Multiphase Modelling of Droplet Dynamics in PEM Fuel Cell Channels

Christian Antetomaso

A Thesis Submitted to

Department of Engineering

Università degli Studi di Napoli "Parthenope"

For the PhD degree in "Energy Science and Engineering"

Cycle:	XXXVIII
Supervisor:	Dr. Bianca Maria Vaglieco, Dr. Simona Silvia Merola
Course Coordinator:	Prof. Laura Vanoli

Summary

The decarbonization of the energy sector and the transition toward renewable or more sustainable sources represent one of the greatest challenges of the 21st century. Despite a growing environmental awareness and the tangible evidence of climate change, the complete abandonment of fossil fuels remains a very long stretch. This is due not only to current technological limitations (since batteries still face critical issues related to energy density, weight, and the environmental impact of production and disposal, in addition to the inadequacy of existing infrastructures), but also to deeply rooted industrial practices and consumer habits. In this context, engineers are called to design compromise solutions that maximize performance within the constraints of available solutions.

Among the emerging technologies, Proton Exchange Membrane Fuel Cells (PEMFCs) stand out as a promising option for both stationary and mobile applications. These electrochemical devices provide power with zero tailpipe emissions, since their only by-products are heat and water. PEMFCs are conceptually simple, silent, and have no moving parts aside from possible flow-control components. Their most widely known application is the Toyota Mirai, a commercial fuel cell vehicle introduced in 2015.

Nevertheless, PEMFCs are still far from reaching full technological maturity. One of the most critical challenges is the management of heat and water, particularly in large systems comprising hundreds of cells per stack, such as the 370-cell stack of the Mirai with a rated peak power of 114 kW. To address this issue, the present work focuses on the study of water droplet removal from the stack channels, combining experimental analysis with extensive numerical simulations.

The research activities can be divided into two main parts. First, an in-depth characterization of the available Gas Diffusion Layer (GDL) was carried out using an image-processing algorithm developed in Vision, a module of the commercial software LabVIEW. Based on the acquired data, a representative GDL geometry was reconstructed, enabling both 2D and 3D simulations aimed at investigating the interaction between droplets of different sizes and the porous structure.

Building on this numerical experience, a new boundary condition was developed initially in ANSYS Fluent and later implemented in OpenFOAM to exploit its greater flexibility, with the purpose of applying a dynamic contact angle for droplets deposited on a surface, starting from the equilibrium of acting forces. The solver with the proposed boundary condition, still under active development at the time of writing, has shown excellent potential in predicting drag forces, shielding effects between droplets and coalescence phenomena.

Contents

Summary i
Contents ii
List of Figures iv
1. Introduction 1
1.1. The path towards carbon neutrality 1
1.2. Proton Exchange Membrane Fuel Cell 3
1.3. Surface wetting 8
1.3.1. Contact angle 8
1.3.2. Wetting on Rough or Structured Surfaces 9
1.3.3. Dynamic Wetting and Contact Angle Theories 10
1.3.4. Contact Angle Hysteresis (CAH) 11
1.4. Numerical modelling of wetting phenomena 11
2. Motivation and scope 16
3. Materials and methods 18
3.1. A realistic GDL 18
3.1.1. Experimental measurements 19
3.1.2. Representative geometry reconstruction 22
3.1.3. VOF Simulations 24
3.1.4. Results and validation 25
3.2. Contact angle modelling in Fluent 30
3.2.1. Static contact angle 32
3.2.2. Quasi-dynamic contact angle UDF 32
3.2.3. Dynamic contact angle UDF 33
3.2.4. Results and validation 34
3.3. Force-Based Dynamic Contact Angle 37
3.3.1. Boundary developement 39
3.3.1.1. readProperties.H 40
3.3.1.2. findDrop.H 40
3.3.1.3. dropG.H and dropVel.H 42
3.3.1.4. printGposition&velocity.H 43
3.3.1.4.1. forcesOnDroplet.H 44
3.3.1.5. printForces.H 46
3.3.1.6. lateralScaleFactor.H 46

3.3.1.7. contactAngleHysteresis.H 48
3.3.1.8. contactAngleCorrelations.H 49
3.3.2. Solver modification 50
3.3.3. Results and validation 51
3.3.3.1. Single drop 58
3.3.3.2. Double drop 62
4. Conclusion and future works 67
5. Brief recap of the other performed activities 69
5.1. Ejector-Based Hydrogen Recirculation 69
5.2. PEMFC 0-D modelling 72
Acknowledgement 75
List of abbreviations 75
List of symbols 76
Bibliography 78
Appendix A - Force-Based Dynamic Contact Angle Pseudocode 84
A.1 - readProperties.H 85
A.2 - findDrop.H 87
A.3 - dropG.H 90
A.4 - dropVel.H 92
A.5 - printGposition&velocity.H 94
A.6 - forcesOnDroplet.H 96
A.7 - printForces.H 100
A.8 - lateralScaleFactor.H 102
A.9 - contactAngleHysteresis.H 105
A.10 - contactAngleCorrelations.H 108
Appendix B - Auxiliary Geometric Functions 110
B.1 - findInterfacePlane() 111
B.2 - findIntersection() 114
B.3 - intersectionArea() 116

List of Figures

Figure 1. Change of GDP, EU population, and GHG emissions with respect to 1990 baseline [2] 1
Figure 2. Change of electricity and heat production mix (a) and employed renewable sources (b) with respect to 1990 [2]. 2
Figure 3. Diagram of a typical PEMFC 4
Figure 4. Typical trend of a PEMFC polarization curve and losses contributions. 6
Figure 5. Typical BP serpentine layout (a) with visualization of droplet and partial meniscus adhering on the channel walls (b) and infrared vision of the water phase in the channels (c)..... 7
Figure 6. Typical equivalent circuit (a) and Nyquist diagram (b) of a PEMFC 7
Figure 7. Difference of contact angle value for hydrophobic (left) and hydrophilic (right) surfaces 9
Figure 8. Schematic of a flat (left), Wenzel (middle) and Cassie-Baxter (right) wettability models..... 9
Figure 9. Visualization of different contact angle at different scale. Adapted from [33]. 10
Figure 10. A droplet deposited on an inclined plane shows two different angles at the front and rear. 11
Figure 11. Comparison between SLIC and PLIC interface reconstruction 13
Figure 12. Visualization of contact angle and its relationship with interface and wall normal 14
Figure 13. Schematic representation of a typical half fuel cell. 18
Figure 14. Top (a) and front (b) view of the interlaced structure of the carbon cloth GDL under investigation; extraction of base geometric parameters (c). 20
Figure 15. Division of membrane active area in 1x1 mm squared ROI. 20
Figure 16. Measured contact angle for various droplet classes. 21
Figure 17. Step performed to model the realistic GDL..... 23
Figure 18. Visualization of the difference between the generated GDL (left: realistic; right: hybrid) 23
Figure 19. Water volume fraction isosurfaces (α = 0.8) for a 300 μm droplet on different GDL configurations: (a) realistic woven GDL, (b) hybrid GDL, and (c) flat reference surface..... 26
Figure 20. Water volume fraction isosurfaces (α = 0.8) for a 600 μm droplet on different GDL configurations: (a) realistic woven GDL, (b) hybrid GDL, and (c) flat reference surface..... 26
Figure 21. Experimental 600 μm droplet motion on the GDL 27
Figure 22. Droplet center of mass displacement (wrt initial deposition) over time..... 28
Figure 23. Droplet mean velocity for different diameter classes and considered GDL surface..... 28
Figure 24. Droplet removal time for different diameter classes and considered GDL surface..... 29
Figure 25. The second droplet was deposited in line with the first one, at the 4th pore..... 30
Figure 26. Water volume fraction isosurfaces (α = 0.8) for a 300-300 μm droplet pair on different GDL configurations: (a) realistic woven GDL, (b) hybrid GDL, and (c) flat reference surface..... 30
Figure 27. (a) Visualization of droplet deposited on channel and (b) morphological parameters (CA and H over B) for several droplet spanning over different chords classes. 31
Figure 28. Droplet is deposited as a spherical cap in the domain..... 32
Figure 29. Validation of the dynamic contact angle model and comparison with the proposed approaches..... 35
Figure 30. Comparison of 0.5 isosurface for static contact angle (grey), quasi-dynamic contact angle (red) and dynamic contact angle (blue); close up on the receding front to highlight the difference in the applied angle at the interface..... 35
Figure 31. Comparison of centre of mass position for different contact angle modelling strategies and three droplet classes. 36
Figure 32. Flowchart of the proposed Force Based Dynamic Contact Angle..... 38
Figure 33. Visualization of the implemented Flood-fill algorithm. After the initial assignment of unique global indices to all cells (a), only cells with a secondary-phase volume fraction above the

