



---

# Invasion of A Pore by an Immiscible Two-Phase Flow (Water/Oil). Numerical Simulation, Neural Network and Neuro-Fuzzy System

RAKOTOVAO Ndimbinarimalala Philémon<sup>1</sup>, RALIJAONA Mbolahasina Soanandrianina<sup>2</sup>, AMBEONDAHY<sup>3</sup>, RATIARISON Adolphe Andriamanga<sup>4</sup>

<sup>1,2,3,4</sup> Dynamics of the Atmosphere, Climate and Oceans Laboratory (DyACO), University of Antananarivo.

---

**Abstract :** In this work, the invasion of a pore by an immiscible two-phase flow (water/oil) is modeled by a computational fluid dynamics approach, assisted by artificial intelligence models (neural network and neuro-fuzzy system). The flow is that of a fluid occupying the pore (oil) displaced by an injected fluid (water). Numerical simulations were performed to understand the influence of certain parameters and to assess the importance of the input speed. Then, we modeled the simulation data using artificial intelligence methods. The models of the neural networks and the neuro-fuzzy systems obtained largely satisfy the chosen criteria: the Root Mean Square Error and the Nash-Sutcliffe criterion.

**Keywords:** Two-phase flow, immiscible, computational fluid dynamic, neural network and neuro-fuzzy system.

---

## I. Introduction

At the pore scale, the invasion of a fluid in a saturated medium by another fluid immiscible with the latter, is a phenomenon encountered in many contexts, ranging from natural phenomena to industrial processes. The contact surface between these two fluids plays a crucial role in the dynamics of the system. In this zone, the thickness is low and the characteristic gradients of certain physical quantities are much greater than in the rest of the domain. A simplified approach is to assume that the thickness of this zone will be negligible, and that the gradients will be integrated and replaced by a jump condition. The thin zone, reduced by one dimension, is called the interface. [1].

In the present work, on the one hand, we will numerically simulate the immiscible two-phase flow (water/oil) using the OpenFOAM software. The objective is to see the influence of certain parameters on the invasion of water in a pore saturated with oil. On the other hand, we model the data of the numerical simulations obtained by the neural network and the neuro-fuzzy system.

## II. Materials and Methods

### 2.1 Physical model

The simulation is carried out on a two-dimensional structure of 1mm×1mm and the radius of the solid  $r_s=0.2mm$  as shows in the figure. The garnet color represents the solid phase and the gray one is the space where the fluid flows. At first, the medium is saturated with oil. Then, water is injected to move the oil to the outlet.

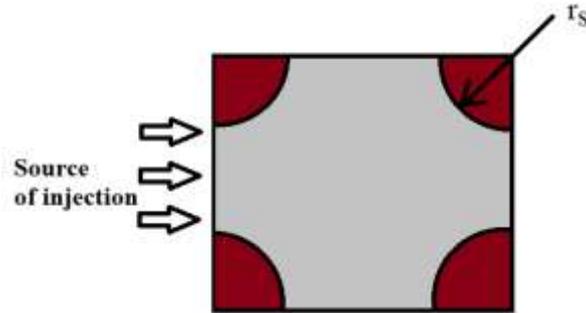


Figure 1 : The representative unit cell.

**2.2 Mathematical model :**

At the pore scale, the flow is governed by the Navier-Stokes equations system of each phase  $i \in \{w, o\}$  :

$$\begin{cases} \frac{\partial \rho_i(x,t)}{\partial t} + \vec{\nabla} \cdot (\rho_i(x,t) \vec{U}_i) = 0 \\ \rho \left( \frac{\partial \vec{U}_i}{\partial t} + \vec{U}_i \otimes \vec{\nabla} U_i \right) = \rho_i \vec{g} - \vec{\nabla} p_i + 2\mu_i \vec{\nabla} \cdot \vec{D} + \vec{F}_{vol} \\ + \text{Initial and boundary conditions} \end{cases} \quad (1)$$

Where  $\rho_i$ ,  $\vec{U}_i$ ,  $p_i$ ,  $\mu_i$  are respectively the density, the velocity, the pressure and the dynamic viscosity of phase  $i$ ,  $\vec{g}$  is the gravity,  $\vec{D}$  is the strain tensor and  $\vec{F}_{vol}$  is the volumetric force.

**2.3 Volume Of Fluid Method (VOF)**

It is a method for monitoring the interface between two fluids. It is based on the filling rate of the cells to determine the temporal evolution of the fluid in the domain. The principle of this method is to introduce a discrete function  $\alpha(x, t)$ , which represents the volume fraction of a fluid in a cell.(figure 2).

The volume fraction is evaluated as follows:

$$\frac{\partial \alpha}{\partial t} + \vec{\nabla} \cdot (\alpha \vec{U}) = 0 \quad (2)$$

0.95	0.4	0	0	0
1	1	0.6	0	0
1	1	1	0.3	0
1	1	1	0.5	0
1	1	1	0.7	0
1	1	1	0.8	0

Figure 2 : Volume fraction value in cells.

$$\begin{cases} \frac{\partial \alpha}{\partial t} + \vec{\nabla} \cdot (\alpha \vec{U}_i) = 0 \\ \frac{\partial \rho_i}{\partial t} + \vec{\nabla} \cdot (\rho_i \vec{U}_i) = 0 \\ \rho \left( \frac{\partial \vec{U}_i}{\partial t} + \vec{U}_i \otimes \vec{\nabla} U_i \right) = \rho_i \vec{g} - \vec{\nabla} p_i + 2\mu_i \vec{\nabla} \cdot \vec{D} + \vec{F}_{vol} \\ + \text{Initial and boundary conditions} \end{cases} \quad (3)$$

Finally, the system of partial differential equations to be solved is composed of the Navier-Stokes equation for each phase and that of the presence rate, represented by the system of equations below:

**2.4 Numerical modeling**

We used OpenFOAM (Open Field Operation And Manipulation) software. This software is primarily a C++ library that was developed for the CFD (Computational Fluid Dynamics) field. This software solves partial differential equations by the finite volume method [2] and mainly oriented towards fluid mechanics. We used the InterFoam solver. This solver solves two-phase flow problems using the VOF method.

