transport to the molecular level, and convective, where chaotic advection in the flow occurs (Hardt et al., 2004).

In this work, Wang et al. (2012) uses a Y-shaped micromixer to propose 3 new geometries made with cylindrical grooves adjoining the main straight channel. The variation between geometries in the depth of the cylindrical groove used. Next, he studies both the experimental and simulated results, which is possible since the construction of the micromixer was carried out. The micromixer construction is a cheap and fast process, one of the properties that helps testing new geometries. With this, it is possible to conclude which is the best geometry to promote the highest percentage of mixing. In Section 4.1.1 these results are discussed in-depth and compared with those obtained in this work for the same geometry.

Mixing fluids under laminar conditions is a challenge for almost any application. Usually, this type of problem requires an increase in the size of the channel to increase the surface area, where the process of molecular diffusion, dominant in flows of this nature, occurs. In an attempt to increase the surface area without having to increase the length of the device, many works choose to insert obstacles in the flow path, creating a disturbance that, despite creating an increase in pressure drop, is capable of increasing the percentage of mixing (Rahmannezhad and Mirbozorgi, 2019; Ortega-Casanova, 2017a; Alam et al., 2014).

Alam et al. (2014) proposed a geometry for the construction of a micromixer, with circular obstructions along the flow. The results were compared with a micromixer with T-geometry, with the same pattern of obstructions. All results are from numerical simulations with Reynolds Numbers ranging from 0.1 to 60. Various parameters were evaluated: how the position and amount of obstructions vary percentage of mixing of the system, how did a micromixer behave without any obstruction, and how the variation of Reynolds Number generate different results. In general terms, the author's conclusion is that percentage of mixing increased along

with the increase in the number of obstructions, as well as with the increase in their heights. However, with Reynolds numbers approaching 60, these effects attenuated.

Rahmannezhad and Mirbozorgi (2019) discuss a local optimization using the response surface model (RMS) on grooved micromixers with obstructions. All geometries used for the optimization have the same type of grooved wall, varying between 3 different types of obstructions (square, circle, and diamond) and simulating, for each type of obstruction, different diameters, and positions. Through the evaluation of 3 parameters (pressure drop, percentage of mixing, and mixing energy cost), two of which are independent, it was possible to find the best configuration for each obstruction geometry. This article, and that of Wang et al. (2012) are of vital interest for this work as they will be reviewed and taken as a starting point for the construction of the geometries used. In Chapter 4 there is a space to discuss the results obtained in more depth.

On the other hand, in the field of passive convective micromixers, Lin (2015) proposes a numerical characterization of simple three-dimensional chaotic micromixers. Using two representative types, 3D serpentine micromixer and the square-wave micromixer with square grooves, it was possible to investigate the flow and mixing characteristics in a wide range of Reynolds Numbers: from 8 to 160. The evaluation was focused on mixing quality and pressure drop. At low Reynolds numbers, the mixing process is mainly dominated by the diffusion mechanism, while in higher Reynolds Numbers the mixing is enhanced by the gradually intensive chaotic advection.

Tayeb et al. (2020) studied the thermal mixing performance for 3 different geometries using a non-Newtonian fluid inside a two-layer crossing channels micromixer. In this case, there was not only concern about mixing in a scenario with laminar flow characteristics, but also about promoting thermal efficiency. The results showed that using this type of geometry deals well with both problems, especially the C-geometry, adopted by the authors as the optimal geometry, among those tested.

2.3 Solution Methods

Once the problem is defined, there are three main ways to proceed with the solution in order to understand all the parameters related to the flow. In this section, experimental, analytical, and numerical solutions will be described, not only to solve flow behavior in micromixers problems but to solve flow problems in general.

There is no intention in this section to define a solution method as the best method but to illustrate that each solution form is the one that best applies to a set of problems. Proof of this is that to have a holistic understanding of the problem, combining more than one method is often necessary, as some of the articles studied here show.

Solutions using experimental methods are the basis of all scientific knowledge, which makes this solution method the oldest among the three presented. However, one cannot have the mistaken impression that being older, is of lesser value Andersen and Hepburn (2015).

It is exactly the experimental results that led to the possibility of building models that explain the observed phenomenon, allowing the development of equations that can later be solved by analytical and numerical methods.

In the case of fluid mechanics, the experimental solution is normally used to validate the numerical method, that is, the researcher repeats the same experiment both experimentally and numerically and compares the results. If there is assertiveness in the numerical model, this allows the model to be extended to other cases with similar properties Xia et al. (2016).

This approach occurs because the main weakness of experimental methods is their difficulty in being scaled, and reproduced, and, in cases of high complexity, it often becomes impractical to perform. In addition, experimental methods require financial investment and a work structure not always available in research centers. In addition, one of the main drawbacks of working with experimental methods lies in the fact that the solution is subject to a range of possible errors such as equipment calibration, human operation of the equipment, solvent purity, parameter control (flow rate, pressure, temperature, etc.) in the equipment used, among others. Therefore, experimental tests end up not being representative of complex

problems.

Examples of experimental solutions involving micromixers include the work done by Wang et al. (2012), as discussed earlier, Chen et al. (2020), who studies passive micromixers with fractal-like tree structures, Sun et al., who investigates a magnetic micromixer under microwires and uniform magnetic field, Ahmadi et al. (2021), which analyzes the effects of baffle configuration and number on inertial mixing in a curved serpentine micromixer and Xia et al. (2016) which analyzes a planar micromixer with gaps and baffles based on field synergy principle. All the articles cited in this section end up solving the problem both experimentally and numerically and performing the comparison.

Starting from equations obtained from observations of the phenomena, it is tempting to solve such equations to obtain an exact and immediate solution to the problem. For some simple physical problems, this is possible, however, many mathematical models that seek to describe reality do not have an analytical solution. In this section, we will talk about those that actually have, given their simplicity, or based on a series of assumptions and considerations, exact analytical solutions.

In the case of fluid mechanics, there are the Navier-Stokes equations for a flow, which explain the observed phenomenon. However, the mathematical nature of these equations does not allow, with currently available techniques, to develop a comprehensive analytical solution to any problem with complex geometry.

One of the main problems in the analytical solution of the Navier-Stokes equations lies in our inability to understand the turbulence process. There is a famous story in the field of physics attributed to the German physicist Werner Heisenberg where he, on his deathbed, would have said (Powers, 1993):

> *When I meet God, I am going to ask him two questions: Why relativity? And why turbulence? I really believe he will have an answer for the first.*

Despite being just a story, there is an alive and complex scientific field of turbulence process.

Works that analytically solve the Navies-Stokes equations, presented below in Equations 2.2 and 2.3, must carry out a series of considerations (such as laminar flow, steady-state, among others), which restricts their solutions. Unfortunately, the

parameters that this work will use don't allow the analytical approach from being used.

$$\nabla.\mathbf{u} = 0. \tag{2.2}$$

$$\rho\frac{D\mathbf{u}}{Dt} = -\nabla p + \mu\nabla^2\mathbf{u}. \tag{2.3}$$

Where ρ is the density, $\mathbf{u}$ is the flow velocity, p is the pressure and μ is the viscosity. It is possible to obtain analytical solutions to the Navier-Stokes equations in some very particular cases. However, in micromixer applications, these solutions are not possible for the given model, boundary conditions and initial state, which leads to the need to approach the problem either by experimental or numerical approach.

Once it is defined that for a complex model such as a micromixer, it is impossible to solve the Navier Stokes Equations analytically, it is clear that there are two options: an experimental model (already discussed) or a numerical model.

Numerical methods are defined as algorithms that allow, through well-defined steps, to approximately solve a problem whose analytical solution does not exist or requires much effort. In this way, numerical methods offer very precisely, although inexact, solutions. Algorithms used in numerical methods need convergence criteria to be defined, that is, an interval where variations are no longer relevant, and it is possible to trust the solution obtained.

Some problems require very complex algorithms to solve, while others require a lot of computational power to run. An example of this is factoring numbers into primes, the heart of cryptography. It is not worth, in this work, to go into details about problems of type P or NP, but almost all security systems are based on the fact that the algorithms we currently have require a lot of computational power to perform the factorization of large numbers, which allows using these numbers, whose prime factors are the key to security, to encrypt information.

Unlike analytical solutions, in the field of fluid mechanics, we can consider all real physical and flow conditions, thus obtaining a real and precise solution, which

allows us to understand the selected model.

To solve the governing differential equations, they need to be discretized using a numerical approach, and the control volume must be refined to smaller cells, thus creating a simulation mesh, where the union of all these cells composes the original control volume (Deglon and Meyer, 2006). The resolution of these equations in each of the cells is done using an appropriate solution method for the type of problem (solver) and must be monitored in some parameters such as consistency, stability and convergence (Sobachkin and Dumnov, 2013). Finally, it is important to note that the solver links the flow parameters of all cells to solve the proposed problem (Mohammadian et al., 2020).

2.4 Machine Learning

One of the main concepts when analyzing a machine learning process is that we can evaluate an improvement in the performance of a machine when performing a certain function due to learning about its own experience. The application of Machine Learning covers several fields, such as robotics, virtual personal assistants (like Alexa, from Amazon), data mining, and natural language processing, among others (Ray, 2019).

In this way, it is possible to define Machine Learning as the field of study that gives computers the ability to learn without necessarily being explicitly programmed. The use of Machine Learning (ML) depends on some algorithms to manage and process data. Currently, there is no one-fits-all algorithm, and it is necessary, for each problem, to make the correct selection of the algorithm that will be used. There are several algorithms, such as Supervised Learning, Unsupervised Learning, Semi-supervised learning, Reinforcement Learning, Multi-task Learning, Ensemble Learning, Neural Network, and instance-based learning (Mahesh, 2020). Figure 2.4 presents the main existing algorithms.

2.4.1 Neural Network

The construction of neural networks is inspired by the functioning of a human brain, where billions of neurons are connected in dense layers. The artificial

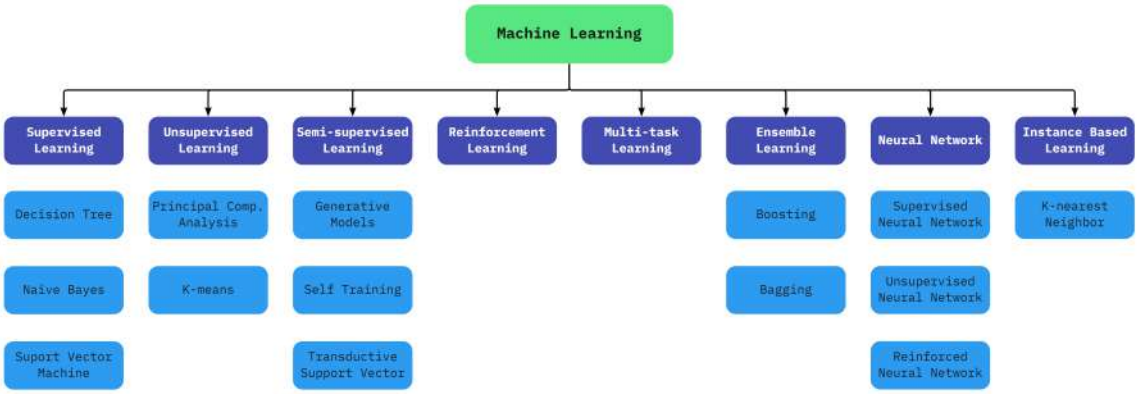


Figure 2.3: Representation of different Machine Learning algorithms, inspired by Mahesh (2020).

neural network consists of a layer of input neurons, a number n of hidden neurons, and a final layer of neurons corresponding to the output. The connection between neurons of different layers is called weight (Wang, 2003). Figure 2.4 represents the architecture of a Neural Network.

The output, h_i, of neuron i in the hidden layer is given by Equation 2.4, where $\rho()$ is called activation function, N the Number of input neurons, V_{ij}, the weights, x_j inputs to the input neurons, and T_i^{hid} the threshold terms of the hidden neurons.

$$h_i = \rho\left(\sum_{j=1}^{N} V_{ij}x_j + T_i^{hid}\right) \tag{2.4}$$

Although several works are using Machine Learning in the study of fluid mechanics, as synthesized in Brunton et al. (2020) and Brunton (2022), in the field of micromixers study, the use of Machine Learning is almost unprecedented. It was possible to find 2 papers that studied this field, both by the same research group and with the same approach. In both Ortega-Casanova and Granados-Ortiz (2020) and Granados-Ortiz and Ortega-Casanova (2021), the use of Machine Learning is proposed as a strategy to reduce the simulation time required to perform an optimization of the geometry of a micromixer. Local optimization is performed on both,

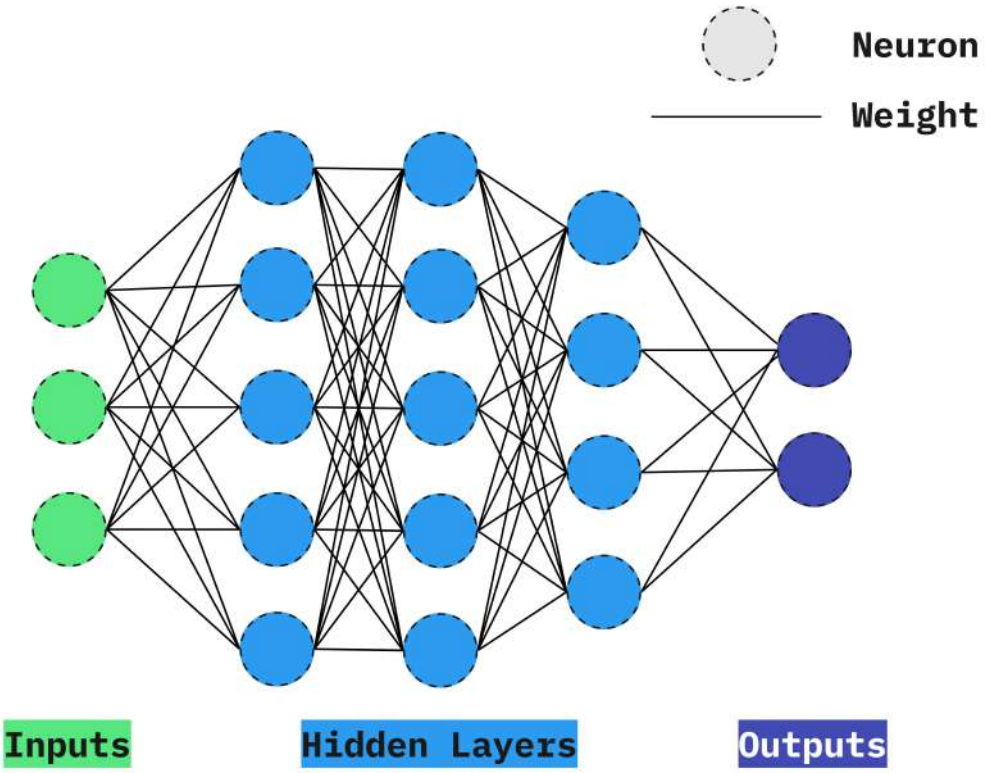


Figure 2.4: Architecture of a Neural Network inspired by Wang (2003).

using the combination of CFD and ML data.