defined threshold retain their vofID field values (b). When the droplet spans multiple processors (c), the algorithm compares neighbouring cells across the interface and shares the lowest index value (d). During the subsequent iterations, this updated index is propagated through all connected cells (e) until full consistency is achieved (f). 42
Figure 34. To avoid the presence of cusps at the contact line (left), a linear scalar factor is implemented to smooth out the trend from advancing-receding angles (right) 47
Figure 35. The same 0.02 mm^3 droplet was deposited on surface with increasing hydrophobic behaviour 51
Figure 36. A B600 and B400 droplet were deposited at 1 mm distance (left), and their centers of masses distance was monitored (right); simulation predictions align quite well with experimental visualization..... 52
Figure 37. Drag coefficient over Reynolds number, adapted from Schlichting and Gersten [87] 53
Figure 38. Comparison between boundary evaluated drag force and Eq. 45 54
Figure 39. Initial drop has a spherical shape (left); once it is deposited on a surface with given contact angle, it reaches a stable condition as a spherical cap (right) 56
Figure 40. Forces acting on a droplet deposited over an inclined plane 56
Figure 41. Experimental data from Podgorsky [88] and Le Grand et al. [89], numerical simulations and analytical trend from Eq. 51 58
Figure 42. Center of mass position (top) and longitudinal drag force (bottom) for droplets B400, B500, and B600 on a hydrophilic surface under a 15 m/s airflow..... 59
Figure 43. Center of mass position (top) and longitudinal drag force (bottom) for droplets B400eq, B500eq, and B600eq on a hydrophobic surface under a 15 m/s airflow 59
Figure 44. Front and top view of isosurfaces ($\alpha = 0.5$) for a B400 (blue), B500 (red) and B600 (yellow) droplet deposited on an hydrophilic surface at different timestamp 60
Figure 45. Front and top view of isosurfaces ($\alpha = 0.5$) for a B400eq (blue), B500eq (red) and B600eq (yellow) droplet deposited on an hydrophobic surface at different timestamp 61
Figure 46. The parametric study investigated the reciprocal distance (right), the droplet order and shielding effect (middle) and offset (left) 62
Figure 47. Longitudinal drag force for B600-B600 droplets pair on a hydrophilic surface under a 15 m/s airflow 63
Figure 48. Longitudinal drag force for B600-B600 droplets pair on a hydrophobic surface under a 15 m/s airflow 63
Figure 49. Longitudinal drag force for B400-B600 (top) and B600-B400 (bottom) droplets pair on a hydrophilic surface under a 15 m/s airflow 64
Figure 50. Longitudinal drag force for B400-B600 (top) and B500-B600 (bottom) droplets pair on a hydrophilic surface under a 15 m/s airflow and starting with 150 microns offset 65
Figure 51. Detail of the mesh in the primary/secondary mixing region 70
Figure 52. Mach number along the symmetry axis for different meshes..... 71
Figure 53. Comparison with experimental data for primary massflow over different pressure levels 71
Figure 54. Overall dimensions of the proposed geometry (a), CAD model of the ejector (b), and 3D-printed prototype (c). 72

1. Introduction

1.1. The path towards carbon neutrality

The so-called “Fit for 55 Package” [1] is an ambitious set of proposals by the European Union aimed at achieving a primary target and an intermediate milestone: a 55% reduction of greenhouse gas emissions compared to 1990 levels, followed by complete carbon neutrality by 2050. Greenhouse gases (GHG) refer to all gases that directly contribute to global warming by trapping heat within the Earth’s atmosphere. According to data reported by the European Commission [2] and illustrated in Figure 1, over the past thirty years, despite a slight increase in population and an approximately 60% growth in Gross Domestic Product (GDP), per capita and per GDP unit GHG emissions have followed a clear downward trend. This demonstrates that EU strategies are not mere rhetoric, but a viable pathway for developed countries, which are expected to lead the way in promoting sustainable practices.

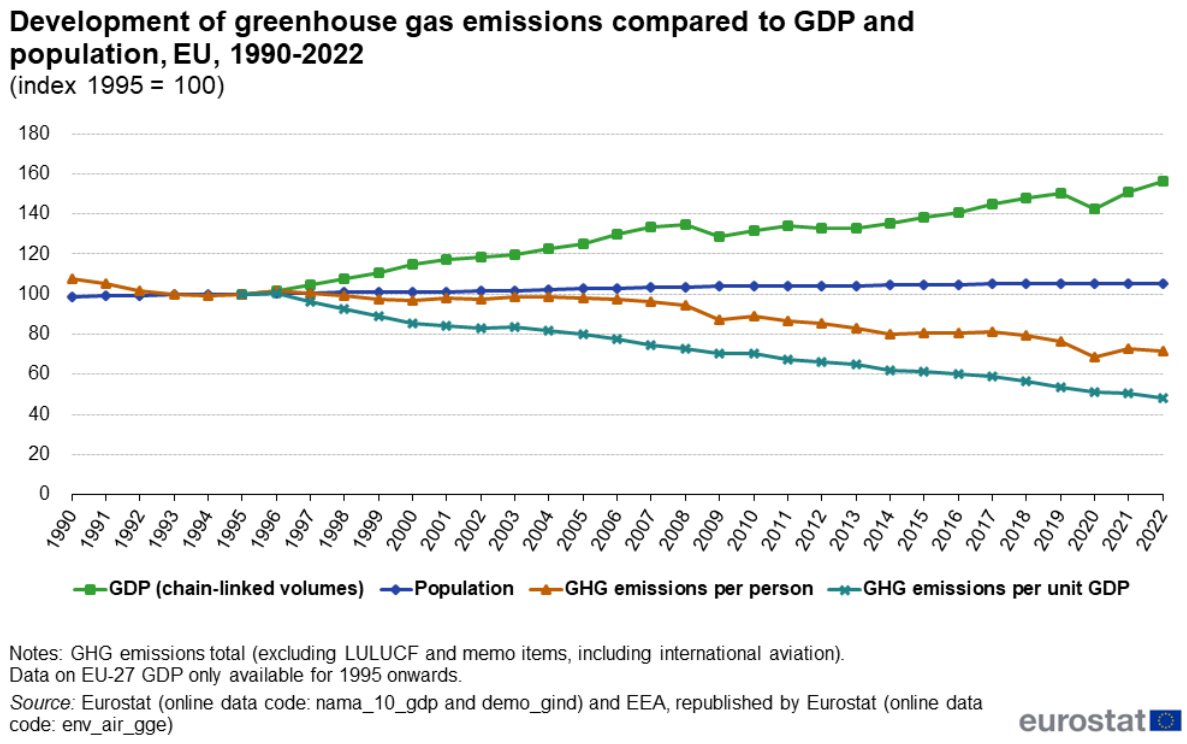
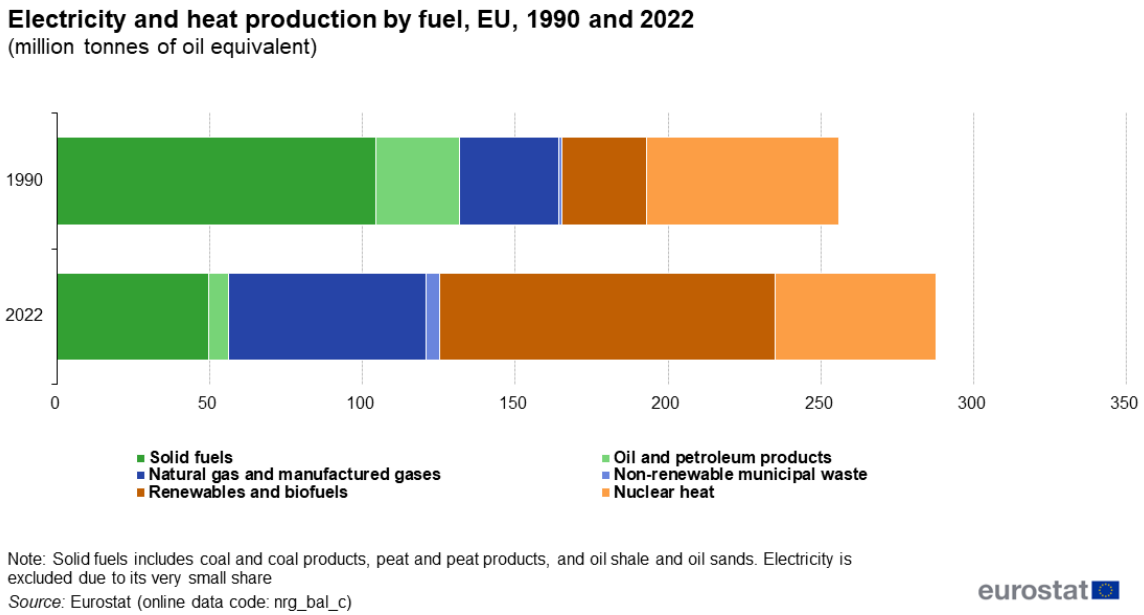