In this case study, the input speed was varied. Knowing the value of the speed, the value of the time step can be calculated by the number of current ( $C_o$ ) by the criterion of Current-Friedrich-Lewis (CFL):

$$C_o = \frac{\Delta t |v_\alpha|}{\Delta x} \leq C_{max} \quad (4)$$

Where  $\Delta t$  is the time step size,  $v_\alpha$  the input speed,  $\Delta x$  is the space step and  $C_{max}$  is the maximum current number.

**2.4.1 Mesh of the representative unit cell**

With the blockMesh tool, we generated hexahedral structured meshes. Figure 3 shows the mesh of the representative unit cell.

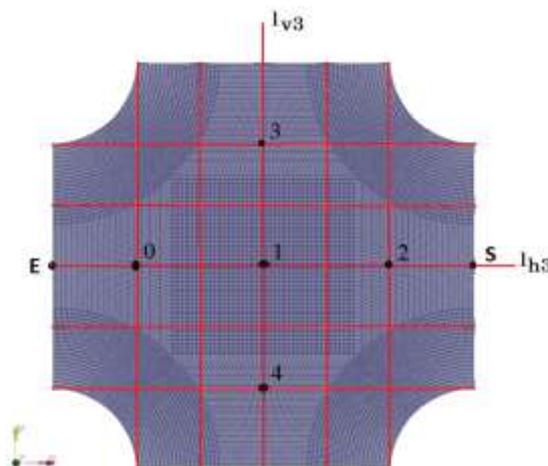


Figure 3 : Schematic of unit cell after meshing

Results will be represented on the horizontal line  $l_{h3}$  and on the vertical line  $l_{v3}$ . The temporal variations will be represented on the five points 0, 1, 2, 3 and 4.

#### 2.4.2 Initial and boundary conditions

We have implemented the following boundary conditions :

Table 1 : Boundary conditions

Parameters	Input	Top and bottom walls	Output
Velocity	Input speed	Non-slip	ZeroGradient
Pressure	ZeroGradient	FixedFluxPressure	fixedvalue

Let us explain the different types of boundary conditions used in this table.

- zeroGradient: the normal speed gradient is equal to 0;
- FixedFluxPressure: adjust the gradient to the pressure so that the flux is that given by the boundary condition on the velocity;
- fixedValue: the value of the field is specified;

#### 2.4.3 Parameters

In our simulation, we used the parameters in Table 2 below. We took petroleum as wetting fluid and water as non-wetting fluid.

Table 2 : Values of physical parameters for the simulation.

Parameters fluids	Characteristics of fluids	
	Water	Oil
Density(kg/m <sup>3</sup> )	1000	500
Kinematic viscosity(m <sup>2</sup> /s)	10	2.10
Surface tension (N/m)	0.018	

### III. Results

#### 3.1 Phase distributions in the unit cell

The interface tracking method (VOF), allows to have the distribution of the phases in the domain (figure 4). The first column corresponds to an entry velocity  $v = 1.5 \times 10^{-2} m/s$ , the second column to  $v = 3 \times 10^{-2} m/s$ . The red color represents the wetting phase (oil), the blue that of the non-wetting phase (water). The color between the two phases represents their interface. At the beginning, the domain is saturated by the wetting fluid (oil). The non-wetting fluid (water) is injected from the left part with the imposed inlet speed. We see in figure 4 that the interface forms a parabolic curve for the two values of the input velocity. This is due to the fact that the velocity on the walls is zero (adhesion condition) and increases progressively as one moves away from the walls. In both cases we see that the water is pushing the oil out. By increasing the input velocity (Figure 4.(a')-(b')), we see that the interface is ahead of the first (figure 4.(a)-(b)). This means that the speed of the water injected at the inlet has an influence on the displacement of the oil in place.

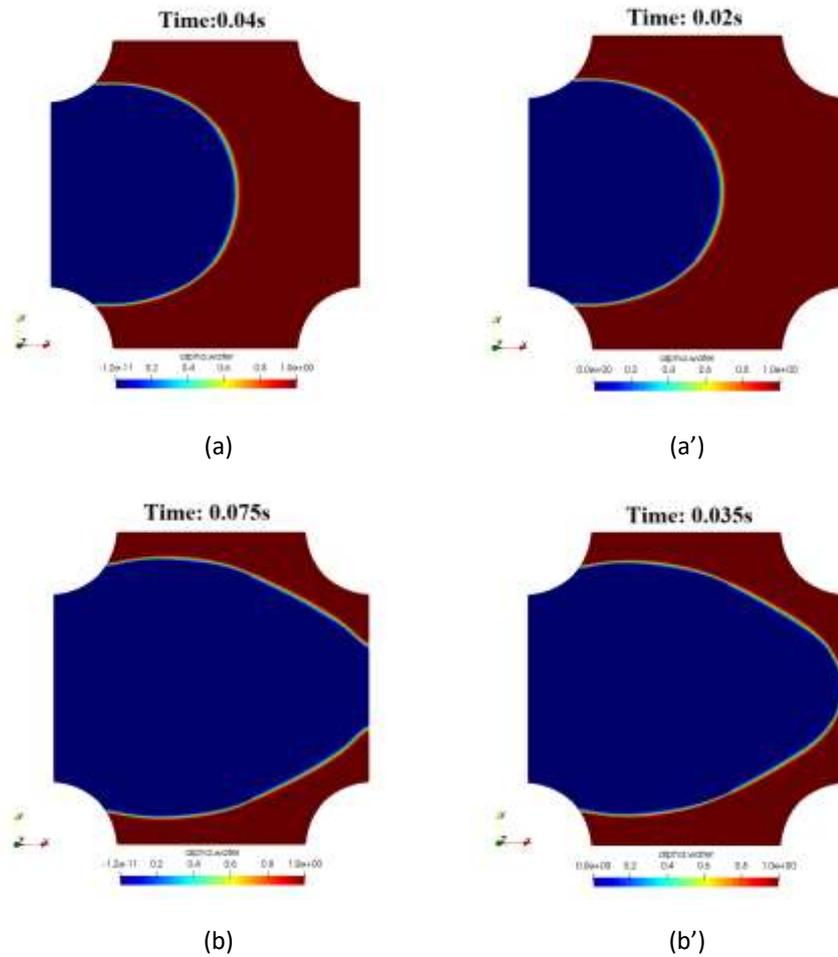


Figure 4 : Phase distributions and interface shape, (a)(b) for  $v = 1.5 \times 10^{-2} m / s$  and (a')(b') for  $v = 3 \times 10^{-2} m / s$ .