2.5 Optimization Methods

Optimization processes aim to obtain, for a given system, its maximum (or minimum) value, for a series of properties, through the system's variation of properties and dimensions. In this context, several algorithms can be selected for this purpose. The selection of the algorithm for a given problem depends on the objective function characteristic, the nature of the constraints, the number of dependent and independent variables, and the complexity of the problem (Simon, 2013).

The stochastic and evolutionary optimization methods are the most suitable for problems of the nature described above because they have flexibility and robustness, are suitable for solving discrete optimization problems, or in analyzes that are highly nonlinear, high-dimensional, or that present difficulty of being optimized by

classical methods. Among the main methods of stochastic and evolutionary optimization, it is possible to mention some, such as Genetic algorithm (GA), simulated annealing (SA), differential Evolution (DE), tabu Search (TS), artificial bee colony (ABC), and particle swarm optimization (PSO) (Simon, 2013).

2.5.1 Genetic Algorithm

The genetic algorithm, which will be used in the optimization part of this work, is a computational model of biological evolution (Forrest, 1996). GA takes elements from the Darwinian theory of evolution, being a population-based algorithm, where the survival of a population of creatures is tested and their genes stimulated (Goldberg and Holland, 1988).

The GA starts with an initial population, composed of several solutions, which represent the chromosomes of the individuals. There are two ways to define an initial population: random initialization and heuristic initialization. The first uses a completely random initial population, based on some probability distribution, such as the Gaussian distribution. The second indicates that the initial population is composed using known heuristic solutions. The approach usually adopted is the combination of a heuristic set with a random set (Venkateswarlu and Jujjavarapu, 2020).

After defining the Initial Population, 3 simple and sequential operations are performed. The reproduction operation selects the best chromosomes, the crossover operation recombines the best parts of the chromosomes while the mutation operation internally alters the chromosomes to improve their performance. This step, in addition to improving the result, ends up playing a decisive role in optimizing the function globally (and not locally) (Venkateswarlu and Jujjavarapu, 2020).

Finally, GA evaluates the fitness of each individual in a population using a fitness (objective) function and evaluates whether another iteration is necessary, based on the adopted convergence criterion. (Goldberg and Holland, 1988).

Figure 2.5 shows a flowchart of Genetic Algorithm.

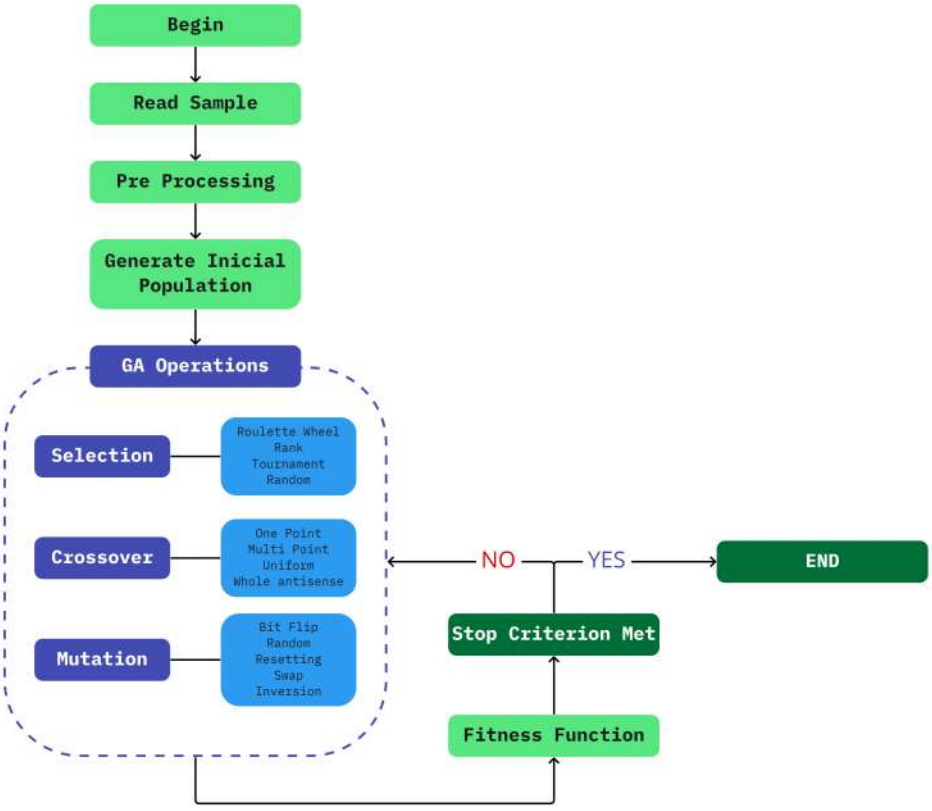


Figure 2.5: Flowchart of Genetic Algorithm inspired by Simon (2013).

Chapter 3

Methodology

Chapter 3 focuses on bringing clarity to the methodology applied in this work. It is divided into several sections, due to the combination of several independent techniques to carry out this work. It is even possible to affirm that the value contributed by this work resides mainly in the proposed variety of methods, which allowed us to obtain results with high precision and low computational cost.

Section 3.1 talks about the construction of geometries to perform the simulations. In this section dimensions, format and variations are discussed. The process of defining geometries and their visualization using ParaView is also briefly discussed.

Section 3.2 addresses the numerical solution of the presented problem using the OpenFOAM software. In addition, the equations entered in the solver and the selected boundary conditions are discussed.

Section 3.3 presents the training of the neural network used, covering the selected parameters. There is a discussion about the use of TensorFlow and about the result expected of some parameters, in addition to discussing a common problem in poorly structured neural networks: overfitting.

Finally, Section 3.4 presents the geometry optimization process, discussing the application of the genetic algorithm for the selection of the geometry that represents the global optimum of the system. This section also discusses some convergence criteria to be adopted.

3.1 Construction of Geometries

As discussed in Chapter 2, it is necessary to make the domain discrete so that it is possible to use a solver and deal with fluid dynamics problems. The geometry construction process was performed using **Gmsh**. Gmsh is an open-source 3D finite element mesh Generation with a built-in CAD engine and post-processor (Geuzaine and Remacle, 2009). As the main mesh generation software (Royer et al., 2021), Gmsh was chosen because it was built following a philosophy of being fast, light and user-friendly (Geuzaine and Remacle, 2009). According to Geuzaine and Remacle (2009), the use of the software takes place through 4 modules:

Geometry: the abstract, object-oriented geometry layer permits to write all the algorithms independently of the underlying CAD representation. At the source code level Gmsh is thus easily extensible by adding support for additional CAD engines.

Mesh: using the abstract geometrical interface it is also possible to interface additional meshing kernels.

Solver: a socket-based communication interface allows to interface Gmsh with various solvers without changing the source code; tailored graphical user interfaces can also easily be added when more fine-grained interactions are needed.

Post-processing: the post-processor can be extended with user-defined operations through dynamically loadable plug-ins. These plug-ins act on post-processing datasets (called *views*) in one of two ways: either destructively changing the contents of a view, or creating one or more views based on the current view.

This work was inspired by the works of Wang et al. (2012) and Rahmannezhad and Mirbozorgi (2019). Because of this, the selection of the geometry, presented in Table 1, was that of a micromixer with Y-geometry, as presented in Figure 3.1. The dimensions of the diameters and offsets of the obstructions were varied in a range on $20 - 100\mu m$ and $0 - 150\mu m$, respectively.

Table 3.1: Dimensions of micromixer.

Channel lenght (L)	$2cm$
Channel width (W)	$200\mu m$
Channel depth (d)	$20\mu m$
Distance between two CGs	$1000\mu m$
Diameter of CGs	$200\mu m$
Distance between two obstructions	$1000\mu m$

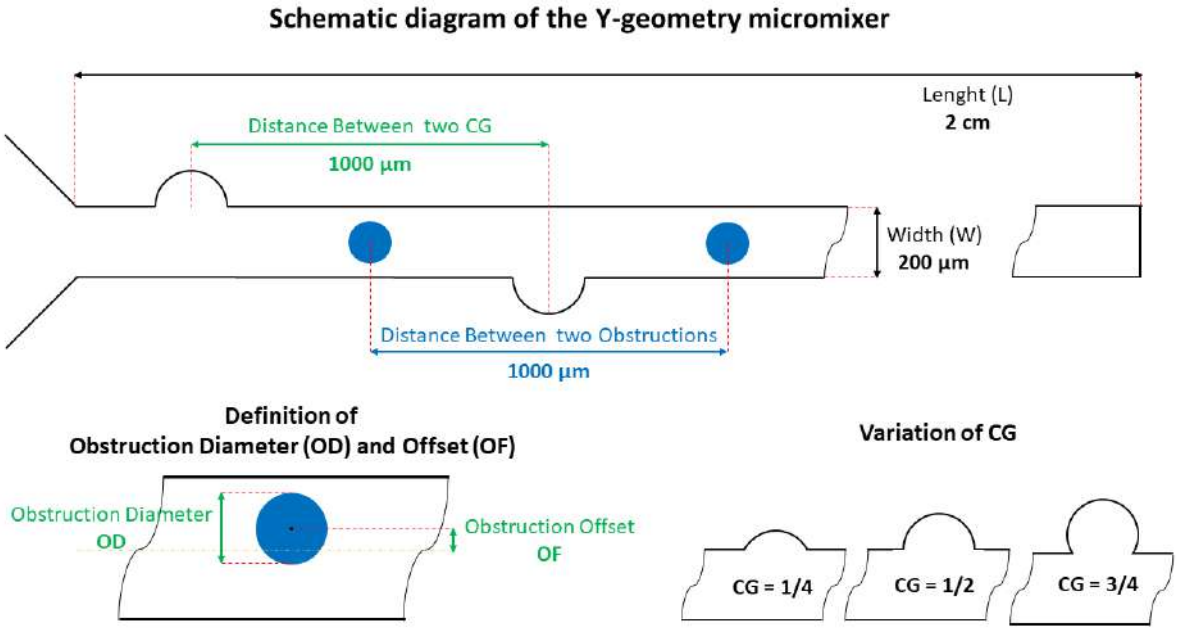


Figure 3.1: Schematic diagram of the Y-geometry micromixer.

The geometries presented by Wang et al. (2012) and addopted by Rahmannezhad and Mirbozorgi (2019) have different groove depths, ranging from 0 to 3/4 cylindrical groove, as presented in Figure 3.

In the present work, geometries with CG=1/4 and CG=1/2 and without obstructions were used for validation and CG=1/2 with obstructions were used to perform the optimization process.

The construction of the mesh in Gmsh was carried out in 3 steps, both in models with obstruction and in models without obstruction: the first one represents the entrance of the fluid in the system (Figure 3.3), the second represents the sequential and repeated parts of the system model, containing in each case the re-

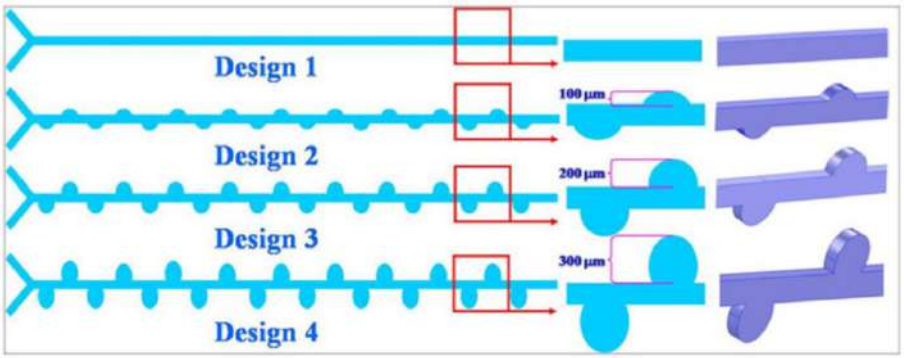


Figure 3.2: Two-dimensional (top view) and three-dimensional (right column) scheme of the four designs studied in Wang et al. (2012). Design 1 is the straight channel and designs 2, 3, and 4 are the channels with CGs 1/4, 1/2 and 3/4, respectively.

spective obstruction and different groove sizes (Figure 3.4), and a last construction to end the geometry of the system (Figure 3.5). Figures 3.3, 3.4, and 3.5 show the constructions that correspond to the system with GC=1/4 and without obstruction, used to validate the simulation with the comparison with the results obtained with Wang et al. (2012). The other geometries were made in a similar way, with different cylindrical groove (using CG=0, CG=1/4, CG=1/2 and CG=3/4), as well as with the presence (or absence) of obstructions, these in different geometries. The final mesh obtained is presented in Figure 3.6, with a detail about the points of interest.