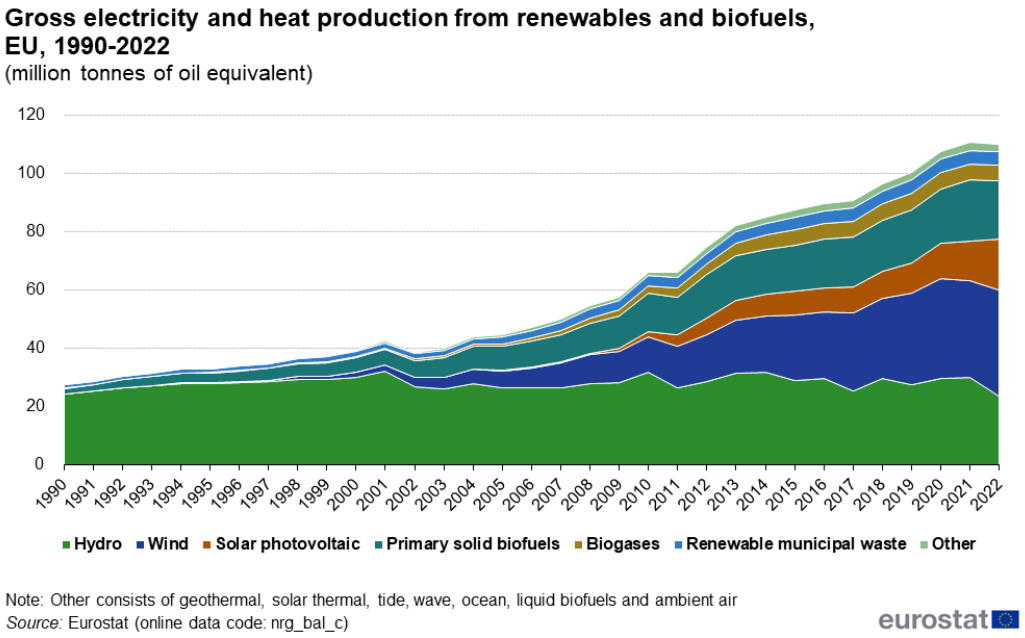
Figure 1. Change of GDP, EU population, and GHG emissions with respect to 1990 baseline [2]

Nevertheless, the challenge remains extremely complex, and solutions are rarely straightforward. While full electrification represents the clear direction for the global energy transition, it remains uncertain whether the current energy infrastructure can support, for instance, a complete conversion of the automotive fleet to electric vehicles [3]. Moreover, even if full electrification were achieved, GHG emissions would not necessarily be eliminated but

merely geographically shifted unless the overall energy mix is also decarbonized. Figure 2 highlights how the energy mix has gradually shifted toward renewable or lower-carbon sources, such as from coal to natural gas, yet a significant share of energy demand is still met by fossil fuels.



(a)



(b)

Figure 2. Change of electricity and heat production mix (a) and employed renewable sources (b) with respect to 1990 [2].

Another key challenge is the inherent variability of renewable technologies such as wind and solar, whose generation patterns can fluctuate not only seasonally but also within a single day. Electricity production from these sources is strongly dependent on weather conditions and does not always align with peak demand periods. This mismatch necessitates the availability of storage systems or flexibility solutions capable of absorbing excess energy during high generation periods and releasing it when demand exceeds supply. Electrochemical batteries provide an effective short-term balancing solution on hourly or daily scales, but their capacity for seasonal balancing remains limited.

In this context, the Power-to-X [4] concept has been gaining traction as a complementary strategy to conventional electricity storage. The concept involves using excess or difficult-to-manage renewable electricity to generate secondary energy carriers that are more easily storable, transportable, and usable in sectors different from the original point of production, such as e-fuels [5] or green hydrogen [6]. Through electrolysis and subsequent chemical synthesis, renewable energy can be converted into high-density molecules capable of long-term storage, extending the benefits of decarbonization to hard-to-electrify sectors such as heavy industry, maritime, and aviation. This strategy allows a better exploitation of production peaks, albeit at the cost of reduced overall cycle efficiency. For example, the final cost of green hydrogen strongly depends on the efficiency and expense of water electrolysis, which remains energy-intensive and relatively costly compared to conventional methane steam reforming [7], which in turn produces CO_2 and results in the so-called grey hydrogen. This illustrates a clear trade-off between environmental sustainability and energy efficiency, indicating that Power-to-X is an essential tool for ensuring flexibility and stability in future renewable energy systems, provided that the input electricity is abundant, low-cost, and, ultimately, green. This raises the question of how green hydrogen can be effectively utilized in energy systems.

1.2. Proton Exchange Membrane Fuel Cell

Proton Exchange Membrane Fuel Cells (PEMFCs) are a highly promising technology for the direct generation of electricity from hydrogen, owing to their high-power density, modularity, and absence of local emissions. These cells operate at relatively low temperatures, typically between 60 and 90 °C, and exhibit rapid start-up times, making them particularly suitable for automotive applications where quick transition to operational conditions is essential. A PEMFC is an electrochemical system that converts the chemical energy of hydrogen into electrical energy through redox reactions [8]. The overall reaction is:

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O \quad (1)$$

The corresponding electrode half-reactions are:

$$H_2 \rightarrow 2H^+ + 2e^- \quad \text{(anode)} \tag{2}$$

$$\frac{1}{2}O_2 + 2H^+ + 2e^- \rightarrow H_2O \quad \text{(cathode)} \tag{3}$$

Protons migrate through the membrane to the cathode, while electrons flow through the external circuit, providing electrical energy to the load. The cell power is determined by the current and cell potential, modulated by activation, ohmic, and mass transport losses [8]. Figure 3 illustrates a conceptual diagram of a typical PEMFC, highlighting that water and heat are the only by-products, while unreacted fuel is often recirculated through pumps, ejectors, or both to improve system efficiency [9,10].