**3.2 Spatio-temporal variation of pressure**

Figure 5 represents the temporal variation of the pressure on five (5) points of figure 3 and for the two different values of the entry velocity.

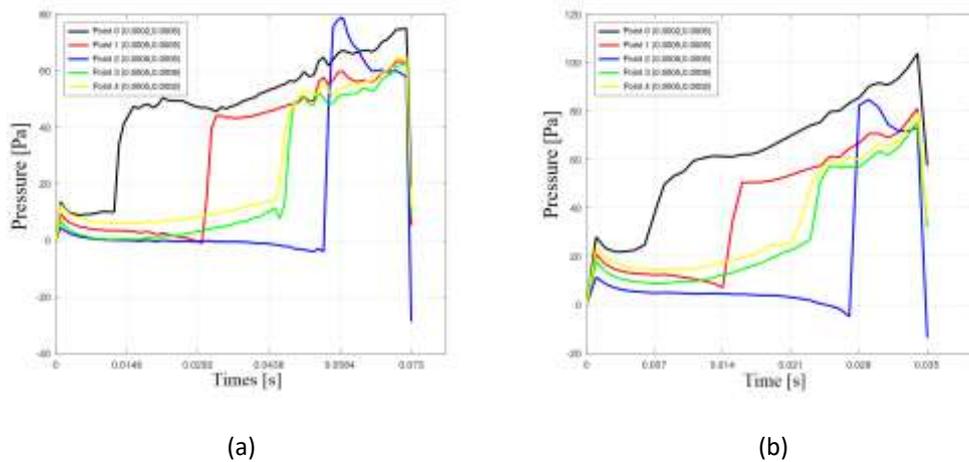


Figure 5 : Temporal variation of pressure (a) for  $v = 1.5 \times 10^{-2} m / s$  and (b) for  $v = 3 \times 10^{-2} m / s$ .

On these two figures, the curves are identical for each point. The sudden increase in pressure observed on the curves represents the interface location. Indeed, the curves clearly show that the interface first passes through point 0, then through point 1, then through points 3 and 4, and finally through point 2. Thus, the pressure within water is low compared to that within oil, allowing the flow to be governed by the pressure gradient.

Figure 6 shows the spatial variation of the pressure. On the corners located at the entrance, a strong pressure is observed due to the fact that the injected fluid exerts a force on the walls. We also see the decrease in pressure from the entry to the exit of the domain. This means that our flow is governed by the pressure gradient.

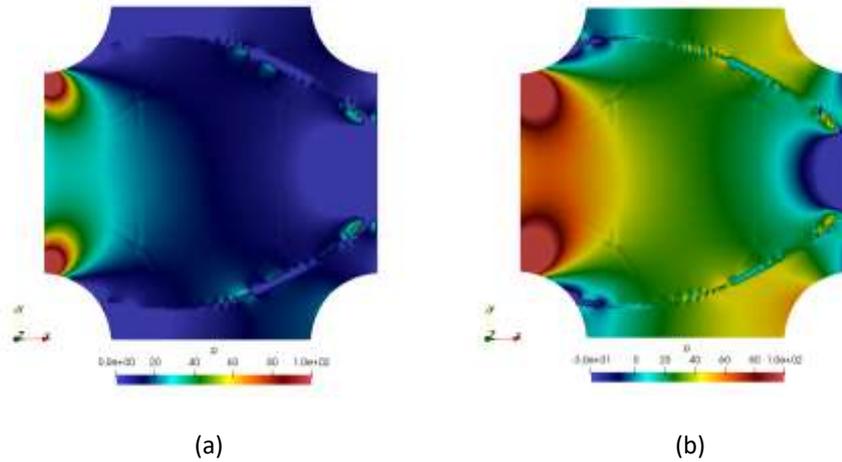


Figure 6 : Spatial variation of pressure (a) for  $v = 1.5 \times 10^{-2} \text{ m/s}$  and (b) for  $v = 3 \times 10^{-2} \text{ m/s}$ .

### 3.3 Velocity field

Water is injected in the domain with a constant speed. Figure 7 shows the flow for a value of the inlet velocity of the order of  $10^{-3}$  (left) et  $10^{-2}$  (right). The direction of flow is from left to right. In both cases, the flow remains laminar. In the figure of the first column where the entry velocity is lower, the flow is influenced by the shape of the elementary cell. However, in the second column, the direction of flow tends to be only in the x direction.

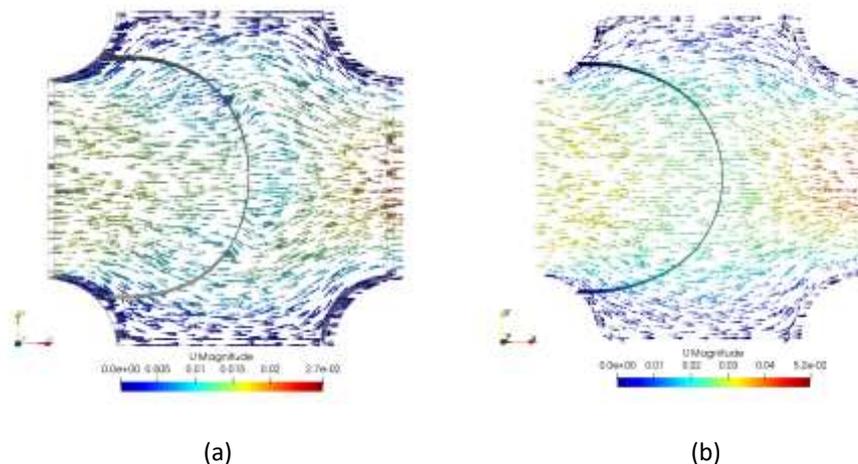


Figure 7 : Velocity vector fields (a) for  $v = 1.5 \times 10^{-2} \text{ m/s}$  and (b) for  $v = 3 \times 10^{-2} \text{ m/s}$ .