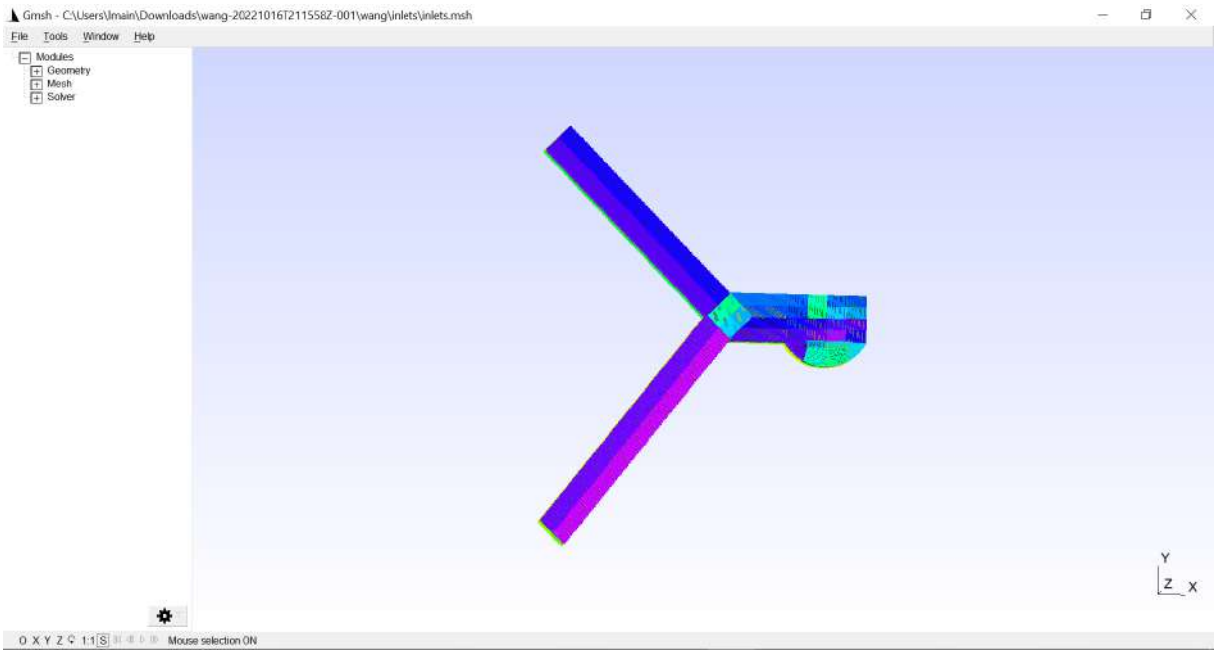


Figure 3.3: Visualization in Gmsh of the fluid input in the proposed geometry.

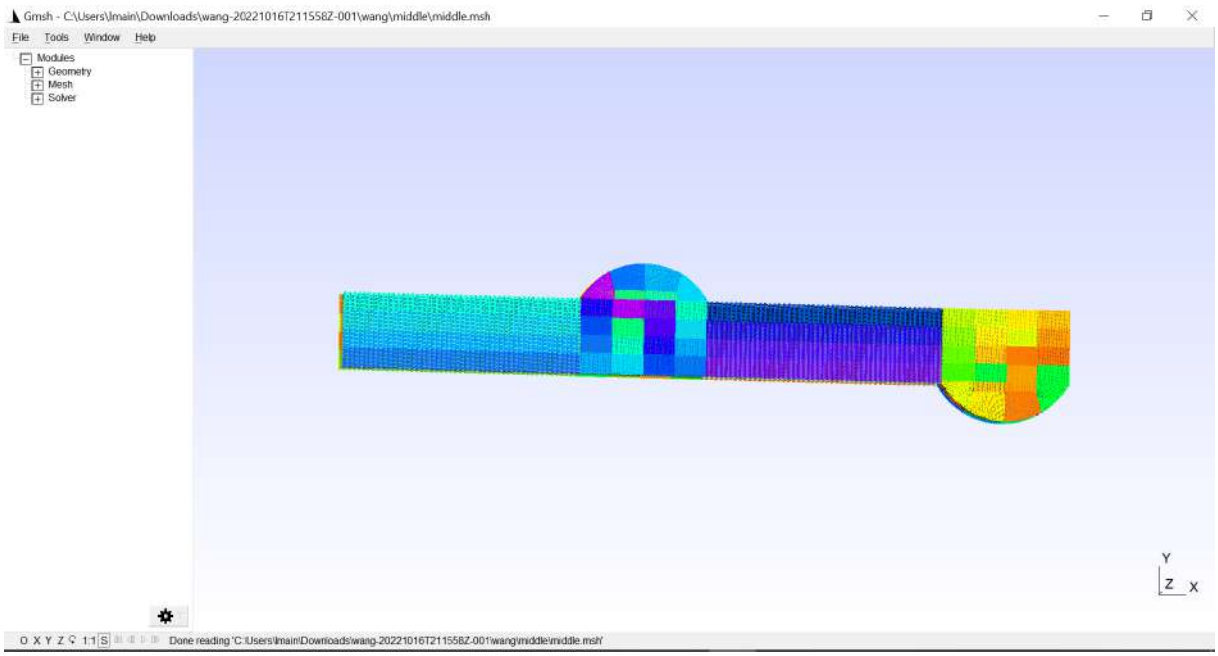


Figure 3.4: Visualization in Gmsh of the first repetition structure in the proposed geometry.

3.2 Simulations Using OpenFOAM

In total, 265 simulations were carried out, where the variation of the OD, OF and the geometry of the obstruction occurred. The simulations were done in a HPC Cluster with procesor Inter Xeon ® E5-2640 v4 2.4GHz, where the training calculation was performed mainly in a Tesla P100 GPU with 16GB VRAM. In this case, each simulation of our dataset, took around 4 hours.

All simulations were performed using OpenFOAM for stationary and incompressible flow using the finite volume method. The obstruction of channels with cylindrical grooves CG = 1/2 was considered to be in a circle shape, in a range of OD obstruction dimensions from 20 to 140μm and OF obstruction displacement from 10 to 160μm (the offset in this case refers to the distance from the bottom point of the circle to the bottom wall of the channel for all possible OD values).

The continuity, Navier-Stokes and convection-diffusion-species equations (Eq. 3.2, Eq. 3.2 and Eq. 3.2) were solved for a uniform laminar flow of a Newtonian fluid with constant properties. The flow has the same velocity at both inputs, but different mass concentrations of solute C, being $C = 1 \quad mol/m^3$ in the lower entry and $C = 0 \quad mol/m^3$ in the upper entry, ie, free of solute.

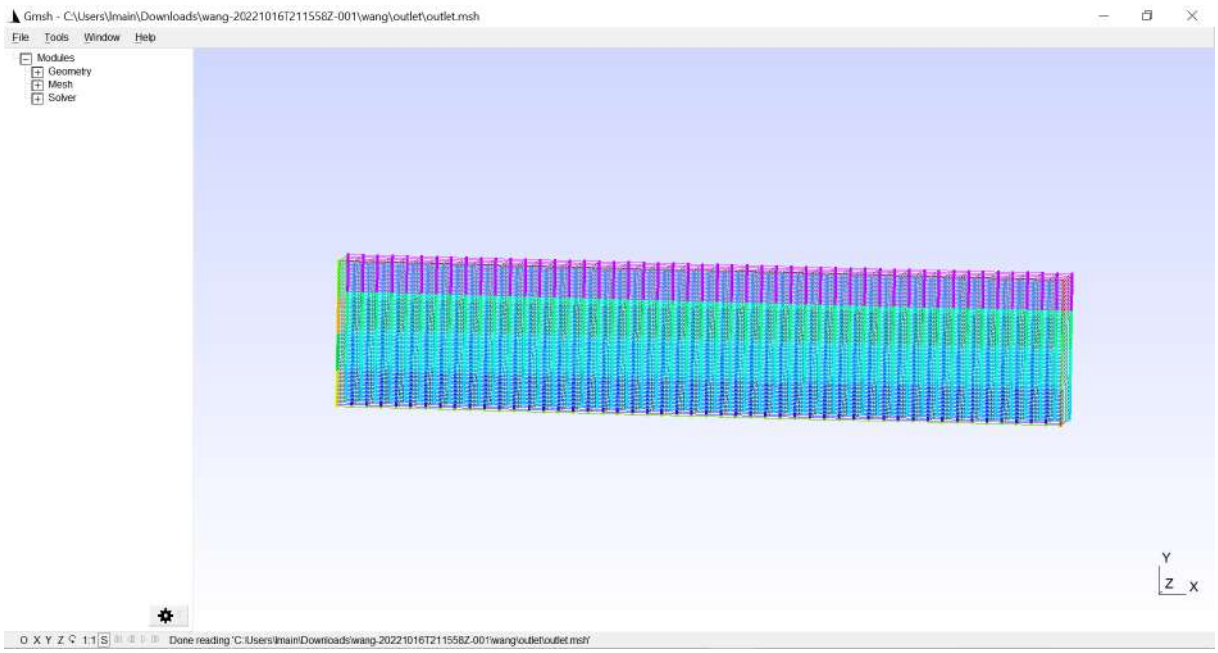


Figure 3.5: Visualization in Gmsh of the fluid output in the proposed geometry.

$$\nabla.\mathbf{u} = 0 \tag{3.1}$$

$$\rho\frac{D\mathbf{u}}{Dt} = -\nabla p + \mu\nabla^2\mathbf{u} \tag{3.2}$$

$$\frac{DC}{Dt} = \gamma\nabla^2 C, \tag{3.3}$$

Where $\overrightarrow{u}$, p and C are, respectively, the velocity, pressure and concentration, and ρ, μ and γ are, respectively, the density, viscosity and fluid diffusion coefficient.

The boundary conditions used in the inputs were normal uniform velocity, based on the Reynolds number (Re), no slip on the walls and gauge pressure ($p = 0Pa$) at the output. The density, dynamic viscosity and diffusion coefficient of the fluid are $\rho = 998kg/m^3$,$\mu = 8.9 \cdot 10^{-4}Pa \cdot s$ and $\gamma = 10^{-9}m^2/s$, respectively. The Reynolds number is defined as $Re = \rho U_{in}W/l$, where U_{in} is the flow inlet velocity and W is the channel width. The Reynolds number was considered $Re \approx 1$ in the validation case and in the other simulations.

The distribution of concentration levels across the width of the main channel can be used to assess the level of fluid mixing in micromixers. The distributions

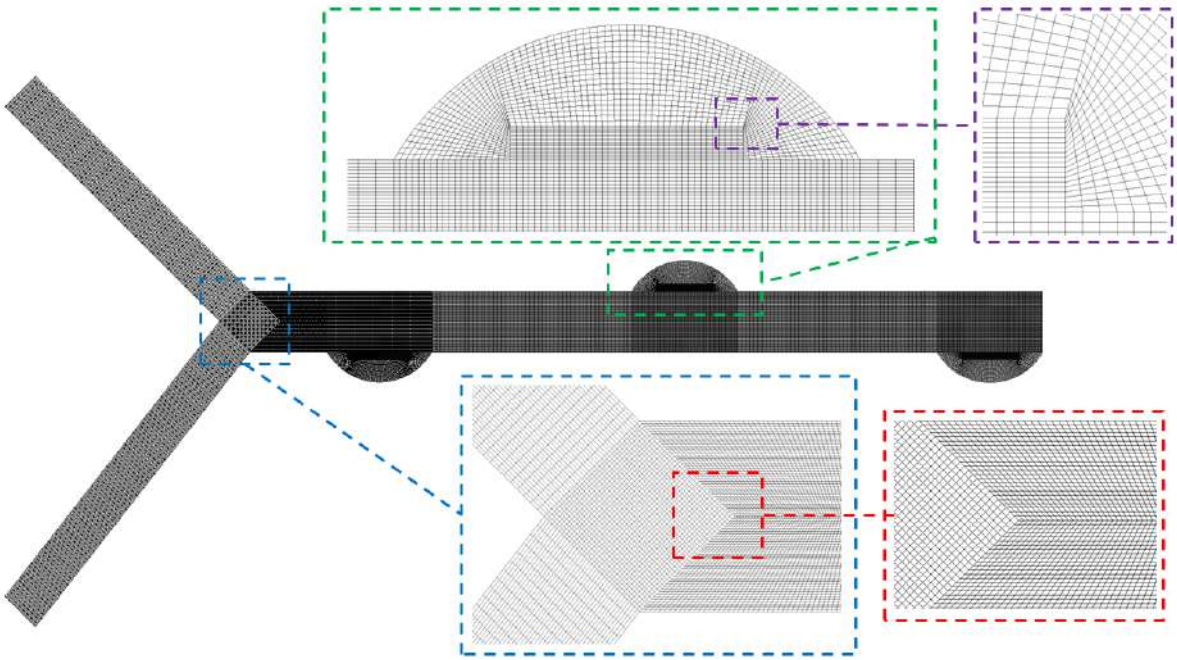


Figure 3.6: Visualization of the final Geometry.

at $x = 0mm$ and $x = 20mm$ were consider to evaluate the pressure drop and the distribution at $x = 16mm$ was consider to evaluate the percentage of mixing, as presented in Figure 4.

Figure 3.7: Measurements to calculate pressure drop ($x = 0mm$ and $x = 20mm$) and blend percentage ($x = 16mm$).

The blend percentage (φ) is determined by the following equation (Lin et al., 2007),

$$\varphi = \left(1 - \frac{\sigma}{\sigma_{max}}\right) \cdot 100\%, \tag{3.4}$$

where σ is the standard deviation, and the subscript max denotes the initial unmixed state in the micromixer (0.5 in this case). The standard deviation can be calculated by the concentration distribution as

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(C_i - \overline{C_i})^2}, \quad \overline{C_i} = \frac{\sum_{i=1}^{N} C_i}{N}, \tag{3.5}$$

where N is the number of sampling across the channel width, C_i is the concentration of sampling i, and $\overline{C_i}$ is the mean value of the concentration. In addition, the mixing energy cost (mec) (Ortega-Casanova, 2017a), is also used to estimate the efficiency of the micromixers, as it measures the pumping power needed to obtain one percent of the mixture. So it can be defined as

$$mec = \frac{Q\Delta P}{\varphi}, \tag{3.6}$$

where Q is the flow rate through the mixing channel and ΔP is the pressure difference between the output and inputs of the channel.As Q is constant, in this work, we will consider only the value of the ratio $\Delta P/\varphi$. The governing equations were solved until the residuals reached below 10^{-9}, which means that all flow properties remain constant throughout the iterations. The methodology of this study is validated by reproducing the results of Wang et al. (2012) and Rahmannezhad and Mirbozorgi (2019).