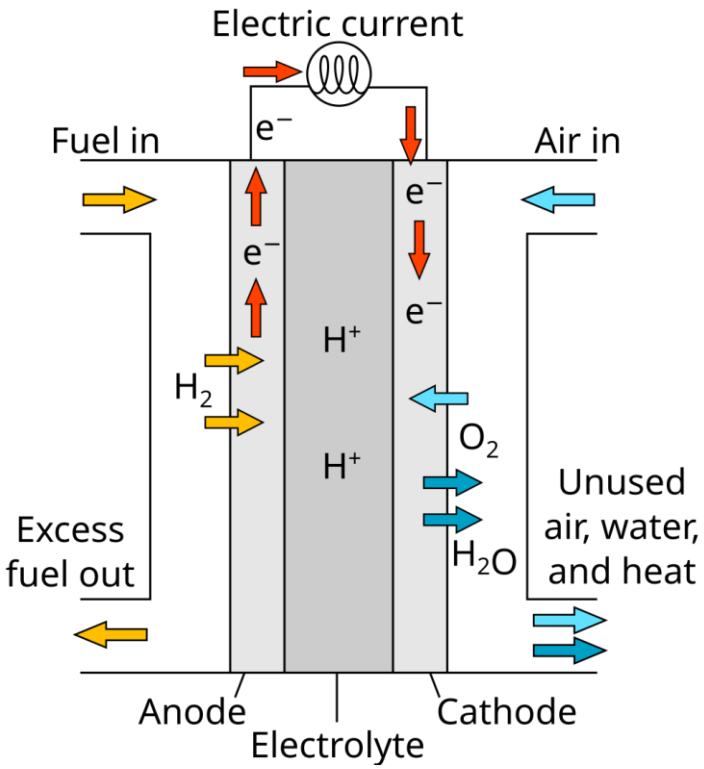


Figure 3. Diagram of a typical PEMFC

Activation losses arise from the kinetic barriers of the electrochemical reactions at the electrodes and represent the initial energy required to start the reactions. A simplified representation is given by the Tafel equation:

$$\eta_{act} = a + b\log(i) \tag{4}$$

where i is the current density and b is the Tafel slope. This correlation is empirical and must be calibrated for each experimental setup. Activation losses are most significant at low current densities during cell start-up.

Ohmic losses result from the internal resistance of the cell, including the membrane, bipolar plates, and electrical contacts. They are described by Ohm's law:

$$\eta_{ohm} = i \cdot R_{tot} \quad (5)$$

Where R_{tot} is the total cell resistance. Ohmic losses increase linearly with current density and depend on the proton conductivity of the membrane and the electronic conductivity of the components.

Concentration losses are associated with the limited diffusion of reactants to the catalytic surfaces and become significant at high current densities. They can be expressed as:

$$\eta_{conc} = \frac{RT}{nF} \ln\left(\frac{i_{lim}}{i_{lim} - i}\right) \quad (6)$$

where R is the gas constant, T the temperature, n the number of electrons transferred, F Faraday's constant, and i_{lim} the limiting current at which concentration losses become relevant. Figure 4 shows a typical polarization curve of a PEMFC, highlighting the three regions corresponding to activation, ohmic, and concentration losses.

While the polarization curve characterizes the intrinsic electrochemical behaviour of a single cell, the practical deployment of PEMFC systems depends on broader technological and durability considerations. Over the past two decades, Proton Exchange Membrane Fuel Cells (PEMFCs) have progressed from laboratory-scale prototypes to commercially deployed systems in both stationary and mobility sectors [11]. Their high mass power density and rapid transient response make them particularly attractive for transportation applications, including passenger vehicles, buses, and heavy-duty trucks, where fast refuelling and long driving range are strategic advantages over battery-based solutions [12]. In stationary applications, PEMFC systems are employed for backup power, distributed generation, and combined heat and power units, where reliability and modular scalability are key requirements [13].

Despite these advancements, large-scale commercialization still faces substantial technical and economic barriers. System cost remains strongly influenced by platinum-group catalyst loading, membrane durability, and balance-of-plant components. From a durability perspective, performance degradation arises from catalyst dissolution, membrane chemical aging, carbon corrosion, and mechanical stress induced by hydration cycles [14,15]. Achieving long operational lifetimes under dynamic load conditions remains a central research challenge, particularly for automotive duty cycles.

Among the various limiting phenomena, water management plays a pivotal role because it directly couples electrochemical performance with multiphase transport. This coupling is particularly critical in the cathode region, where liquid water generation, capillary transport through the porous media, and phase change phenomena determine the local reactant availability. Proper hydration of the membrane and removal of produced water are essential: a dry membrane reduces proton conductivity, whereas excess water can lead to flooding, obstructing channels and limiting reactants transport. Water produced at the cathode Catalyst Layer (CL) passes through the Gas Diffusion Layer (GDL) and microstructure pores before condensing in the channels of the Bipolar Plates (BP), where water vapour can condense and influence gas flow distribution and electrochemical efficiency [16,17,18].

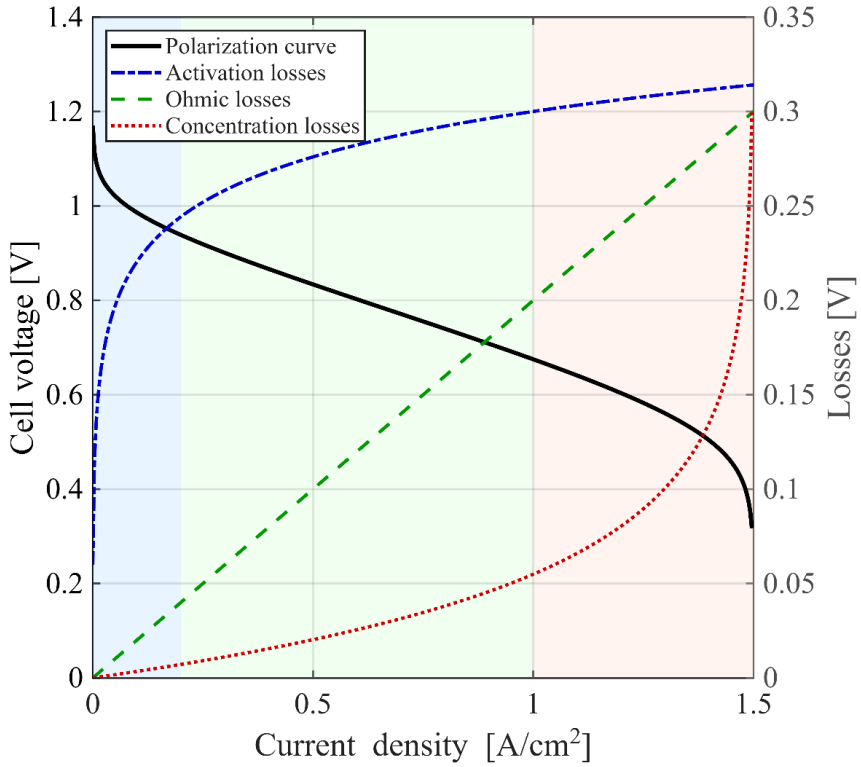


Figure 4. Typical trend of a PEMFC polarization curve and losses contributions.

Advanced experimental techniques, including neutron radiography, X-ray tomography, and optical imaging in transparent cells, have enabled real-time visualization of droplet formation and motion, supporting the validation of multiphase water transport models [19,20]. Numerical models solve coupled multiphase transport, electrochemical reactions, and thermal balances to estimate liquid saturation and humidity gradients under varying operating conditions [18,21].

From a diagnostic perspective, direct in situ measurement of water content is challenging, making non-intrusive techniques such as Electrochemical Impedance Spectroscopy (EIS) invaluable. Figure 5 shows a serpentine layout for the channels of a PEMFC. The visualizations on the right (b–c) illustrate the complexity and the variety of shapes that a water droplet can assume within such channels. These morphologies depend not only on the specific wettability

of the surface but also on the number of walls with which the liquid phase is in contact. Visualizations of this kind are extremely difficult to obtain during actual cell operation.