### 3.4 Speed intensity

Figure 8 shows the spatial variation of the velocity in the domain. As we mentioned before, we impose the no-slip velocity condition on the upper and lower edges. This is verified for both cases, hence the low values of the velocity on the walls. The speed increases towards the inside of the domain. By observing the

variation of the speed of the entry towards the exit, we see that there is initially a reduction until the center of the field. Then we see a rise towards the output, due to the shrinking at the output.

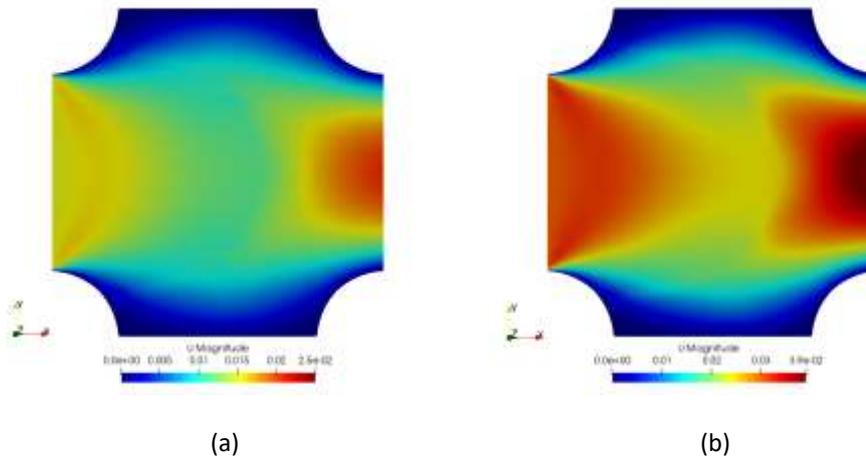


Figure 8 : Spatial variation of velocity (a) for  $v = 1.5 \times 10^{-2} m / s$  and (b) for  $v = 3 \times 10^{-2} m / s$ .

### 3.5 Velocities at points 0 to 4 as a function of time

Figure 9 shows us the time variation of speed for each input speed. In these figures, we see the velocity profiles as a function of time recorded on the five points of figure 3.

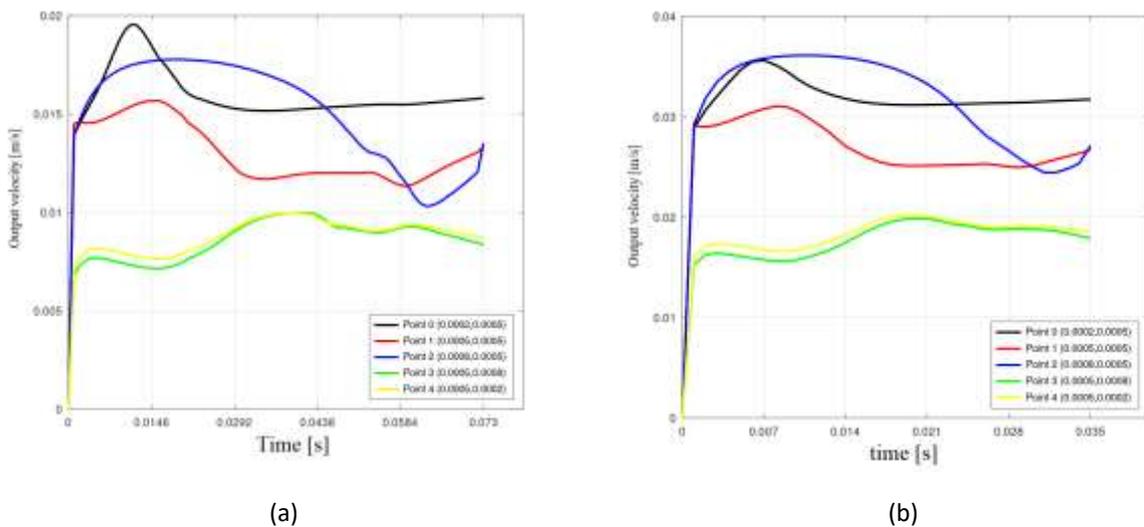


Figure 9 : Comparison of the speed profiles on the five points (a) for  $v = 1.5 \times 10^{-2} m / s$  and (b) for  $v = 3 \times 10^{-2} m / s$ .

We see that the velocity profiles look the same for each point in both cases. But the maximum value reached for each point depends on the input speed. For example for point 0 (the black curve), on the left figure, it reaches a speed  $v_{max} = 0.019 m / s$  after  $t = 0.01 s$  of simulation. And on the figure on the right, it reaches a speed  $v_{max} = 0.035 m / s$  after  $t = 0.6 ms$  of simulation. The black color curves and blue represent the values of the velocities respectively at the entrance and at the exit of the domain. That is why their values are superior compared to others. In the center of the domain, velocity shows an average value compared to the other points. And for points 3 and 4 velocities are lower because they are near the top and bottom walls of the domain.