A summary of the steps performed in this section are shown in Figure 3.8.

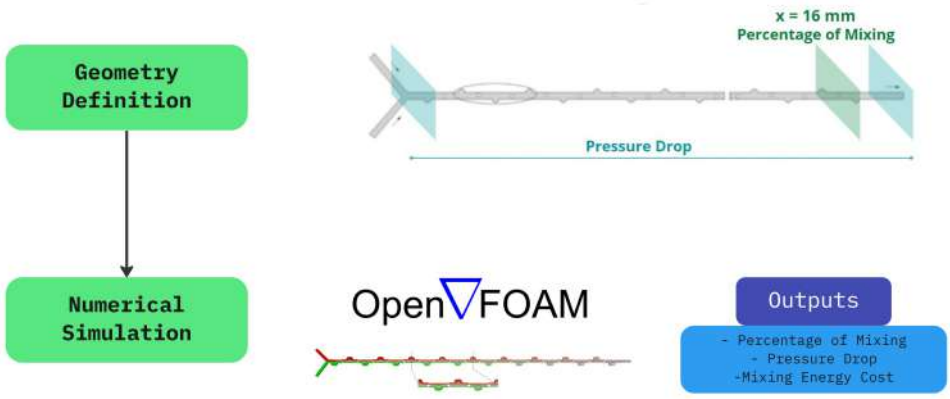


Figure 3.8: Measurements to calculate pressure drop ($x = 0mm$ and $x = 20mm$) and blend percentage ($x = 16mm$).

3.3 Neural Network Training

This section used two completely open-source platforms: TensorFlow and Keras. The first is the platform that allows building deep neural networks, while the second corresponds to an Application Programming Interface (API) that allows simplified access and through simpler programming languages, such as Python.

TensorFlow is a complete open-source platform for machine learning. Basically, it allows the implementation of various mathematical tasks, including training deep neural networks, using various tools, frameworks, and community resources. The development and processing in TensorFlow are performed in C++, allowing high performance in computing operations on matrix and vector. The name TensorFlow originates from the core of the framework used: tensor. Any operation performed in TensorFlow involves using a tensor, an n-dimensional vector, or a matrix representing data of any nature. All values in a tensor contain the same data type with a known shape. The shape of the data is the dimensionality of the matrix or vector.

The data from the simulations performed using CFD made it possible to train a dense neural network using the TensorFlow library. In this neural network, the inputs used were the OD and OF, while the outputs were the pressure drop value and the percentage of mixing. Although we also evaluated the mixing energy cost, the neural network did not calculate this because it is a relationship between the other two variables, as presented in Equation 3.2.

All of the Deep Neural network was made using Python 3.9. At first, it was necessary to import libraries to perform the algebraic operations, graph visualization, machine learning development and data treatment, as shown above (Hunter, 2007; Harris et al., 2020; McKinney et al., 2010; Abadi et al., 2015; Pedregosa et al., 2011).

```
### Library Import
### ------- ###

# Math and Graph visualization libraries
import math
import matplotlib.pyplot as plt
```

```
7 import numpy as np
8 from numpy.random import seed
9 seed(1)
10 import pandas as pd
11 import statsmodels.api as sm
12 import statsmodels.formula.api as smf
13 from numpy.random import randint
14 from numpy.random import rand
15
16 # Machine Learning Libraries
17 import tensorflow
18 tensorflow.random.set_seed(1)
19 from tensorflow.python.keras.layers import Dense
20 from tensorflow.keras.layers import Dropout
21 from tensorflow.python.keras.models import Sequential
22 from tensorflow.python.keras.wrappers.scikit_learn import
      KerasRegressor
23 from sklearn.metrics import mean_absolute_error
24 from sklearn.metrics import mean_squared_error
25 from sklearn.model_selection import train_test_split
26 from sklearn.preprocessing import MinMaxScaler
27
28 # Data tratment libraries
29 import os
30 import csv
31 from pandas.plotting import scatter_matrix
32 from matplotlib import gridspec
```

Listing 3.1: Code for import libraries.

To perform the neural network setup, it is necessary to upload and process the data. The data obtained in the simulations were processed and compiled into a single file. This file is uploaded as shown in line 5, on Listing 3.2 of the code below. Next, the input and output tensors of the neural network are defined. The input tensor (X) comprises each simulation's OD and OF information. The output tensor (Y) includes the values obtained for the Percentage of Mixing and Pressure Drop.

When setting up a neural network, part of the data must be separated for training and another part for validation. In this work, 80% of the data were used

for training and 20% for validation, as shown in line 14 ($test\_size = 0.2$) on Listing 3.2.

Finally, the data are transformed to the $0 - 1$ scale, to perform the neural network training, as shown in lines 17-23 on Listing 3.2.

```
1 ### Pre code preparation
2 ### ------ ###
3
4 # Load Results from simulation:
5 df = pd.read_csv('result_circle.csv')
6 dataset = df.values
7
8 # Define tensor of inputs (X)
9 # Define tensor of known outputs (Y)
10 X = dataset[:,2:4]
11 Y = dataset[:,6:8]
12
13 # Split the dataset for training and testing (proportion of 20% over
      80%)
14 X_train, X_val, y_train, y_val = train_test_split(X, Y, test_size=0.2)
15
16 # Data Transformation
17 scaler_x = MinMaxScaler()
18 scaler_y = MinMaxScaler()
19
20 xtrain_scale=scaler_x.fit_transform(X_train)
21 xval_scale=scaler_x.fit_transform(X_val)
22 ytrain_scale=scaler_y.fit_transform(y_train)
23 yval_scale=scaler_y.fit_transform(y_val)
```

Listing 3.2: Code for tensor preparation.

The definition of the neural network happens, as presented in the code below. This is possible, as discussed, thanks to the use of Keras. The model type (sequential) is defined in line 4, on Listing 3.3. Next, the input layer, of dimension 2 (for reading the X tensor) is added. Finally, a layer with 125 neurons is added, with a dropout equal to 75%. The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. Inputs not set to 0 are scaled up by $1/(1 - rate)$ such that the sum of all infor-

mation is unchanged. Finally, line 8 on Listing 3.3 creates the output layer, with a dimension equal to 2 (tensor Y).

Lines 12 and 13 on Listing 3.3 compile the model and allow its execution. In line 13 of the code a variable is defined as *epochs* = 200. The number of epochs is a hyperparameter that defines the number of times that the learning algorithm will work through the entire training dataset.

```
1 ### Neural Network definition
2 ### ------ ###
3
4 model = Sequential()
5 model.add(Dense(250, input_dim=2, kernel_initializer='normal',
      activation='relu'))
6 model.add(Dense(125, activation='relu'))
7 model.add(Dropout(0.75))
8 model.add(Dense(2, activation='linear'))
9 model.summary()
10
11
12 model.compile(loss='mse', optimizer='adam', metrics=['mse','mae'])
13 history=model.fit(xtrain_scale, ytrain_scale, epochs=200, batch_size
      =100, verbose=1, validation_split=0.2)
```

Listing 3.3: Code for Deep Neural Network setup.

Finally, the code is run to display training and testing loss (lines 8-14, on Listing 3.4). Loss analysis is important to study whether overfitting has occurred. Overfitting is an error that occurs when the neural network has memorized training data but is unable to generalize to new situations. This is evidenced by an error in the training set very close to zero, but a large error when the network is used in other datasets. Neural networks with many layers and neurons, as well as small training datasets, can lead to overfitting. In addition to promoting network reduction, there are resources to prevent overfitting. One of them is the dropout, used in this work, as explained above.

Finally, the validation dataset is predicted to compare with the results obtained in the simulation.

```
1 ### Results from Neural Network Training
```

```
### ------ ###

# Print each Epoch
print(history.history.keys())

# Plot data from training and test (To compare Loss)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()

# Predict outputs for validation dataset
predictions = model.predict(xval_scale)
# Use the Scaler inverse transform to return the results to the
    original format
predictions = scaler_y.inverse_transform(predictions)

# Print tensor of input's validation
print(X_val)
# Print Predictions made over the validation tensor
print(predictions)
```

Listing 3.4: Code for show and export results.

A summary of the steps performed in this section are shown in Figure 3.9.

3.4 Geometry Optimization

The optimization was performed using the Python library Genetic Algorithm, which allows to find for an objective function its minimum point, as explained in Chapter 2.5.1.

The genetic algorithm method used was NSGAII (Deb et al., 2002). For this work, 3 objective functions (φ, ΔP and $\Delta P/\varphi$) were selected and optimized individually, outputting the values OD and OF for each of the 3 optimized geometries. Then, the geometries selected as optimal were simulated using OpenFOAM

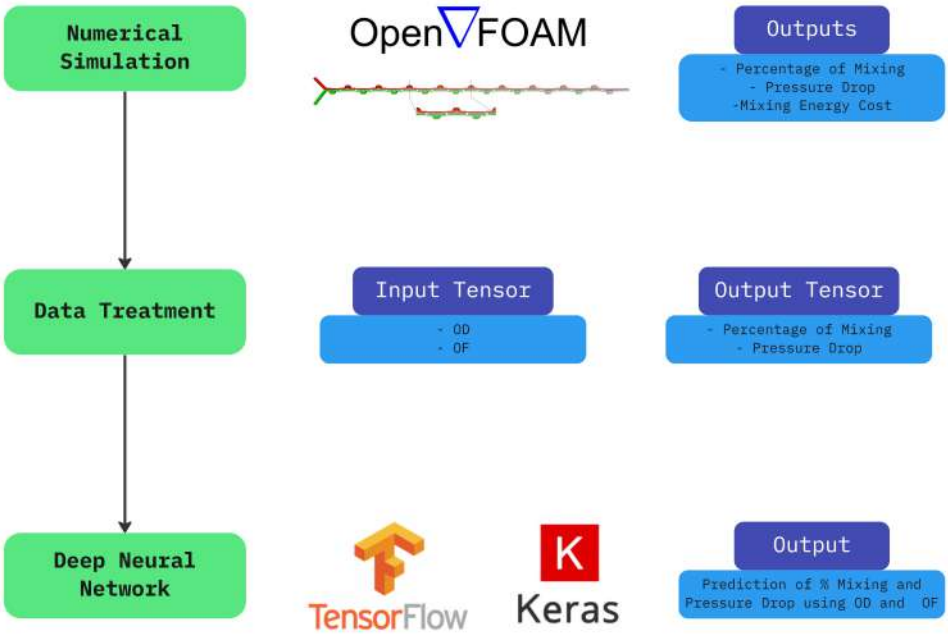


Figure 3.9: Summary of the steps performed to obtain and process data.

to compare the values obtained.

A summary of the steps performed in this section are shown in Figure 3.10.

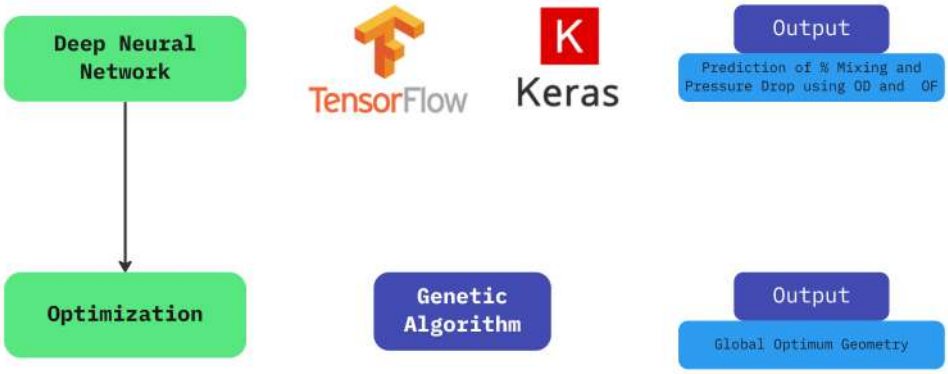


Figure 3.10: Summary of the steps to optimize the geometries.

Chapter 4

Results and Discussion

We performed 265 simulations, changing the OD and OF values, in an HPC Cluster with processor Intel Xeon® E5-2640 v4 2.4GHz, where the training calculation was performed mainly in a Tesla P100 GPU with 16GB VRAM. In this case, each simulation of our dataset, took around 4 hours. Figure 3.1 illustrates one of the cases with simulated obstacles in OpenFOAM.

The simulation results were used to train a dense neural network using the TensorFlow library. This neural network can predict new configurations (changing OD and OF) that can be used to carry out the global optimization process.Therefore, the neural network was trained considering the values of OD and OF as input variables and the values of $\Delta P/\varphi$ and φ as output. With an adequate neural network model, the optimization was performed in order to maximize φ and minimize $\Delta P/\varphi$. The GeneticAlgorithm and Platypus Python libraries were used for the cases of single objective and multiobjective functions, respectively.

Figure 4.1 is a schematic representation of the stages performed in this work.