EIS decomposes the cell response into resistive and capacitive contributions associated with ohmic resistance, charge transfer, and mass transport. Changes in water saturation directly affect the impedance spectra, thereby enabling inference of the cell hydration state [22,23]. Figure 6 presents a typical equivalent circuit and the corresponding Nyquist plot. At high frequencies, the impedance is mainly governed by R_1, the high-frequency resistance, while at lower frequencies the capacitive elements are fully charged, showing the combined resistance $R_1 + R_2$.

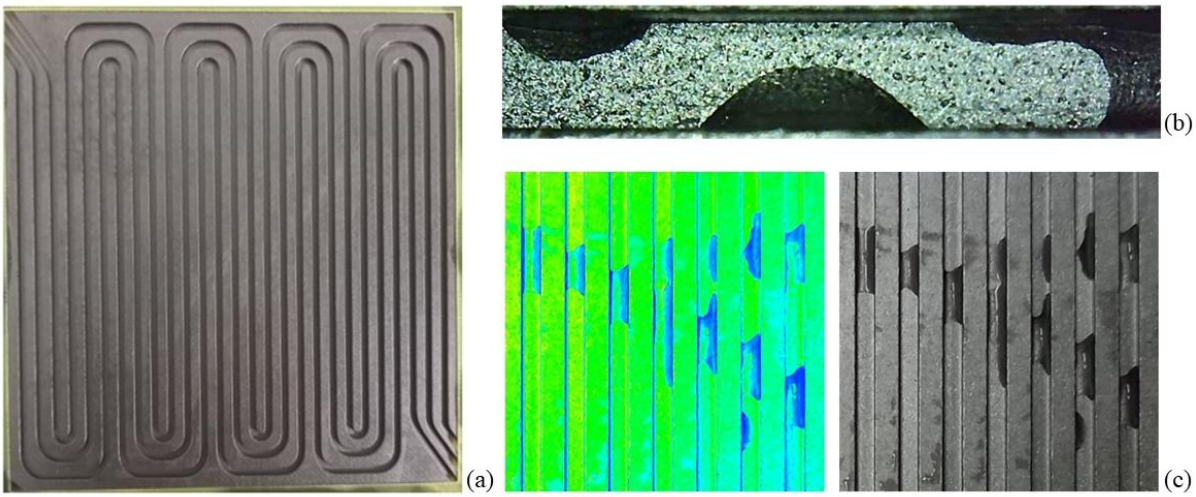


Figure 5. Typical BP serpentine layout (a) with visualization of droplet and partial meniscus adhering on the channel walls (b) and infrared vision of the water phase in the channels (c)

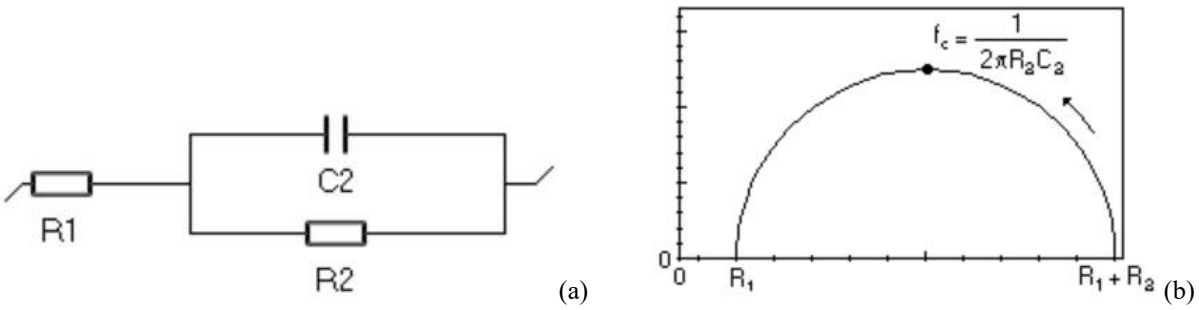


Figure 6. Typical equivalent circuit (a) and Nyquist diagram (b) of a PEMFC

For these reasons, improving the understanding of liquid water dynamics within PEMFC channels remains a fundamental research objective. The present work addresses this issue by developing and validating numerical tools aimed at resolving droplet behaviour under controlled operating conditions.

1.3. Surface wetting

Despite its apparent simplicity, wetting is a multiscale and multiphysics process. It involves molecular interactions at the three-phase contact line, capillary and viscous forces at the mesoscale, and possibly turbulence and inertia at the macroscale. The simultaneous interaction of these mechanisms across several orders of magnitude in space and time makes wetting one of the most complex problems in fluid dynamics. This phenomenon plays a key role in numerous industrial sectors [24], including printing [25], coating [26], water management in vehicles [27] and in fuel cells [28].

1.3.1. Contact angle

The wettability of a surface by a liquid is described by the contact angle θ, defined as the angle formed between the solid surface and the tangent to the liquid interface at the contact point. This angle depends on both the liquid properties and the surface morphology.

At this stage, it is important to distinguish between:

- Static contact angle (θ_{sta}): the contact angle of a droplet at equilibrium.
- Dynamic contact angle (θ_{dyn}): the contact angle of a moving droplet, such as a droplet sliding on a car windshield.

Dynamic contact angles can further be classified based on motion direction: advancing angle (θ_{adv}) and receding angle (θ_{rec}).

Under static conditions, the contact angle can be described by Young's equation:

$$\cos(\theta_{sta}) = \frac{\sigma_{SV} - \sigma_{SL}}{\sigma_{LV}} \quad (7)$$

where σ_{SV}, σ_{SL}, and σ_{LV} are the interfacial energies of the solid-vapor, solid-liquid, and liquid-vapor interfaces, respectively.

A surface is considered hydrophobic when its static contact angle exceeds 90°, with the droplet forming a more spherical shape. Conversely, if the liquid spreads and assumes the shape of a spherical cap, the surface is defined as hydrophilic, with contact angles below 90° [29].

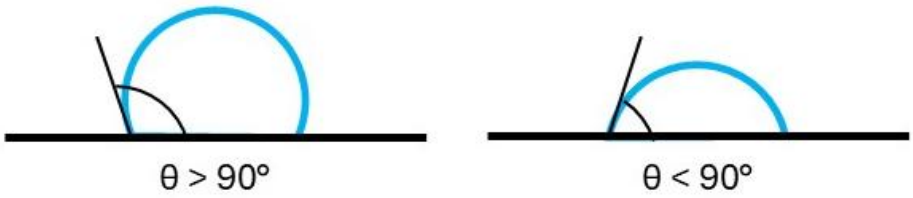


Figure 7. Difference of contact angle value for hydrophobic (left) and hydrophilic (right) surfaces

1.3.2. Wetting on Rough or Structured Surfaces

On rough or structured surfaces, the apparent contact angle can differ from the value predicted by Young's equation. Two main models describe this behavior:

- Wenzel model [30]: applicable to rough surfaces where the liquid penetrates the surface pores. The apparent contact angle θ_W is given by:

$$cos\ (\theta_W) = r \cdot cos(\theta) \tag{8}$$

  where θ is the contact angle on a smooth surface and r is the roughness factor, defined as the ratio of the actual wetted surface area to the projected (geometric) surface area, with $r \geq 1$.

- Cassie-Baxter model [31]: describes surfaces where the liquid does not penetrate the pores but remains on top, with air pockets present between the liquid and the solid. The apparent contact angle θ_{CB} is given by:

$$\cos(\theta_{CB}) = f_s \cdot \cos(\theta) - (1 - f_s) \tag{9}$$

  where f_s is the fraction of solid surface in contact with the liquid, thus $f_s < 1$.

Figure 8 illustrates the differences between these two models.