### 3.6 Influence of the radius of the solid on the speed.

We set the injection speed to  $v = 1.5 \times 10^{-2} \text{ m/s}$  and varies the radius of solid  $r = 0.0002 \text{ m}$ ,  $r = 0.00025 \text{ m}$ ,  $r = 0.00030 \text{ m}$  and  $r = 0.00035 \text{ m}$ . Figure 14 represents the evolution of the velocity as a function of the radius of the solid for the injection velocity  $v = 1.5 \times 10^{-2} \text{ m/s}$ , on the left along the horizontal line  $l_{v3}$  and on the right along the vertical line  $l_{h3}$ . In the figure on the left, the curves are parallel. From the entrance of the domain to the middle, the more the radius of the solid increases, the slower the flow in the medium. On the other hand, at the exit of the domain, the speed increases with the radius. This is due to the decrease of the output width. The decrease in speed as a function of the radius of the solid is also observable.

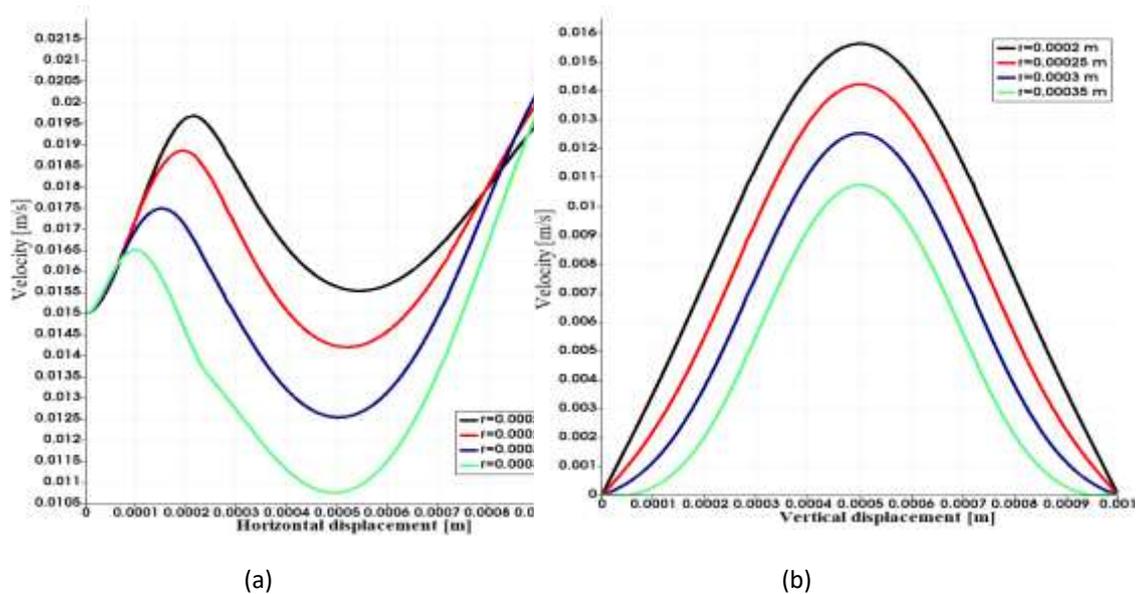


Figure 10: Comparison of velocity profiles as a function of the radius of the solid.

### 3.7 Influence of inlet speed on outlet speed and pressure

To see the influence of the entry speed, six (6) simulations were carried out by varying the entry speed :  $0.010 \text{ m/s}$ ,  $0.015 \text{ m/s}$ ,  $0.020 \text{ m/s}$ ,  $0.025 \text{ m/s}$ ,  $0.030 \text{ m/s}$  and  $0.035 \text{ m/s}$ . Figure 15 represents the variations of velocity and pressure at the exit of the domain as a function of the entry velocity. We see from Figure 15-(a) that the output velocity curve shows two different slopes. Between the input speed  $v = 0.010 \text{ m/s}$  and  $v = 0.015 \text{ m/s}$ , slope is the lower than that of the curve for the speed greater than  $v = 0.015 \text{ m/s}$ . In figure 15-(b), the pressure variation at the outlet has three phases. When the input speed is between  $v = 0.010 \text{ m/s}$  and  $v = 0.015 \text{ m/s}$ , the slope of the pressure variation is medium. When the input speed is between  $v = 0.015 \text{ m/s}$  and  $v = 0.020 \text{ m/s}$ , the slope is high, and for speed greater than  $v = 0.020 \text{ m/s}$ , the slope is low. These phases indicate the proportionality of the variation of other parameters as a function of the input speed. As on the exit speed, it presents two phases and three that on the pressure at the exit.

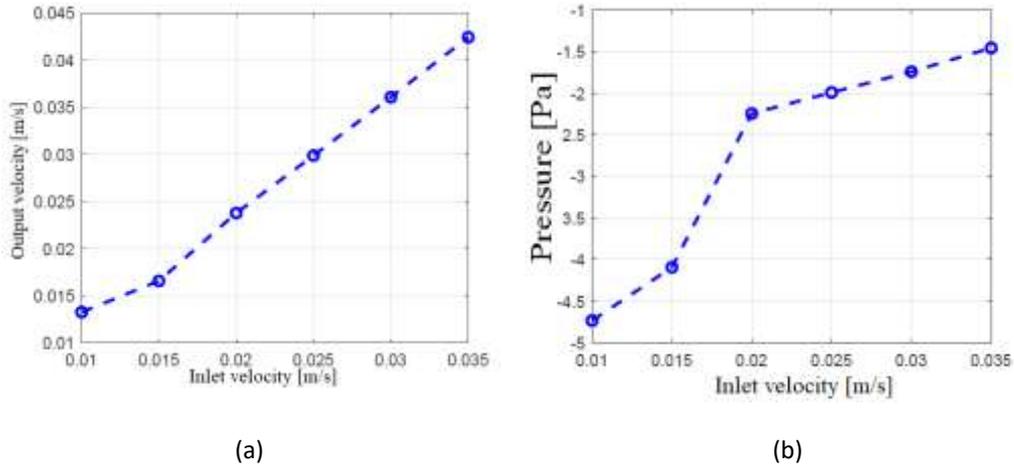


Figure 11 : Representative curve (a)  $V_s=f(V_e)$  and (b)  $P=f(V_e)$ .