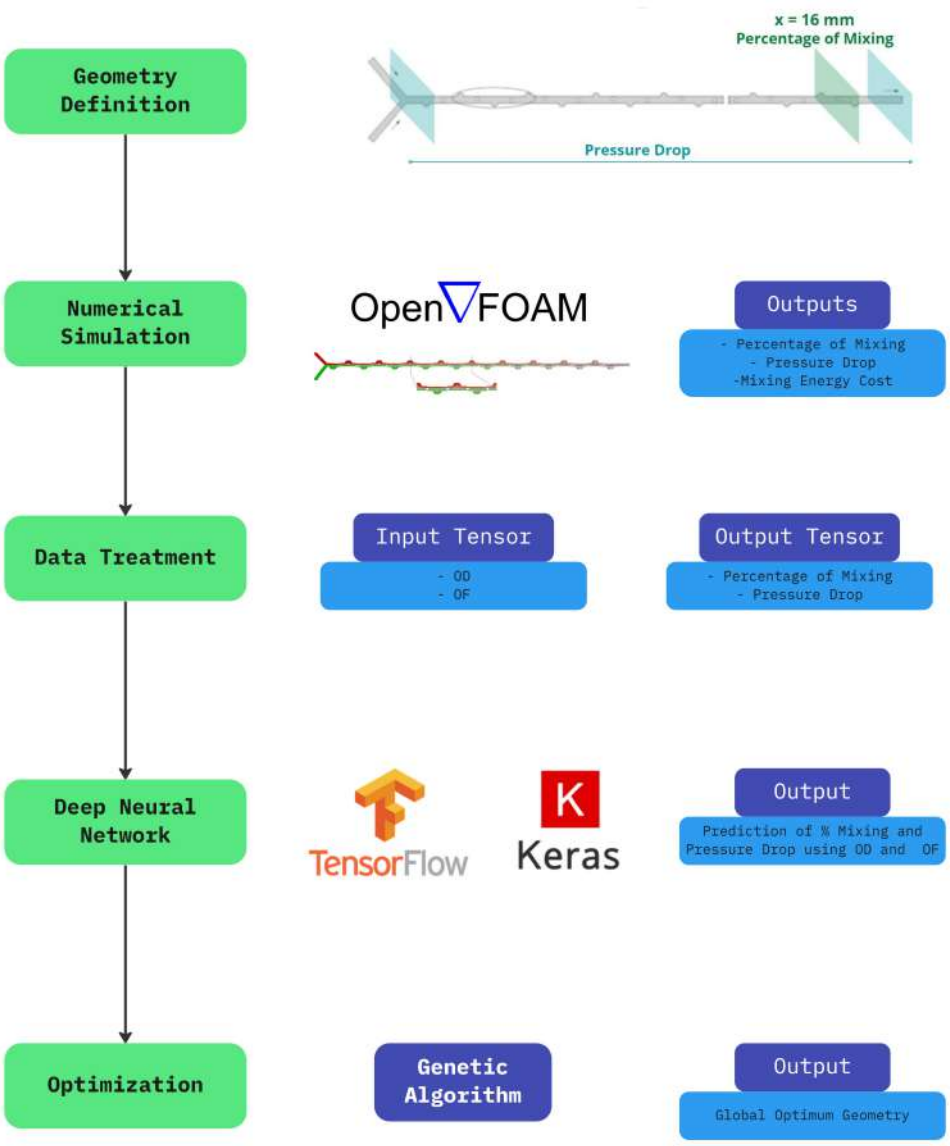


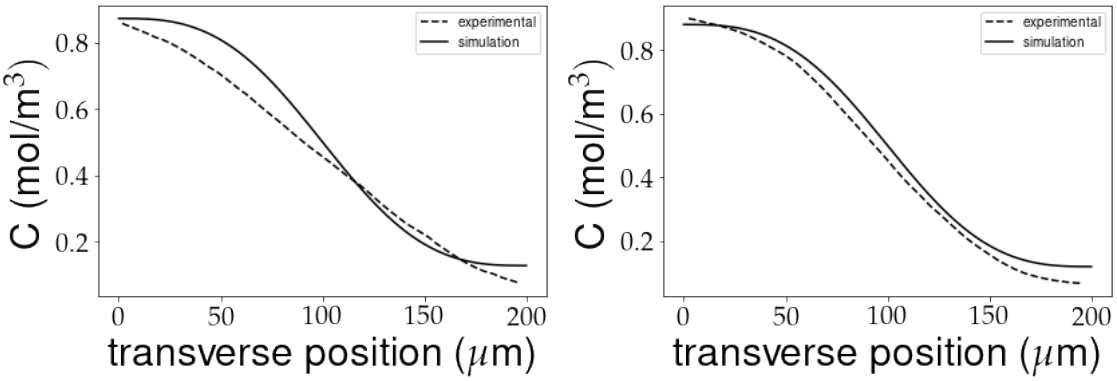
Figure 4.1: Schematic representation of the stages of this work, described in Chapter 3.

4.1 Validation

The concentration distribution at the position $x = 16mm$ was numerically evaluated in this work and compared with the experimental results presented in Wang et al. (2012) and Rahmannezhad and Mirbozorgi (2019), as shown in Figure 4.2. This comparison is necessary to validate the results obtained in this work and is not unprecedented in the literature: Rahmannezhad and Mirbozorgi (2019) uses the same comparison to validate their own results.

The concentration distribution is related to the blend percentage, as shown

in Equation 3.4. The results obtained show that the simulation results are consistent with the experimental results. It is important to emphasize that data from unpublished simulations were used in this work. The comparison with data from the literature served the sole purpose of validating the results of the simulations.



(a) (b)

Figure 4.2: Comparison between experimental and numerical (present study) concentration distribution at the distance of 16 mm for CG (a) 1/4 and (b) 1/2, with $Re = 1$.

In addition, the comparison between the flow lines and the transverse velocity in the cap region was also performed. As can be seen in Figures 4.3 and 4.4, the results obtained computationally showed again a lot of precision in relation to the experimental data.

The data in Figure 4.2 can be used to evaluate the predicted mixture percentage experimentally (Wang et al., 2012) and numerically (this study). Using Eq. 3.4, φ is calculated for CG 1/4 and 1/2, respectively, as 45.5% and 39.1% by the experimental data from Wang et al. (2012), and 42.4% and 41.4% for the numerical data for this study. Considering the methodological differences between the experiment and simulation, the agreement between the results is considered good enough.

(a) (b)

Figure 4.3: Simulated results obtained by Wang et al. (2012) of streamlines and transverse velocity contour of designs CG (a) 1/4 and (b) 1/2, with $Re = 1$.

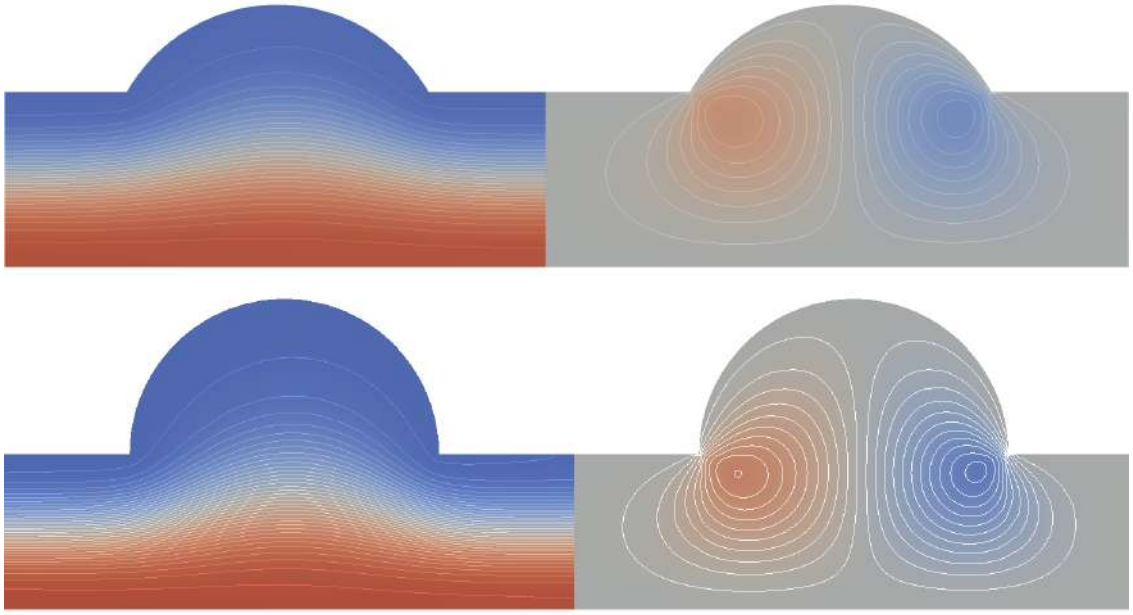


Figure 4.4: Simulated results obtained with OpenFOAM of streamlines and transverse velocity contour of designs CG 1/4 (top) and 1/2 (bottom), with $Re = 1$.

From the results obtained from the simulations in the distribution through the cross section of the micromixer at the position $x = 16mm$, it was possible to define σ, φ and $\Delta P/\varphi$, shown in Figs. 4.5 and 4.6.

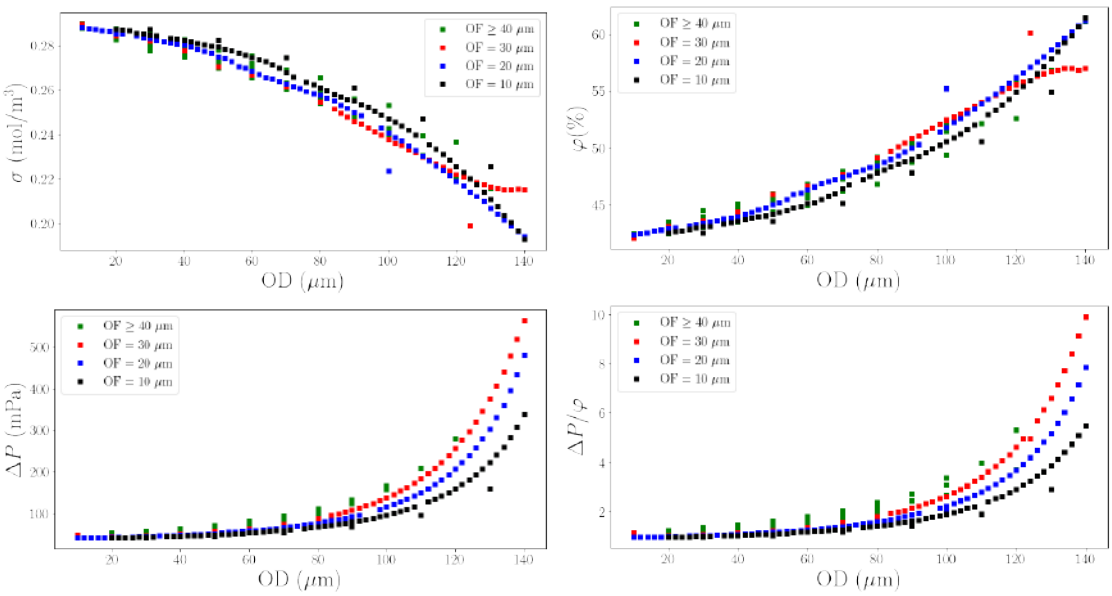


Figure 4.5: Data obtained from simulations: σ, φ, ΔP e $\Delta P/\varphi$ as function of OD.

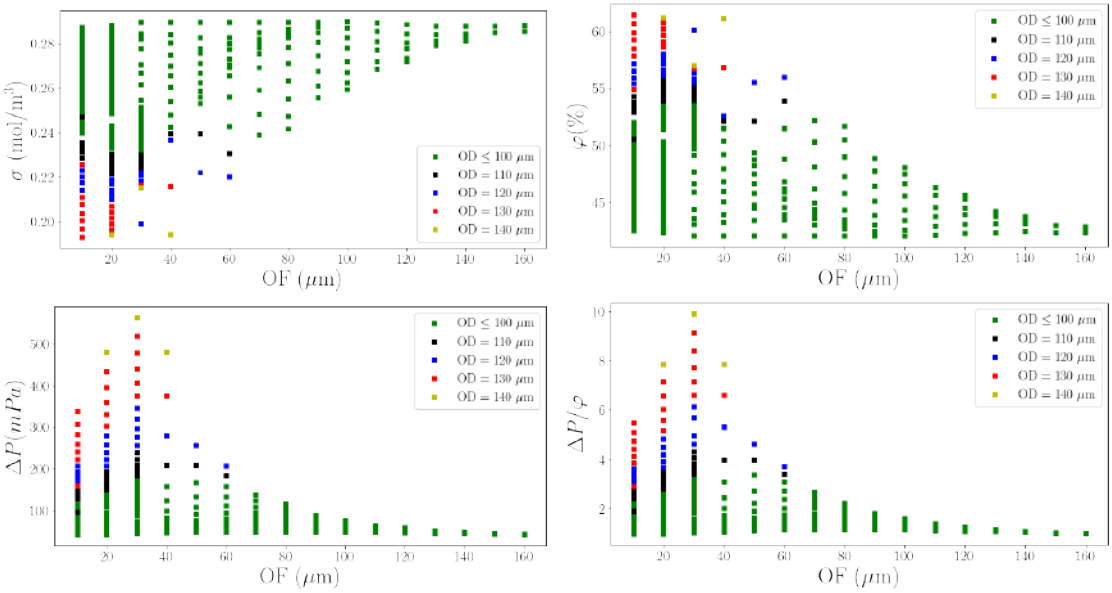


Figure 4.6: Data obtained from simulations: σ, φ, ΔP e $\Delta P/\varphi$ as function of OF.

The values obtained in the simulation performed were compared with those described in the literature, being consistent with what was expected.

Interestingly, as OD increases, both φ, ΔP and $\Delta P/\varphi$ also increase. As OF increases, both φ, ΔP and $\Delta P/\varphi$ tend to decrease. This shows how difficult it is to find OD and OF values that simultaneously satisfy the maximization of φ and the minimization of ΔP and $\Delta P/\varphi$.