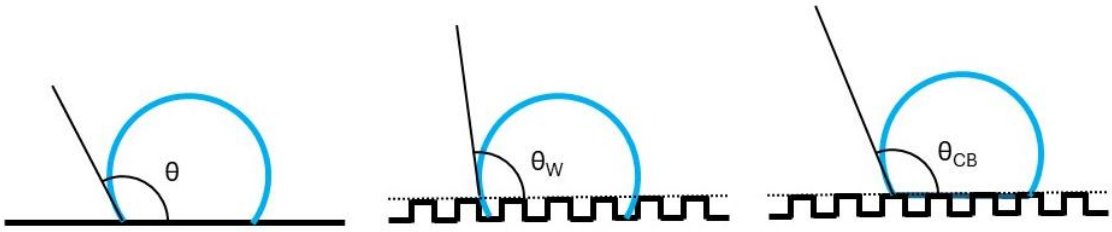


Figure 8. Schematic of a flat (left), Wenzel (middle) and Cassie-Baxter (right) wettability models

1.3.3. Dynamic Wetting and Contact Angle Theories

Contact angle measurement can be performed using various methodologies reported in the literature [32]. But it is important to note that, under real conditions, droplets rarely remain static. Moreover, measurements often reflect a macroscopic apparent angle rather than the microscopic contact angle θ_m that satisfies Eq. (7) [33] (see Figure 9).

Microscopic angles are governed by intermolecular forces, while macroscopic variations arise from viscous effects at the mesoscopic scale. Several models have been developed to describe dynamic contact angles:

- hydrodynamic theory focuses on viscous dissipation in the corner flow near the contact line. It connects the microscopic contact angle θ_m to an intermediate-scale contact angle θ_d through relationships of the form [34]:

$$\theta_d^3 = \theta_m^3 + Ca \ln\left(\frac{\epsilon}{\epsilon_m}\right) \quad (10)$$

  where ϵ and ϵ_m are appropriately chosen macroscopic and microscopic length scales, respectively. This model has been refined by Dussan [35] and Cox [36], but experimental studies near the contact line (<10 μm) show discrepancies, suggesting that θ_m may also depend on the contact line velocity [37].

- molecular-kinetic theory [38] describe the motion of the contact line in terms of the statistical dynamics of molecules near the three-phase line. While parametrization can be challenging, predictions are often comparable to those of hydrodynamic models.

- interface formation model [39] uses non-equilibrium thermodynamics to account for interfacial mass densities. It describes energy dissipation associated with the creation and destruction of interfaces as the contact line moves, in addition to standard hydrodynamic dissipation. This approach links the microscopic dynamic contact angle directly to the flow, enabling simulations at high Capillary numbers, such as in high-speed coating. At low Capillary and Reynolds numbers, the model reduces to a form similar to the hydrodynamic theory.

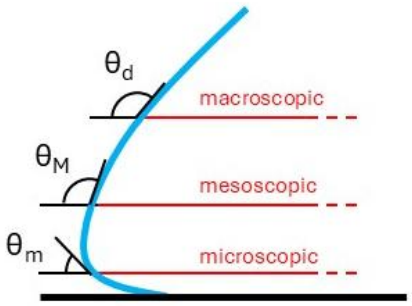


Figure 9. Visualization of different contact angle at different scale. Adapted from [33].

1.3.4. Contact Angle Hysteresis (CAH)

As already described, the contact angle is not unique but depends on the forces acting on the droplet and on its motion history. Contact Angle Hysteresis (CAH) is evaluated as the difference between the advancing angle θ_{adv} and the receding angle θ_{rec} of a droplet on a surface. This phenomenon originates from surface roughness, chemical heterogeneity, and pinning of the contact line, which prevent the droplet from assuming a single equilibrium shape. As a result, the droplet's shape and position depend on its wetting history, leading to metastable configurations [40]. A high CAH indicates strong resistance to droplet motion, whereas low CAH corresponds to more mobile, slippery surfaces [41,42,43].

To reproduce hysteresis in numerical simulations, empirical models such as Kistler's model [44] are often used. These models incorporate θ_{adv} and θ_{rec} as equilibrium limits, enabling the simulation of pinning, depinning, intermittent motion, and oscillatory behaviour of droplets on complex surfaces.

To link the present discussion to what we already saw in chapter 1.2, in PEMFC surface wettability and CAH affect the behaviour of water within the flow channels [45]. Low CAH can facilitate water removal, reducing the risk of flooding, while high CAH can hinder droplet motion, leading to inefficient liquid management [46].

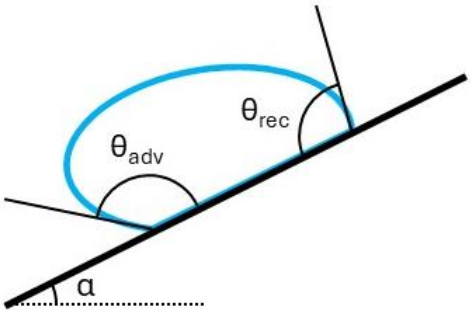


Figure 10. A droplet deposited on an inclined plane shows two different angles at the front and rear.

1.4. Numerical modelling of wetting phenomena

Computational Fluid Dynamics (CFD) has become an indispensable tool for the analysis and understanding of complex phenomena occurring within electrochemical devices such as PEMFCs. Through CFD, it is possible to virtually "look inside" the cell during operation, exploring convective and diffusive mass transport, the behavior of liquid water within the porous layers and along the flow channels, as well as the coupling between temperature fields, electrochemical reactions, and capillary forces. Multiphase simulations allow the investigation

of phenomena that are difficult or impossible to capture experimentally in situ, providing both qualitative insights and quantitative data to support analytical models and design strategies.

In this context, “multiphase flow” refers to systems in which two or more phases coexist, with dynamically evolving interfaces. In PEM fuel cells, this manifests as the formation of thin films, slugs and droplets due to water forming in the CL, migrating through the GDL, and eventually condensing and being transported along the flow channels of the BP. Modeling these processes is essential to predict flooding conditions, optimize channel geometry, design effective water removal strategies, and evaluate the influence of surface wettability on pressure drop and overall cell performance [45-48].

Several numerical approaches exist for modeling multiphase systems, differing in how they track or capture the interface. In “front-tracking” methods [49], the interface is explicitly discretized and followed in time, providing high geometric accuracy but suffering from topological complexity when droplets merge or break up. In contrast, “interface-capturing” methods describe the interface implicitly through field variables defined across the entire computational domain, allowing straightforward handling of coalescence and breakup at the expense of geometric precision. Among interface-capturing techniques, the Volume of Fluid (VOF) method [50] is widely adopted for engineering applications. It introduces a volume fraction field α representing the proportion of a given phase in each computational cell and reconstructs the interface geometry through Piecewise Linear Interface Calculation (PLIC) algorithms, which are an evolution of Simple Line Interface Calculation (SLIC) algorithms. The field α varies between 0 and 1, denoting the local fraction of liquid phase: $\alpha = 0$ corresponds to pure gas, $\alpha = 1$ to pure liquid, and intermediate values ($0 < \alpha < 1$) identify the cells cut by the interface. As highlighted from Figure 11, SLIC reconstruction creates an interface which is oriented in the direction of the axis; on the other hand, with PLIC the gradient $\nabla\alpha$ provides both the position and orientation of the interface, while its magnitude defines a thin transition region where surface tension forces act. In these interfacial cells, curvature and capillary stresses are evaluated in a diffuse yet continuous way, enabling the modeling of multiphase dynamics with sharp and mass-conserving interfaces. For both cases is worth noting a couple of fundamental points: mass is conserved, which is one of the predominant pros with respect to other methods [51]; however we “pay” for that since some geometrical discontinuities are visible on the interface. Local discontinuities in the reconstructed interface between neighboring cells may lead to jumps in curvature and, consequently, to numerical artifacts during simulations.