**3.8 Modeling by the method of artificial intelligence**

The study proposed in this section consists in modeling the simulation data with the neural network [3] and neuro-fuzzy [4] artificial intelligences. This is to model the output speed as a function of the input speed and the pressure at the end of each simulation on the line  $l_{h3}$ . We collected the data from these six (6) simulations as shown in Table 3.

Table 3 : Simulation data.

Input speed (Ve)	Pressure (P)	Output speed (Vs)
0.010	-4.7323	0.01326
0.015	-4.0921	0.016572
0.020	-2.2452	0.02378
0.025	-1.9922	0.02988
0.030	-1.7427	0.036074
0.035	-1.4571	0.042429

To evaluate the models obtained by the artificial intelligence method, we calculate the root mean square error (RMSE: Root Mean Square Error) and the Nash-Sutcliffe criterion (NASH) given respectively by relations 5 and 6:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (V_{s_{iO}} - V_{s_{iS}})^2}{n}} \tag{5}$$

with,  $V_{s_{iO}}$  et  $V_{s_{iS}}$  are the observed and simulated output velocity, respectively. The model is well optimized if the value of the RMSE is close to zero.

$$NASH = \left( 1 - \frac{\sum_{i=1}^n (V_{s_{iO}} - V_{s_{iS}})^2}{\sum_{i=1}^n (V_{s_{iO}} - \bar{V}_{s_{iO}})^2} \right) \tag{6}$$

with,  $\overline{Vs_{io}}$  the average of the observed exit velocity. The Nash criterion indicates that, if NASH=100%, the adjustment is perfect, on the other hand if NASH<0, the model is bad.

In modeling statistical data, the sample size should be at least thirty (30). In fact, 60% of the data is used for training the model and 40% for testing. In our case, we cannot perform these thirty simulations. We therefore resorted to the interpolation method in order to increase the sample size. The interpolation functions are represented by relations 7 and 8 :

$$P(Ve) : 0.01983 \times Ve^4 - 0.3051 \times Ve^3 + 1.454 \times Ve^2 - 1.564 \times Ve - 4.399 \quad (7)$$

$$Vs(Ve) : 4.778 \times 10^{-5} \times Ve^4 - 0.0008341 \times Ve^3 + 0.005105 \times Ve^2 - 0.006442 \times Ve + 0.0153 \quad (8)$$

For our model, we have two input parameters (Ve and P) and one output parameter (Vs). By the method of multiple regression, the equation obtained is of the form:

$$Vs(Ve, P) = \alpha_1 \times P^2 + \alpha_2 \times Ve \times P + \alpha_3 \times Ve^2 + \alpha_4 \times P + \alpha_5 \times Ve + \beta_0 \quad (9)$$

Equation 9 shows the relationship between the input and output variables that we will use in the artificial intelligence modeling. Now let's see the value of each coefficient and their unit:

- $Vs$  is the output variable;
- $\alpha_1 \times P^2$  is the first input variable, with  $\alpha_1 = 0.0004058 m.Pa^{-2}.s^{-1}$  ;
- $\alpha_2 \times Ve \times P$  is the second input variable, with  $\alpha_2 = 0.05721 Pa^{-1}$  ;
- $\alpha_3 \times Ve^2$  is the third input variable, with  $\alpha_3 = 1.693 s.m^{-1}$  ;
- $\alpha_4 \times P$  is the fourth input variable, with  $\alpha_4 = 0.002482 s.m(Pa.s)^{-1}$  ;
- $\alpha_5 \times Ve$  is the fifth input variable, with  $\alpha_5 = 1.134$  ;
- $\beta_0$  constant which indicates that there are several other variables related to output speed but our study is limited to these five variables.

### 3.8.1 Models Parameter

In Table 4, we present the parameters of the models that we used. As each model has a particular way of processing the data, we will present in this table the number of input and output parameters as well as the type of the model.

Table 4 : RNA and Neuro-Fuzzy parameters used.

Models	Parameters		Kind
	Input	Output	
Neural Network	5	1	Multilayer Perceptron [5]
Neuro-fuzzy	5	1	ANFIS [6]

### 3.8.2 Neural Network Optimization Parameter

The neural network model (figure 16) contains an input layer with five (05) different variables, an output layer and four hidden layers whose two extremes each have ten (10) neurons and those at the center each have eight(8) neurons.

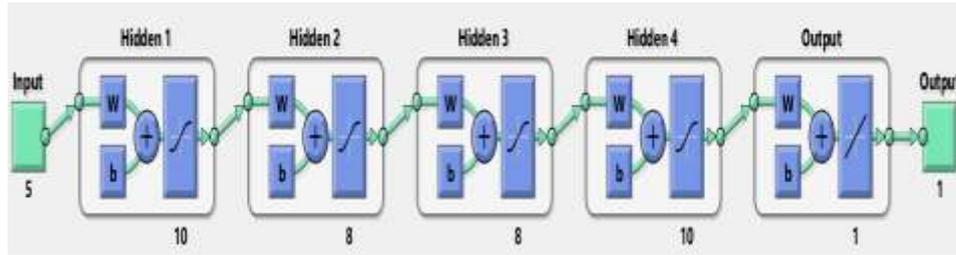


Figure 12 : Neuronal architecture of the model under Matlab.

After six (6) iterations, learning is terminated under the criterion of maximum validation error  $1.6 \times 10^{-10}$ . The best performance for validation was recorded for iteration 6 with an error of  $1.56 \times 10^{-9}$ . Figure 17 represents the performances of the learning and validation phase of the neural network.