The first parameter analyzed was the pressure drop. In Figure 4.5 we can see, in the graph that relates pressure drop and OD, that for the same offset values, we have an increase in pressure drop as the diameter of the obstruction increases. This result is expected since the increase of an obstacle dimension in the same position (OF) would cause a greater pressure drop. Furthermore, in Figure 4.6, the graph that relates pressure drop and OF demonstrates that, for a given value of OD, there is an OF that generates a local maximum point in the pressure drop. A possible explanation for this fact is that, for very small values of OF, we have a system with half flow symmetry, given that it is divided almost equally between the two paths generated by the obstruction. This symmetry may reduce the disturbance generated, in addition to reducing its interaction with the walls of the system, by placing the obstruction at an almost maximum distance between both walls. As the OF increases, we see this symmetry reduce, as well as increase the effect that the wall causes, increasing the disturbance caused. However, after a certain point, the space between the obstruction and the wall becomes so small that the flow through it becomes irrelevant to the system as a whole. In this case, the obstruction no longer behaves like a flow divider and starts to behave similarly to a groove attached to the wall. From that point on, we observed a reduction in the pressure drop.

The analysis performed on the behavior of the blend percentage reached an interesting result, which illustrates the difficulty in optimizing the system. In Figure 4.5, the graph that relates φ with OD, shows that for small values of OF, it is clear that an increase in the value of OD generates an increase in the percentage of mixing. However, when the value of OF is increased, a rapid drop in the blend percentage is observed for all configurations, as can be better visualized in Figure 4.6 in the graph that relates φ with OF, through the green markers. A possible explanation for this phenomenon is that, for low values of OF, greater obstructions create greater perturbations, which increase the area of interaction between the stream flows, improving the blend percentage. However, as the value of OF increases, an asymmetry is created in the system that generates two different flows. In this case, the higher the OD value, the more insignificant the flow that follows the shortest path generated is for the system. Therefore, with the increase in OD, we see a reduction in φ. On the other hand, as the OF increases, the more secondary

this flow becomes, which explains the similarity between the values obtained for the higher values of OF.

Figure 4.7 shows the relationship between ΔP and φ that must be optimized.

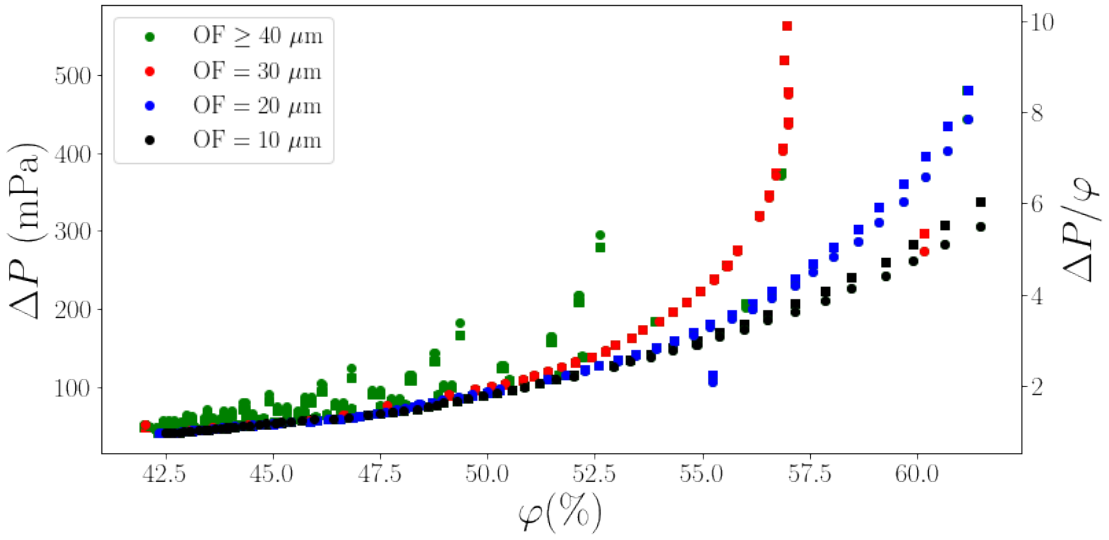


Figure 4.7: Pressure drop ΔP and $\Delta P/\varphi$ as functions of the mixture φ.

4.2 Neural network training and test

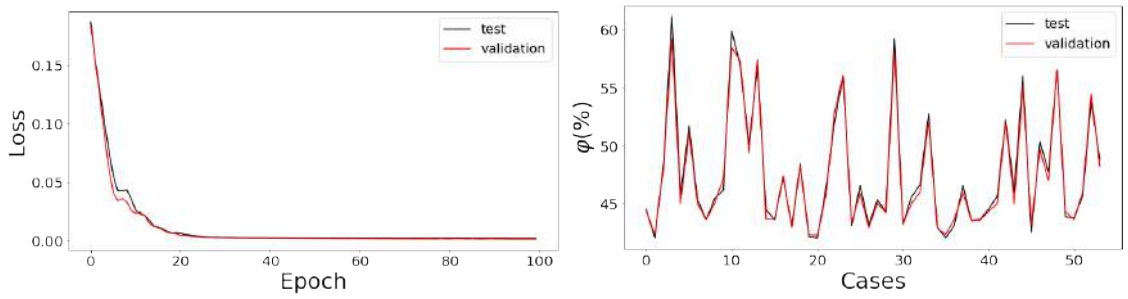
Part of the results shown in Figures 4.5 and 4.6 were used to train a dense neural network, which generated a predictive model that allows, from a given geometry, to predict the values of the pressure drop ΔP, the mixing percentage φ and $\Delta P/\varphi$.

The data obtained in the simulations were organized containing the variables to be used in the neural network, that is, OD and OF for input, φ and ΔP for output. These data were separated into training data and test data with the ratio value 0.2, which means that 20% of the data was be used for test and 80% for training. The selection of the training data is made by an randomized algorithm, as shown in Line 14 at Listing 3.2. Herein, we mapped values in the range from 0 to 1, which ensures that the weight of all variables, both input and output, are the same.

The neural network requires the definition of the model, we defined our dense neural network. It is worth emphasizing here the importance of observing through the graph whether overfitting occurs, which is identified by the training data pre-

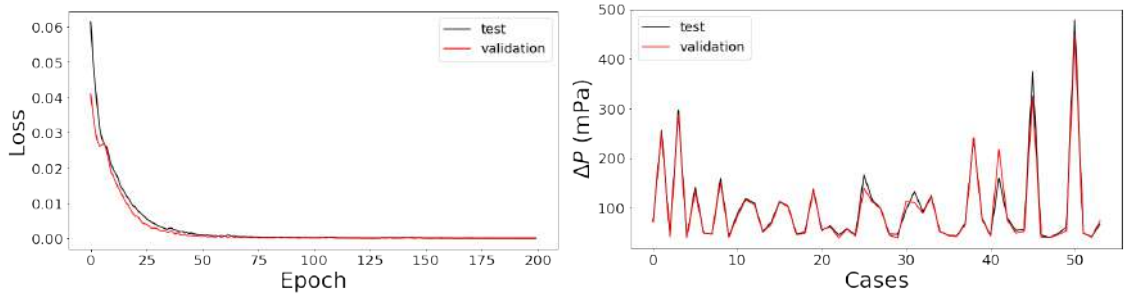
senting a value higher than the curve of the simulation data. Finally, test data can be applied to the found network for its validation.

The mean square error for pressure drop was less than 6.1%, while the mean square error for percentage φ was less than 1.3%. The comparison between the values obtained in the simulations using OpenFOAM (labeled as "test") and those predicted by the neural network (labeled as "validation"), from the OD and OF values, are shown in Figures 4.8 - 4.10.



(a) (b)

Figure 4.8: (a) Loss function for the training data and (b) comparison of the value φ between the test and validation data. The average error calculated was 0.979%.



(a) (b)

Figure 4.9: (a) Loss function for the training data and (b) comparison of the value ΔP between the test and validation data. The average error calculated was 2.604%.

4.3 Optimization and verification

The optimization was first performed using Python's *Geneticalgorithm* library, which allows the definition of only one objective function. Three different

optimizations involving the maximization and minimization of (i) φ, (ii) ΔP and (iii) $\Delta P/\varphi$ were considered.

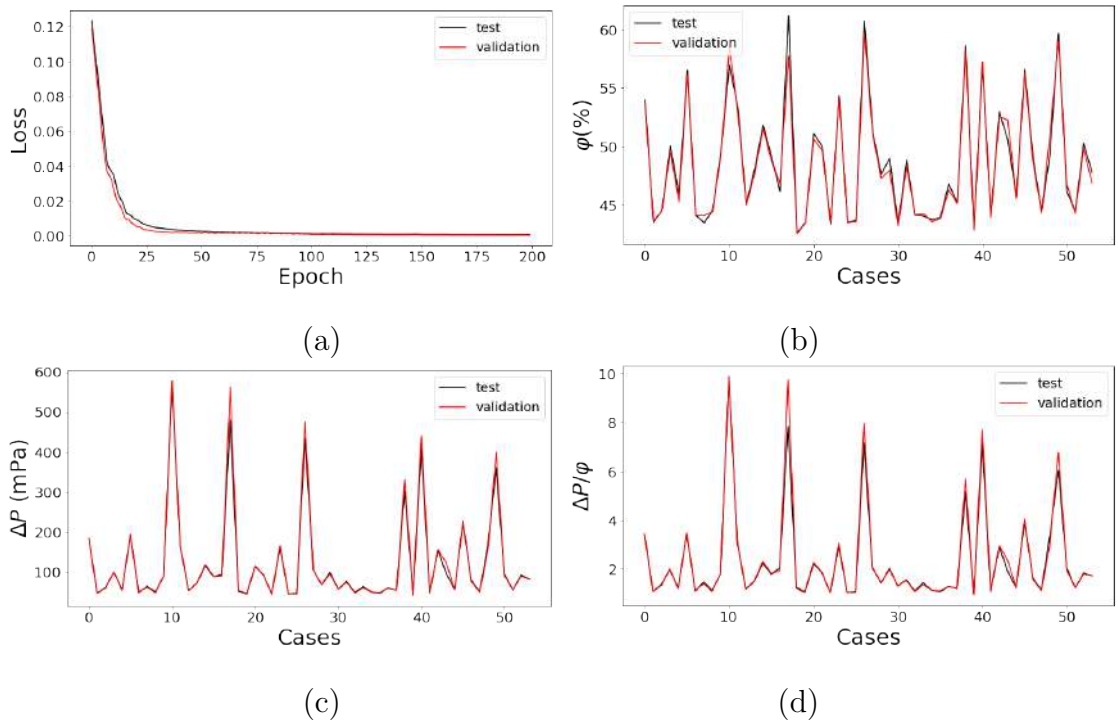


Figure 4.10: (a) Loss function for the training data, comparison of (b) φ, (c) ΔP and (d) $\Delta P/\varphi$ between the test and validation data. The average error calculated was 0.972% and 1.820%, respectively.

As an example, $\Delta P/\varphi$ was chosen as the objective function with the developed network model, the input data is OD and OF and the prediction of these values are used as output of the objective function. Furthermore, the function is subject to a restriction due to the width of the channel, where the diameter of the obstacle must be smaller than this value, that is, $OD + OF > 200\mu m$. If this relationship is satisfied, it must assign a penalty value that will be added to the output of the objective function. The penalty should be a value much higher than those usually obtained at the exit to ensure that this situation is far from the minimum of the function.

If, instead of the global (or local) minimum, the interest is in finding the maximum of the objective function, the only change to be made is in the output signal turning it to minus.

Once the appropriate objective function is defined, the execution of the ge-

netic algorithm is performed. The parameters of this algorithm must be defined, such as population size, mutation and crossover rate and maximum number of iterations. In this problem we chose to exclude the cases with very low ΔP, as they also correspond to low φ ($< 50\%$), which is obtained when considering OD ranging only from 10 to 150 and OF from 10 to 160.

The values obtained for the objective functions, as well as the respective OD and OF are presented in Tables 4.1 and 4.2.

Table 4.1: Maximum values obtained with the optimization.

ObF	Maximum	φ	ΔP	OD (μm)	OF (μm)
φ (%)	59.45	59.45	-	139	41
ΔP (mPa)	582.99	-	582.99	144	51
$\Delta P/\varphi$	9.36	59.59	557.66	142	34
$\Delta P_s + 1/\varphi_s$	111.39	42.69	51.34	102	99

Table 4.2: Minimum values obtained with the optimization.

ObF	Minimum	φ	ΔP	OD (μm)	OF (μm)
φ (%)	42.28	42.28	-	10	76
ΔP (mPa)	40.79	-	40.79	14	16
$\Delta P/\varphi$	0.96	42.71	41.05	11	11
$\Delta P_s + 1/\varphi_s$	1.65	56.91	224.12	127	12

The maximum values found for OD are very similar for φ, ΔP and $\Delta P/\varphi$, that is, around $140\mu m$, while the values of OF are small, not exceeding $34\mu m$. That is, they represent a geometry with large obstacles and close to the channel wall. For $\Delta P_s+1/\varphi_s$, the found OD value is smaller, $102\mu m$, with OF $99\mu m$. This corresponds to geometries with diameters around half the width of the channel, which are also located close to the channel wall. Note that in the latter case, the values for φ and ΔP are much smaller than for $\Delta P/\varphi$ which results in 1.20.

For the minimal cases, the OD values for φ, ΔP and $\Delta P/\varphi$ are also small, between 10 and $14\mu m$, while OF varies between 11 and $76\mu m$. In other words, small obstacles close or not to the wall generate flow with very low values for both

variables, what should be expected. On the other hand, $\Delta P_s + 1/\varphi_s$ presents a higher OD value, $127\mu m$ for a small OF value, $12\mu m$. Note that ΔP_s and φ_s are normalized to stay between 0 and 1. While the minimum value of $\Delta P/\varphi$ has low values for both ΔP and φ, the same does not occur for its maximum value, which indicates that this objective function can be used to obtain a condition of φ high and ΔP as low as possible.