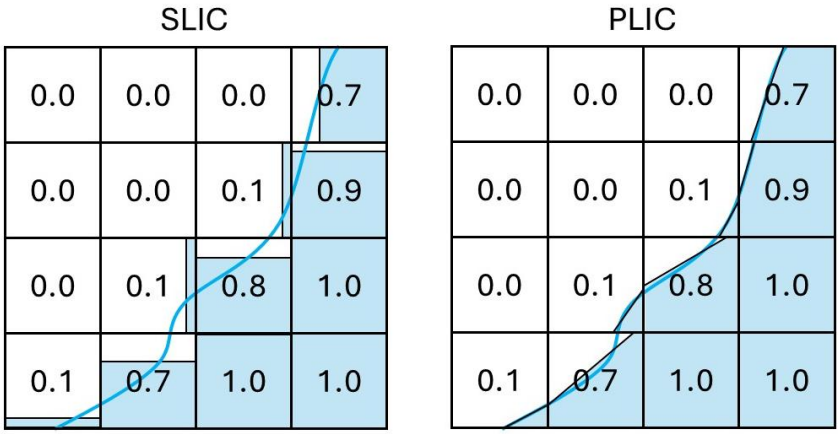


Figure 11. Comparison between SLIC and PLIC interface reconstruction

The accurate evaluation of curvature and surface tension forces remains one of the major challenges of interface-capturing methods. Small numerical errors in curvature discretization can generate "spurious currents" [52], i.e., unphysical parasitic velocities near the interface that can dominate the physical motion of small droplets. These artifacts arise from slight asymmetries in the calculation of the interface curvature which introduce unbalanced capillary forces even in static configurations. The curvature is calculated as follows:

$$\kappa = -\nabla \cdot n_i \tag{11}$$

Where

$$n_i = \nabla\alpha/|\nabla\alpha| \tag{12}$$

Is the normal direction to the interface evaluated as the ratio between the gradient of α and its absolute value. Advanced curvature-smoothing techniques and geometrically consistent reconstruction algorithms are therefore critical to enhance stability and fidelity in VOF simulations.

In the VOF formulation, the transport of α is governed by:

$$\frac{\partial\alpha}{\partial t} + \nabla \cdot (\alpha U) + \nabla \cdot [\alpha(1-\alpha)U_r] = 0 \tag{13}$$

where U is the velocity field and U_r an artificial compression term preserving interface sharpness. The surface tension is modeled through the Continuum Surface Force (CSF)

approach [53], in which interfacial forces are spread over a finite thickness region proportional to ∇α:

$$f_\sigma = \sigma\kappa\nabla\alpha \tag{14}$$

A particularly delicate region for multiphase modeling is the contact line, where the liquid–gas interface meets the solid wall. The contact angle θ defines the local interface orientation, constrained by the wall normal n_w:

$$n_i \cdot n_w = \cos\,\theta \tag{15}$$

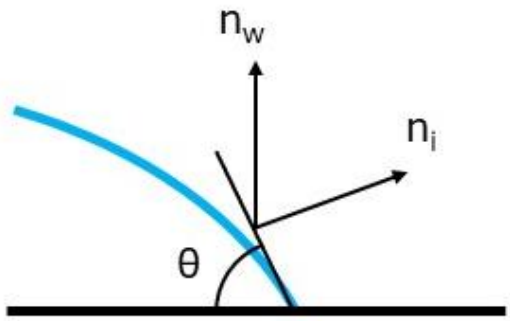


Figure 12. Visualization of contact angle and its relationship with interface and wall normal

Accurate reconstruction of the interface near the wall is essential to correctly capture capillary forces and droplet dynamics [54,55]. However, the finite-volume discretization implies that the geometric representation of the wall and of the interface plane are inherently approximate. As a consequence, the directions of n_i and n_w may not perfectly coincide in the cells adjacent to the wall, leading to local curvature errors or nonphysical pinning of the contact line.

During droplet motion, the apparent contact angle varies as a function of the contact-line velocity U, often expressed through the capillary number

$$Ca = \mu U/\sigma \tag{16}$$

where μ is the dynamic viscosity of the liquid.

To sum up this section, multiphase simulations inherently involve several numerical difficulties: (i) spurious currents from curvature discretization, (ii) mass loss during advection, and (iii) the trade-off between accuracy and computational cost. The geometric accuracy of the interface and the conservation of mass cannot be simultaneously improved beyond a certain

limit: highly accurate geometric reconstructions are computationally expensive and sensitive to mesh orientation, while more robust advection schemes tend to smear the interface or induce liquid volume loss. For this reason, modern implementations seek a balanced compromise between robustness and precision, often through Adaptive Mesh Refinement (AMR) and curvature-correction schemes.

In the classical PLIC formulation, the interface normal is estimated from the gradient of α, and a linear plane is fitted such that the fluid volume inside each cell matches the prescribed α value. A different class of methods, the Level-Set (LS) [56] and Coupled Level-Set–VOF (CLSVOF) [57] formulations, represent the interface as an iso-surface of a signed distance function φ. This provides a smooth and continuous definition of interface normals and curvature, improving geometric accuracy. However, Level-Set methods alone are not mass-conservative, leading to gradual volume loss during advection. Hybrid CLSVOF methods combine the strengths of both: mass conservation from VOF and curvature precision from Level-Set, at the cost of increased complexity [58].

Other techniques, such as the Lattice-Boltzmann Method (LBM) [59], model multiphase flow from a mesoscopic perspective, solving a discrete form of the Boltzmann equation. LBM is highly parallelizable and well-suited for flows in porous or microstructured domains, although it may exhibit instability at high density or viscosity ratios.

As seen in previous sections, from a physical standpoint an adhering droplet exhibits three characteristic contact angles: advancing θ_{adv}, receding θ_{rec}, and static θ_{sta}; it is also worth mentioning that intermediate states present a continuum of configurations influenced by surface roughness, heterogeneities, and contamination. Numerical models, however, often show difficulties in the implementation of a dynamic contact angle. Therefore, extensions of the VOF method through dynamic boundary conditions represent a physically grounded alternative for multiphase modeling. For this reason, accurately describing contact-line dynamics is essential to replicate the observed behavior of liquid water in PEMFC channels.

The present work first explores the potentiality of a static contact angle over a reconstructed GDL; then addresses the challenges linked to contact angle modelling by developing a custom boundary condition within OpenFOAM [60]. The proposed formulation modifies the local contact angle based on instantaneous interface evolution and hydrodynamic conditions, bridging the gap between physically observed wetting dynamics and their numerical representation.

2. Motivation and scope

The aim of this work is to provide a clear, adaptable, and replicable framework for implementing a physically grounded dynamic contact-angle model within numerical simulations of multiphase flows.

The core motivation arises from the limitations of standard boundary-condition-based approaches, which typically manipulate the contact angle locally at the cell level. While computationally convenient, such formulations do not explicitly enforce a global force balance on individual droplets and often rely on tuning parameters (e.g., velocity scales) that lack a direct physical interpretation. The present work introduces a shift toward a droplet-based force-balance formulation, in which adhesion and aerodynamic forces are evaluated at the scale of the entire droplet rather than at individual boundary cells.

In the first part of the study, after reconstructing the upper layer of the GDL of a membrane, droplets of different sizes are deposited to evaluate the temporal evolution of the water phase. In this configuration, a fixed contact angle is adopted, as rolling behaviour dominates droplet removal and the focus is placed on geometric reconstruction and phase tracking.

In the second part, a two-dimensional simulation is performed on a perfectly smooth surface, implying complete droplet adhesion. This simplified configuration serves to isolate and compare the three main contact-angle modeling approaches available in literature: static, quasi-dynamic, and dynamic. The objective is to highlight the fundamental differences in predicted onset of motion and force balance behaviour.