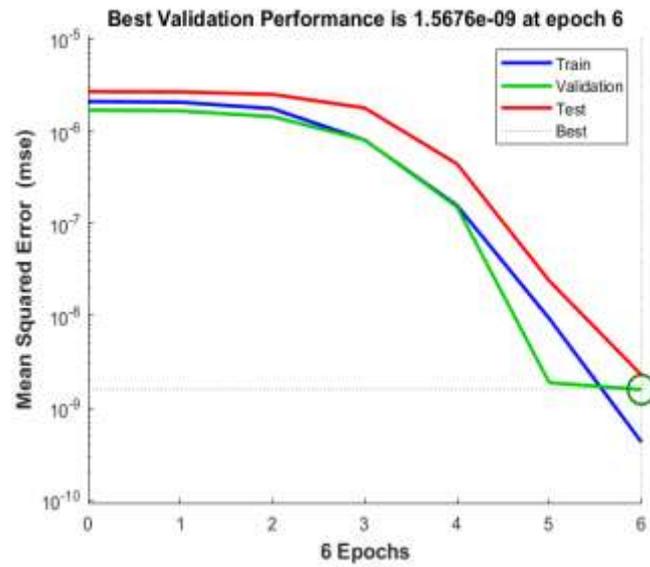


Figure 13 : The performance of the RN learning phase.

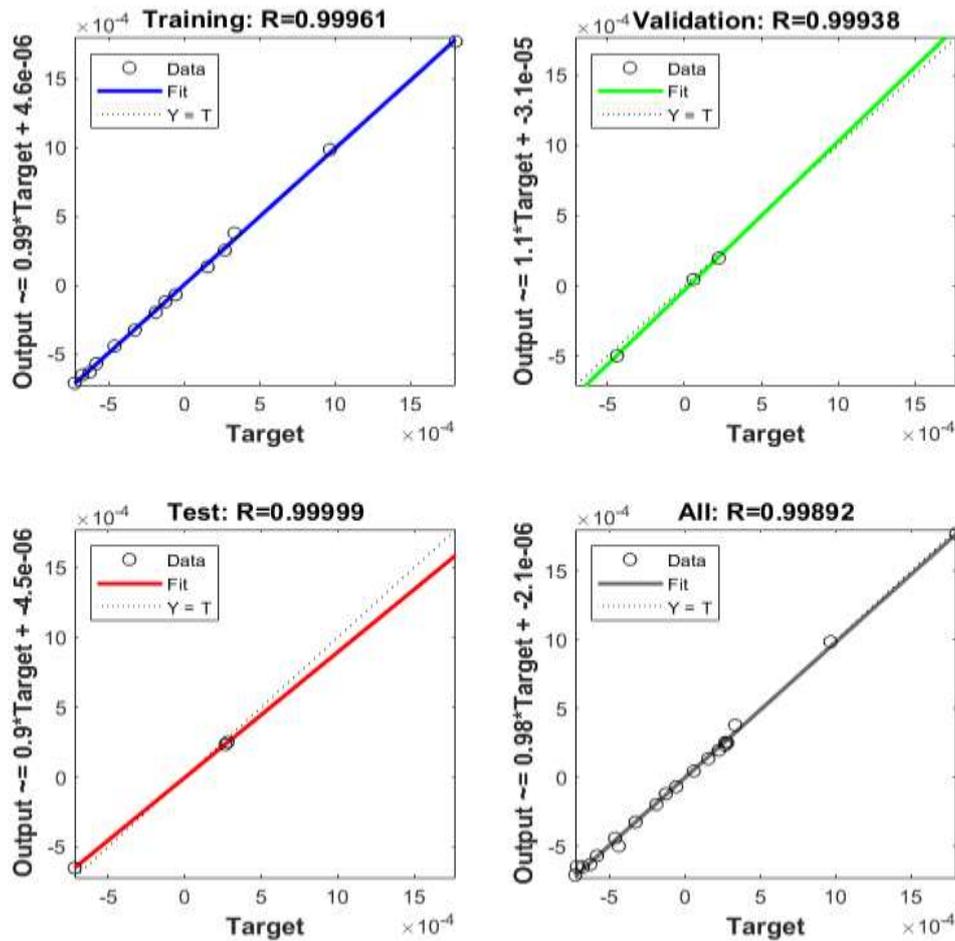


Figure 14 : Correlations between calculated output and desired output.

The calculation of the correlation coefficients between the desired output (Target) and the calculated output (output) makes it possible to validate the selected model. Figure 14 shows these coefficients for each phase, this suggests that we have an excellent model.

### 3.8.3 Neuro-Fuzzy (NF) optimization parameter

We used the hybrid neuro-fuzzy system ANFIS (Adaptive Network Based Inference System). It contains five (5) inputs and one output. Each input is partitioned into three (3) and the output remains constant. We performed supervised gradient descent learning on the first twenty values of the data, that is 66.66%. One hundred (100) iterations were performed during this phase.

Table 5 illustrates the parameter counts of the multilayer perceptron neural network in ANFIS.

Table 5 : Number of neurons per later

Layer number	Layer type	Number of neurons
I	Input	5
II	Fuzzification	$3 \times 5 = 15$
II	Rules	$3^5 = 243$
IV	Standardization	$3^5 = 243$
V	output	1

Let's see the different parametric layer roles:

- The second consists in assigning the input variables to its membership functions, by the "gaussian" function (gaussmf).
- The third concerns the combination of the antecedents of the rules using the méthode "AND"(prod) method.
- Finally the fifth calculates the output by weighted average (wtaver: weighted average).

**3.8.4 IA modeling**

We present the results observed during the learning, testing and validation phases, in order to better analyze the robustness of the models.

Table 6 shows the RMSE values and the NASH criterion of each model on each modeling phase.

Table 6 : RMSE and NASH criterion for each model.

Model	RMS(m/s)			NASH(%)		
	Learning	Test	Validation	Learning	Test	Validation
PMC	$1.09 \times 10^{-4}$	$1.82 \times 10^{-4}$	-	99.99	99.57	-
ANFIS	$1.86 \times 10^{-6}$	$7.46 \times 10^{-5}$	$2.01 \times 10^{-5}$	99.99	99.91	99.99

Figures 15 and 16 respectively represent the results of the model obtained by the multilayer perceptron neural network and Neuro-Flou. The curve in blue line represents the six data of the numerical simulation and that in green color represents the interpolated simulation data. In figure 15, the curve in red in continuous line with a point represents the twenty-five point or 83% of the data used for learning the model and the rest the one in yellow for the test. The model obtained by the neural network is efficient. According to Table 6, the values of RMSE and NASH criterion are satisfactory for each modeling phase.