These optimization results correspond to the adoption of a single objective function, φ, ΔP or $\Delta P/\varphi$. Note that, according to Figures 4.5 and 4.6, both φ and ΔP increase with the value of OD and decrease with the value of OF. Thus, the objective of finding OD and OF values that simultaneously satisfy the conditions of φ maximum, ΔP minimum and $\Delta P/\varphi$ minimum cannot be achieved as the method shown above. It becomes necessary to use a multiobjective genetic algorithm, which was done using Python's Platypus library. The three objective functions, in addition to these functions, must return the same restriction condition used previously. One must inform the number of input variables, objective functions and constraints to the problem. Furthermore, the restriction contained in the objective function output must be complemented so that the range of possible values for the input variables is well defined. The genetic algorithm method used was NSGAII (Deb et al., 2002). At the end, the feasible solutions were filtered out of all the possible solutions found. Pareto curves, as well as simulation data, are shown in Figure 4.11 for easy comparison.

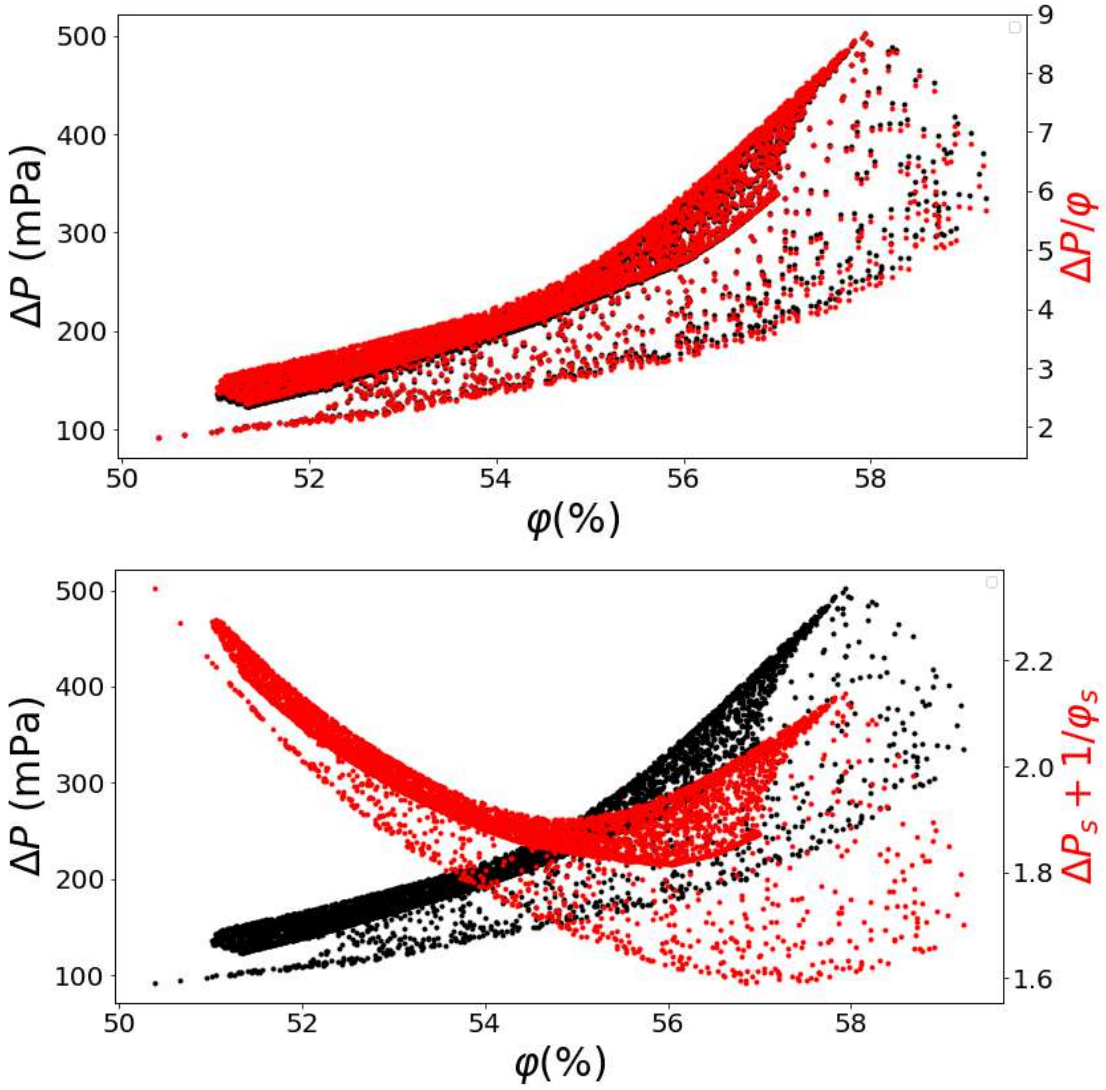


Figure 4.11: Pareto curves that maximize φ and minimize ΔP and $\Delta P/\varphi$ or $(\Delta P_s + 1/\varphi_s)$.

The first curves shown in Figure 4.11 correspond to the best possible values of the objective functions, ranging from the prioritization of minimizing ΔP and $\Delta P/\varphi$ to maximizing φ. The value found for φ_{max} also corresponds to the highest value of ΔP. Likewise, the value found for ΔP_{min} also corresponds to the smallest value of φ. In this work, we will consider, in addition to the optimal point obtained from $(\Delta P_s + 1/\varphi_s)_{min}$, the points with $\Delta P/\varphi$ minimum and φ maximum. In Table 4.3, the chosen case is summarized, as well as the results obtained in the verification simulation.

From the OD and OF values obtained for $(\Delta P_s + 1/\varphi_s)_{min}$ and observing Figures 4.5 and 4.6, it can be noted that the case obtained refers to geometry

Table 4.3: Comparison between the values obtained with optimization and simulation.

ObF	OD (μm)	OF (μm)	φ_{opt}	ΔP_{opt} (mPa)	φ_{sim}	ΔP_{sim} (mPa)
$(\Delta P_s + 1/\varphi_s)_{min}$	131	10	57.47	235.87	57.98	227.67

close to the channel wall, with a medium-sized obstacle. In this region, ΔP tends to decrease and φ remains large, as observed in Rahmannezhad and Mirbozorgi (2019). Thus, the optimized geometry is compatible with what is expected, that is, obstacles that are not too big or too small and close to the channel wall should be prioritized.

Figures 4.12 and 4.13 show the concentration, velocity and pressure profiles obtained through the simulation for the optimal case.

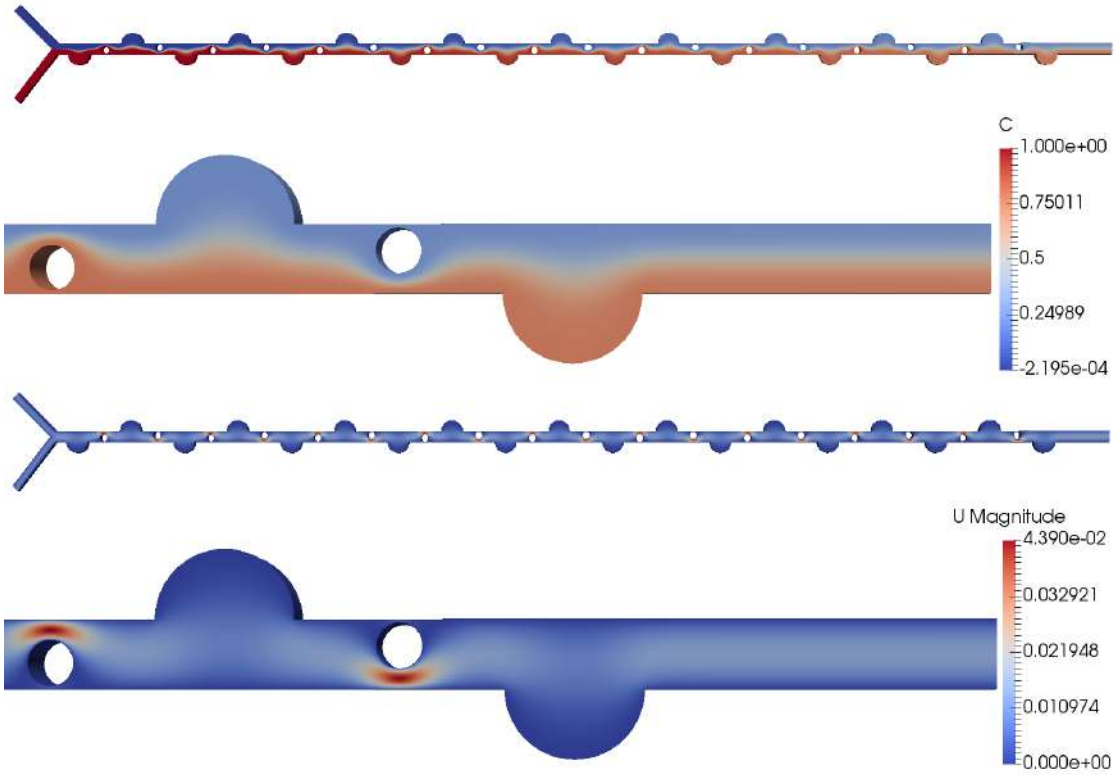


Figure 4.12: Concentration (top) and velocity (bottom) profiles of the optimal cases.

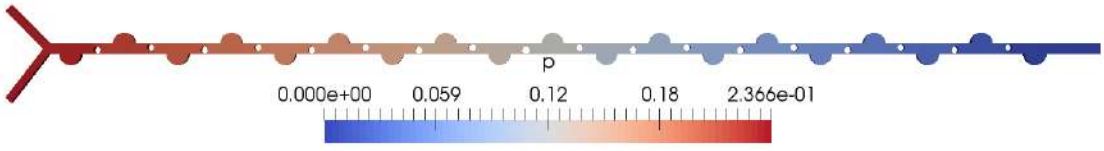


Figure 4.13: Pressure profile of the optimal case.

The errors obtained for φ and ΔP, respectively, for the optimized case were 0.88% and 3.60%. Both the predicted values of φ and ΔP are very accurate, specially for the optimum case. The error for the maximum and minimum cases can be related to the limits of the multivariable method applied.

Chapter 5

Conclusion

A total of 265 different configurations was simulated for a micromixer with Y-geometry micromixer. Originally, one of the goals of this work was to perform simulations using CFD to allow global optimization of the geometry of this micromixer. However, this set of simulations required about 45 days of full computing power and was not sufficient to perform such an optimization process.

In this sense, it was assumed that the use of a dense neural network built from the data obtained with CFD would allow solving the problem of computation time versus precision. In this way, it would be possible to use reliable and accurate data to feed a tool whose computation time is on the order of milliseconds. The goal was to create a system with the computational speed offered by neural networks and the precision achieved by using CFD.

Based on the results of these simulations, which evaluate two independent parameters: blend percentage and pressure drop, which lead to the calculation of the mixing energy cost, it was possible to train a dense neural network that generates a predictive model that allows to predict from a given geometry the values of pressure drop, percentage of mixing and *mec*. The average error for percentage of mixing was less than 1%, while the average error for pressure drop was less than 3%.

The results obtained point to the construction of an optimal micromixer with $OD = 131\mu m$ and $OF = 10\mu m$ for $Re = 1$. This corresponds to a medium size obstacle that is close to the wall, which is compatible with previous results in the literature.

When verifying through the simulation the values of φ and ΔP resulting

from the optimization for the optimal case, we get errors of 0.88% and 3.60%, respectively. Besides the small erros, the values $\varphi = 57.975\%$ and $\Delta P = 227.668$ mPa of the simulated optimal geometry are even larger/smaller than the forecast of 57.47% and 297.46 mPa.

Finally, the use of Machine Learning, more specifically neural networks and genetic algorithm proved to be effective in the study of problems involving optimization, being this approach practically unprecedented in the literature. It is important to emphasize that global optimization could be obtained by simulating 2000 cases using CFD.

However, as each simulation takes around $4h$, the total time to guarantee the global optimization would be about 330 days. With this methodology, the whole process of producing the dataset, training and optimization takes 45 days. This procedure is a tremendous advantage for microfluidic optimization. It should also be emphasized that despite the case studied applied with a geometric optimization, the methodology will be the same for cases involving the optimization of any other parameters, such as input flow or diffusion coefficient, for example.

It is also important to note that this work used software, programs, and systems for its development. The most prominent are: Gmsh 4.11 for the construction of geometries, OpenFOAM 9 (Weller et al., 1998) for the solution of fluid dynamics problems, Python 3.9 for data processing, Keras 2.9 (Chollet et al., 2015), and TensorFlow 2.9 (Abadi et al., 2015) for the construction of neural networks. In addition, we almost only used operating systems that used the Linux Kernel (mainly 5.4 Kernel, using Ubuntu 18.04.5 LTS and Ubuntu 20.04.1 LTS) on this work. All those listed above have in common the fact of being open source, which made it possible, free of charge, to exercise science and learning. We believe that science and human knowledge have a value infinitely above their monetary value, and it is of great importance to do a course conclusion work almost entirely using open-source systems.

Bibliography

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

V. E. Ahmadi, I. Butun, R. Altay, S. R. Bazaz, H. Alijani, S. Celik, M. E. Warkiani, and A. Koşar. The effects of baffle configuration and number on inertial mixing in a curved serpentine micromixer: experimental and numerical study. Chemical Engineering Research and Design, 168:490–498, 2021.

A. Alam, A. Afzal, and K.-Y. Kim. Mixing performance of a planar micromixer with circular obstructions in a curved microchannel. Chemical Engineering Research and Design, 92(3):423–434, 2014.

H. Andersen and B. Hepburn. Scientific method. 2015.

J. D. Anderson and J. Wendt. Computational fluid dynamics, volume 206. Springer, 1995.

A. Arjun, R. Ajith, and S. Kumar Ranjith. Mixing characterization of binary-coalesced droplets in microchannels using deep neural network. Biomicrofluidics, 14(3):034111, 2020.

D. J. Beebe, G. A. Mensing, and G. M. Walker. Physics and applications of mi-

crofluidics in biology. Annual review of biomedical engineering, 4(1):261–286, 2002.

M. M. A. Bhutta, N. Hayat, M. H. Bashir, A. R. Khan, K. N. Ahmad, and S. Khan. Cfd applications in various heat exchangers design: A review. Applied Thermal Engineering, 32:1–12, 2012.