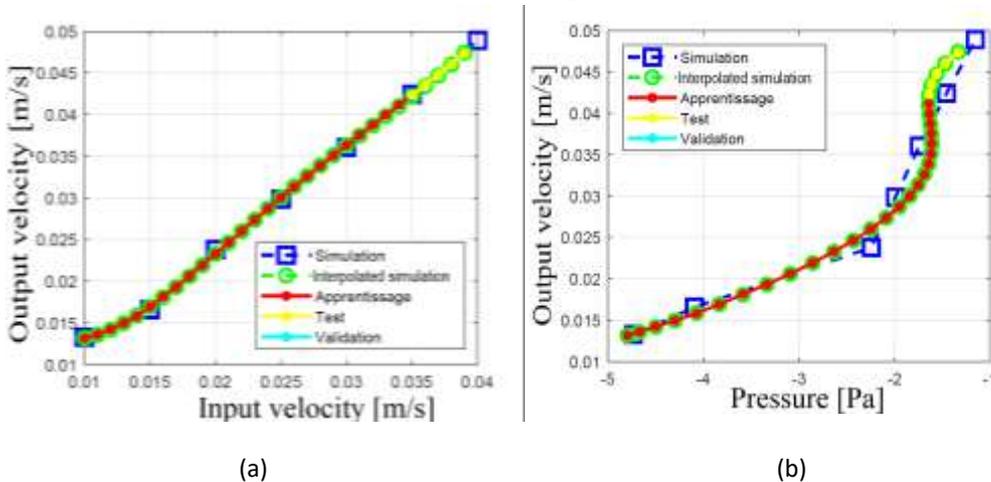


Figure 15 : Models of the neural network (a) Vs=F(Ve) and (b) Vs=f(P)

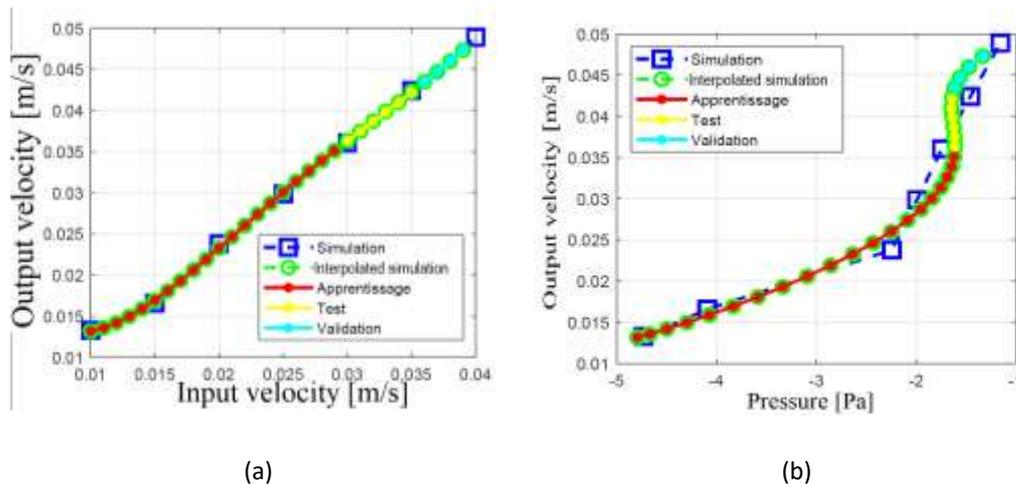


Figure 16 : Neuro-fuzzy models (a)  $V_s = F(V_e)$  and (b)  $V_s = f(P)$

Figure 16, the red curve represents the first twenty points, that is 66% of the data serving as training for the model. The one in yellow represents the test data (20%) and the rest represents the validation data. RMSE values and NASH criteria in table 6 on each phase is satisfactory, we can also say that the model obtained by Neuro-Floue is efficient. If we make a comparison between the two models, we notice the Neuro-Flou gave the best results for each phase. We note a low value of RMSE,  $1.86 \times 10^{-6}$  for the learning phase,  $7.46 \times 10^{-5}$  for the test phase the high value of NASH criterion, 99.91 % for the test phase.

#### IV. Conclusion

In this work, a numerical simulation of a pore invasion was carried out. The pore is initially occupied by oil, water is then injected in order to displace the oil. The Volume of Fluid method was used to follow the interface between the two fluids. The finite volume method was applied in the numerical resolution.

The influence of the entry velocity and the radius of the solid on the distribution of the phases and the evolution of the velocity in the domain was investigated. The pore invasion is an increasing function of the inlet velocity and that the flow is governed by the pressure gradient. Then, we performed two artificial intelligence models (PMC and ANFIS) from our simulation data. The neural network and neuro-fuzzy system models obtained largely satisfy the chosen criteria.

#### V. References

- [1] Fatima-Zahra DAIM : *Étude théorique et numérique de couplages entre écoulements et déformations mécaniques dans l'extraction d'hydrocarbures*, thèse de doctorat, 2004, Université Paris XI Orsay.
- [2] F. Juretic : *Error analysis in finite volume cfd*, Thèse de doctorat, Departement of Mechanical Engineering, Imperial College, 1994.
- [3] Claude TOUZET : *Les réseaux de neurones artificiels. Introduction au connexionnisme*. Cours, exercices et travaux pratiques, 1992.
- [4] Syed Azhar Syed AB. Rahman, Fahmi Asyadi MD Yusof, Mohd Zailani Abu Bakar: *The Method Review of Neuro-Fuzzy Applications in Fault Detection and Diagnosis System*, International Journal of Engineering and Technology IJET-IJENS Vol.10, No,03.
- [5] L. Baghli: *Contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques*. Thèse de doctorat, Université de Henri Poincaré, Nancy, France.1999.
- [6] B. Gorzalczany: *Computational Intelligence Systems and Applications*. Springer, 2002.