O. Bilsel, C. Kayatekin, L. A. Wallace, and C. R. Matthews. A microchannel solution mixer for studying microsecond protein folding reactions. Review of scientific instruments, 76(1):014302, 2005.

S. L. Brunton. Applying machine learning to study fluid mechanics. Acta Mechanica Sinica, pages 1–9, 2022.

S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. Annual Review of Fluid Mechanics, 52:477–508, 2020.

H. Bruus. Theoretical microfluidics, volume 18. Oxford university press, 2007.

L. Capretto, W. Cheng, M. Hill, and X. Zhang. Micromixing within microfluidic devices. Microfluidics, pages 27–68, 2011.

Y. Chen, X. Chen, and S. Liu. Numerical and experimental investigations of novel passive micromixers with fractal-like tree structures. Chemical Physics Letters, 747:137330, 2020.

S. Cheng, W. Chen, and P. Zhang. Developing advanced polymer films based on microfluidic laminar flow. Giant, page 100091, 2022.

F. Chollet et al. Keras, 2015. URL `https://github.com/fchollet/keras`.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation, 6 (2):182–197, 2002.

D. A. Deglon and C. J. Meyer. Cfd modelling of stirred tanks: Numerical considerations. Minerals Engineering, 19(10):1059–1068, 2006.

M. Dhaubhadel. Cfd applications in the automotive industry. 1996.

J. Ducree, T. Brenner, S. Haeberle, T. Glatzel, and R. Zengerle. Multilamination of flows in planar networks of rotating microchannels. Microfluidics and Nanofluidics, 2(1):78–84, 2006.

J. A. Foroushani and M. Sabzpooshani. Determination of hydrodynamic derivatives of an ocean vehicle using cfd analyses of synthetic standard dynamic tests. Applied Ocean Research, 108:102539, 2021.

S. Forrest. Genetic algorithms. ACM Computing Surveys (CSUR), 28(1):77–80, 1996.

R. W. Fox, A. T. McDonald, and J. W. Mitchell. Fox and McDonald's introduction to fluid mechanics. John Wiley & Sons, 2020.

C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. International journal for numerical methods in engineering, 79(11):1309–1331, 2009.

D. Goldberg and J. Holland. Genetic algorithms and machine learning. 3 (2): 95-99, 1988.

F.-J. Granados-Ortiz and J. Ortega-Casanova. Machine learning-aided design optimization of a mechanical micromixer. Physics of Fluids, 33(6):063604, 2021.

A. Groisman and V. Steinberg. Efficient mixing at low reynolds numbers using polymer additives. Nature, 410(6831):905–908, 2001.

P. Hadikhani, N. Borhani, S. M. H. Hashemi, and D. Psaltis. Learning from droplet flows in microfluidic channels using deep neural networks. Scientific reports, 9(1): 1–7, 2019.

S. Hardt, K. Drese, V. Hessel, and F. Scho¨ nfeld. Passive micro mixers for applications in the micro reactor and μtas field. In International Conference on Nanochannels, Microchannels, and Minichannels, volume 41642, pages 45–55, 2004.

C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer,

M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. Nature, 585:357–362, 2020. doi: 10.1038/s41586-020-2649-2.

S. J. Haswell, R. J. Middleton, B. O'Sullivan, V. Skelton, P. Watts, and P. Styring. The application of micro reactors to synthetic chemistry. Chemical Communications, (5):391–398, 2001.

A. N. Hellman, K. R. Rau, H. H. Yoon, S. Bae, J. F. Palmer, K. S. Phillips, N. L. Allbritton, and V. Venugopalan. Laser-induced mixing in microfluidic channels. Analytical chemistry, 79(12):4484–4492, 2007.

J. D. Hunter. Matplotlib: A 2d graphics environment. Computing in Science & Engineering, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

A. Jafari, M. Hasani, M. Hosseini, and R. Gharibshahi. Application of cfd technique to simulate enhanced oil recovery processes: current status and future opportunities. Petroleum Science, 17(2):434–456, 2020.

M. Jain and K. Nandakumar. Optimal patterning of heterogeneous surface charge for improved electrokinetic micromixing. Computers & Chemical Engineering, 49: 18–24, 2013.

S. A. Khan, A. Aabid, and M. A. A. Baig. Cfd analysis of cd nozzle and effect of nozzle pressure ratio on pressure and velocity for suddenly expanded flows. International Journal of Mechanical and Production Engineering Research and Development, 8:1147–1158, 2018.

R.-w. Kim, J.-g. Kim, I.-b. Lee, U.-h. Yeo, S.-y. Lee, and C. Decano-Valentin. Development of three-dimensional visualisation technology of the aerodynamic environment in a greenhouse using cfd and vr technology, part 1: Development of vr a database using cfd. Biosystems Engineering, 207:33–58, 2021.

R. Langtry and F. Menter. Transition modeling for general cfd applications in aeronautics. In 43rd AIAA aerospace sciences meeting and exhibit, page 522, 2005.

I.-B. Lee, J. P. P. Bitog, S.-W. Hong, I.-H. Seo, K.-S. Kwon, T. Bartzanas, and M. Kacira. The past, present and future of cfd for agro-environmental applications. Computers and electronics in agriculture, 93:168–183, 2013.

Y. Lin. Numerical characterization of simple three-dimensional chaotic micromixers. Chemical Engineering Journal, 277:303–311, 2015.

Y.-C. Lin, Y.-C. Chung, and C.-Y. Wu. Mixing enhancement of the passive microfluidic mixer with j-shaped baffles in the tee channel. Biomedical microdevices, 9 (2):215–221, 2007.

M. Maeki, S. Uno, A. Niwa, Y. Okada, and M. Tokeshi. Microfluidic technologies and devices for lipid nanoparticle-based rna delivery. Journal of Controlled Release, 2022. ISSN 0168-3659. doi: https://doi.org/10.1016/j.jconrel.2022.02.017. URL https://www.sciencedirect.com/science/article/pii/S0168365922000918.

B. Mahesh. Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9:381–386, 2020.

M. B. Martínez, E. Pereyra, and N. Ratkovich. Cfd study and experimental validation of low liquid-loading flow assurance in oil and gas transport: studying the effect of fluid properties and operating conditions on flow variables. Heliyon, 6 (12):e05705, 2020.

W. McKinney et al. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference, volume 445, pages 51–56. Austin, TX, 2010.

M. Micheletti and G. J. Lye. Microscale bioprocess optimisation. Current opinion in biotechnology, 17(6):611–618, 2006.

E. M. Miller and A. R. Wheeler. A digital microfluidic approach to homogeneous enzyme assays. Analytical Chemistry, 80(5):1614–1619, 2008.

P. A. Mirzaei. Cfd modeling of micro and urban climates: Problems to be solved in the new decade. Sustainable Cities and Society, page 102839, 2021.

A. Mohammadian, H. Kheirkhah Gildeh, and I. Nistor. Cfd modeling of effluent discharges: A review of past numerical studies. Water, 12(3):856, 2020.

P. D. Morris, A. Narracott, H. von Tengg-Kobligk, D. A. S. Soto, S. Hsiao, A. Lungu, P. Evans, N. W. Bressloff, P. V. Lawford, D. R. Hose, et al. Computational fluid dynamics modelling in cardiovascular medicine. Heart, 102(1):18–28, 2016.

A. Nagaki, K. Kawamura, S. Suga, T. Ando, M. Sawamoto, and J.-i. Yoshida. Cation pool-initiated controlled/living polymerization using microsystems. Journal of the American Chemical Society, 126(45):14702–14703, 2004.

P. Negi and M. Subhash. Method to control flow separation over wind turbine blade: A cfd study. Materials Today: Proceedings, 2021.

M. Nimafar, V. Viktorov, and M. Martinelli. Experimental comparative mixing performance of passive micromixers with h-shaped sub-channels. Chemical engineering science, 76:37–44, 2012a.

M. Nimafar, V. Viktorov, and M. Martinelli. Experimental comparative mixing performance of passive micromixers with h-shaped sub-channels. Chemical engineering science, 76:37–44, 2012b.

T. Norton and D.-W. Sun. An overview of cfd applications in the food industry. 2007.

J. Ortega-Casanova. Enhancing mixing at a very low reynolds number by a heaving square cylinder. Journal of Fluids and Structures, 65:1–20, 2016.

J. Ortega-Casanova. Application of cfd on the optimization by response surface methodology of a micromixing unit and its use as a chemical microreactor. Chemical Engineering and Processing: Process Intensification, 117:18–26, 2017a.

J. Ortega-Casanova. Cfd study on mixing enhancement in a channel at a low reynolds number by pitching a square cylinder. Computers & Fluids, 145:141–152, 2017b.

J. Ortega-Casanova and F.-J. Granados-Ortiz. Using machine-learning to speed-up optimisation in cfd: designing a micromixer. In APS Division of Fluid Dynamics Meeting Abstracts, pages F13–005, 2020.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

R. R. Pethig and S. Smith. Introductory Bioelectronics: For Engineers and Physical Scientists. John Wiley & Sons, 2012.

T. Powers. Heisenberg's war. the secret history of the german bomb. 1993.

S. Preetam, B. K. Nahak, S. Patra, D. C. Toncu, S. Park, M. Syväjärvi, G. Orive, and A. Tiwari. Emergence of microfluidics for next generation biomedical devices. Biosensors and Bioelectronics: X, page 100106, 2022.

J. Rahmannezhad and S. A. Mirbozorgi. Cfd analysis and rsm-based design optimization of novel grooved micromixers with obstructions. International Journal of Heat and Mass Transfer, 140:483–497, 2019.

S. Ray. A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon), pages 35–39. IEEE, 2019.

S. Z. Razzacki, P. K. Thwar, M. Yang, V. M. Ugaz, and M. A. Burns. Integrated microsystems for controlled drug delivery. Advanced drug delivery reviews, 56(2): 185–198, 2004.

A. Rostami, M. Hajaligol, and S. Wrenn. A biomass pyrolysis sub-model for cfd applications. Fuel, 83(11-12):1519–1525, 2004.

A. Royer, E. Béchet, and C. Geuzaine. Gmsh-fem: an efficient finite element library based on gmsh. In 14th World Congress on Computational Mechanics (WCCM), ECCOMAS Congress 2020. Scipedia, 2021.

V. B. R. Silva and J. Cardoso. Computational fluid dynamics applied to waste-to-energy processes: a hands-on approach. Butterworth-Heinemann, 2020.

D. Simon. Evolutionary optimization algorithms. John Wiley & Sons, 2013.

A. Sobachkin and G. Dumnov. Numerical basis of cad-embedded cfd. In NAFEMS World Congress, volume 2013, pages 1–20, 2013.

T. Sprogies, J. Köhler, and G. Groß. Evaluation of static micromixers for flow-through extraction by emulsification. Chemical Engineering Journal, 135:S199–S202, 2008.

M. Ståhl, B. L. Åslund, and Å. C. Rasmuson. Reaction crystallization kinetics of benzoic acid. AIChE journal, 47(7):1544–1560, 2001.

A. I. Stankiewicz, J. A. Moulijn, et al. Process intensification: transforming chemical engineering. Chemical engineering progress, 96(1):22–34, 2000.

J. Sun, Z. Shi, M. Li, J. Sha, M. Zhong, S. Chen, X. Liu, and S. Jia. Numerical and experimental investigation of a magnetic micromixer under microwires and uniform magnetic field. MAGMA-D-21-02429.

N. T. Tayeb, K. Amar, K. Sofiane, L. Lakhdar, and L. Yahia. Thermal mixing performances of shear-thinning non-newtonian fluids inside two-layer crossing channels micromixer using entropy generation method: Comparative study. Chemical Engineering and Processing-Process Intensification, 156:108096, 2020.

T. Tofteberg, M. Skolimowski, E. Andreassen, and O. Geschke. A novel passive micromixer: lamination in a planar channel system. Microfluidics and Nanofluidics, 8(2):209–215, 2010.

C. Venkateswarlu and S. Jujjavarapu. Stochastic and evolutionary optimization algorithms. In Stochastic Global Optimization Methods and Applications to Chemical, Biochemical, Pharmaceutical and Environmental Processes, pages 87–123. Elsevier, 2020.

H. K. Versteeg and W. Malalasekera. An introduction to computational fluid dynamics: the finite volume method. Pearson education, 2007.

C. Wang, F. Li, Z.-c. Ding, and L. Zhang. Numerical simulation of hypersonic flow around an aerospace plane by parallel rans based cfd. Procedia Engineering, 61, 2013.

L. Wang, D. Liu, X. Wang, and X. Han. Mixing enhancement of novel passive microfluidic mixers with cylindrical grooves. Chemical engineering science, 81: 157–163, 2012.

S.-C. Wang. Artificial neural network. In Interdisciplinary computing in java programming, pages 81–100. Springer, 2003.

H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. Computers in physics, 12(6):620–631, 1998.

Z. Wu and N.-T. Nguyen. Hydrodynamic focusing in microchannels under consideration of diffusive dispersion: theories and experiments. Sensors and Actuators B: Chemical, 107(2):965–974, 2005.

C. Wüstenhagen, K. John, S. Langner, M. Brede, S. Grundmann, and M. Bruschewski. Cfd validation using in-vitro mri velocity data–methods for data matching and cfd error quantification. Computers in Biology and Medicine, 131: 104230, 2021.

B. Xia and D.-W. Sun. Applications of computational fluid dynamics (cfd) in the food industry: a review. Computers and electronics in agriculture, 34(1-3):5–24, 2002.

G. Xia, Y. Li, J. Wang, and Y. Zhai. Numerical and experimental analyses of planar micromixer with gaps and baffles based on field synergy principle. International Communications in Heat and Mass Transfer, 71:188–196, 2016.

L. Zhendong, L. Yangcheng, W. Jiawei, and L. Guangsheng. Mixing characterization and scaling-up analysis of asymmetrical t-shaped micromixer: Experiment and cfd simulation. Chemical Engineering Journal, 181:597–606, 2012.