In the third and final part, a three-dimensional simulation is carried out in the OpenFOAM environment, exploiting the flexibility of its open-source structure. Starting from a default OpenFOAM boundary condition (e.g., dynamicAlphaContactAngle), the formulation is extended to develop a solver capable of:

- identifying individual droplets within the computational domain
- reconstructing the interface geometry
- integrating drag forces over the droplet surface
- comparing aerodynamic loads with adhesive forces
- defining a droplet-by-droplet pinning or sliding condition

This approach enables a physically consistent evaluation of the onset of motion, based directly on advancing and receding contact angles and on the instantaneous aerodynamic load. By integrating forces over the entire droplet surface, the model naturally captures non-local effects such as aerodynamic shielding between droplets, which are difficult to reproduce using purely local boundary conditions.

Although a PEMFC channel is adopted as the reference geometry, the addressed challenge is common to all wall-bounded multiphase flow systems in which droplet adhesion and shear-

driven motion coexist. The accurate prediction of droplet deformation, pinning, sliding, and interaction under shear remains a major modelling challenge, particularly in turbulent environments.

It should be noted that the present implementation adopts certain simplifying assumptions. The interface reconstruction and force-integration procedure are currently developed for orthogonal meshes, and droplet motion is approximated through a rigid-body-like sliding assumption in which the contact line velocity is linked to the center-of-mass velocity. While computationally efficient and suitable for moderate Capillary numbers, this approximation limits the description of strongly asymmetric contact-line dynamics and extreme deformation regimes. In particular, at high Capillary numbers, droplets may undergo significant elongation before sliding, potentially leading to breakup or pearling phenomena, which are not fully resolved within the presented formulation.

By combining theoretical modelling, numerical implementation, and controlled validation, this research aims to provide a robust and extensible computational framework for shear-driven wetting phenomena. The developed code is released as open-source software to promote transparency, reproducibility, and further methodological development by the scientific community.

3. Materials and methods

This section will present the modeling methodologies adopted for the different stages of the work described in this thesis. The analysis progresses from the construction of a realistic GDL model employing a constant contact angle, to the implementation of a User-Defined Function (UDF) describing a dynamic contact angle, and finally to the formulation of what will hereafter be referred to as the Force-Based Dynamic Contact Angle boundary condition.

3.1. A realistic GDL

The GDL [61] is a key component of Proton Exchange Membrane Fuel Cells (PEMFCs), located between the CL and the reactant gas flow channels (see Figure 13). It performs several crucial functions that ensure the proper operation of the cell, including:

- the homogeneous distribution of reactant gases toward the catalytic sites [62].
- the drainage of water generated by the electrochemical reactions [63].
- the electronic and thermal conduction across the membrane–electrode assembly [64].
- the mechanical contact support and internal humidity management [65].

Due to its fibrous and highly porous structure (with porosity typically ranging from 0.6 to 0.8), the GDL represents an extremely complex two-phase transport medium, in which the interactions between gas and liquid flows strongly influence the overall efficiency of the fuel cell.

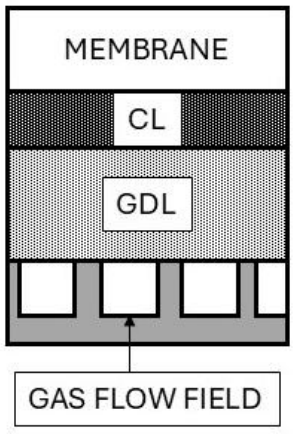


Figure 13. Schematic representation of a typical half fuel cell.

The two main types of GDLs commercially used are carbon paper and carbon cloth [66], which differ essentially in the arrangement of their fibers. The former (carbon paper) features a random and disordered fiber distribution, whereas the latter (carbon cloth) exhibits a regular and easily recognizable woven pattern, similar to that of a textile fabric. This inherent randomness in carbon paper structures makes their complete experimental characterization extremely challenging. Modern X-ray computed tomography techniques [67,68] have made it possible to obtain three-dimensional reconstructions of these materials; however, their high computational cost and limited spatial resolution have motivated the development of stochastic numerical reconstruction methods [69]. Conversely, carbon cloth GDLs exhibit a highly ordered fibre architecture with distinct interlaced layers, allowing their reconstruction to rely primarily on the identification of characteristic geometrical parameters specific to each type of GDL, which can then be numerically reproduced [70]. The GDL employed in the present study and used as the basis for the first stage of the modelling activity belongs to this latter category.

3.1.1. Experimental measurements

In order to support the numerical simulations and validate the computational models, a comprehensive experimental campaign was carried out to characterize both the microstructural and wetting properties of the GDL. The experimental activities focused on determining the geometrical features of the woven carbon cloth, quantifying its static wettability through contact angle measurements, and analyzing droplet dynamics under controlled airflow conditions. The overall goal was to obtain reliable input data and validation benchmarks for the CFD simulations of water droplet detachment and transport on realistic GDL surfaces. The used GDL is shown in Figure 14.

The GDL under investigation corresponded to the carbon cloth layer of a Celtec®-P1000 Membrane Electrode Assembly (MEA), featuring an active area of 45 cm² (6.72 × 6.72 cm). The woven texture of this layer, consisting of interlaced carbon fibers, provided the characteristic surface topography relevant to the study of droplet adhesion and removal. All experiments were performed on pristine, untreated GDL samples to ensure representative measurements of intrinsic material properties. The surface morphology and texture of the GDL were characterized using a digital optical microscope (Technaxx TX-158) equipped with an 8-LED ring crown illuminator and a 1000× zoom system. In full-chip configuration, this system achieved a spatial resolution of 5.8 µm/pixel. High-resolution images were acquired across the GDL surface to capture the woven arrangement of fibers and quantify the structural parameters (fiber diameter, pore size, and warp/weft pitch).

Following methodologies derived from the textile industry [71,72], a custom image-processing routine was developed in NI Vision 2020. The algorithm extracted the cross-sectional profiles of the interlaced carbon fibers and delineated the geometrical profile of the fabric surface. These data were later used to define the sinusoidal wave parameters (height, wavelength, and weft width) employed in the 2D and 3D geometric reconstruction for CFD domain generation.

The GDL surface was mapped in 25 Regions of Interest (ROIs, see Figure 15), each of 1 × 1 cm², to ensure spatial representativeness of the measurements.

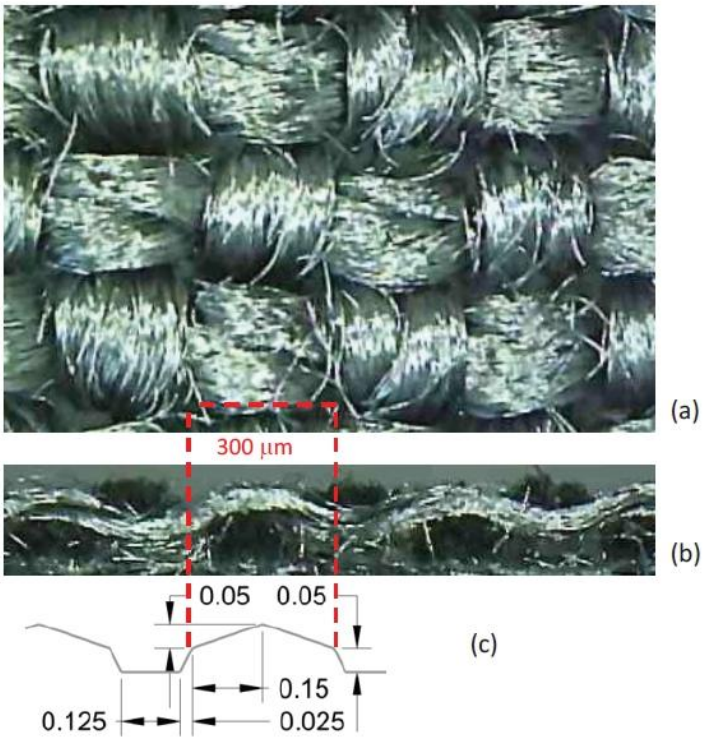


Figure 14. Top (a) and front (b) view of the interlaced structure of the carbon cloth GDL under investigation; extraction of base geometric parameters (c).

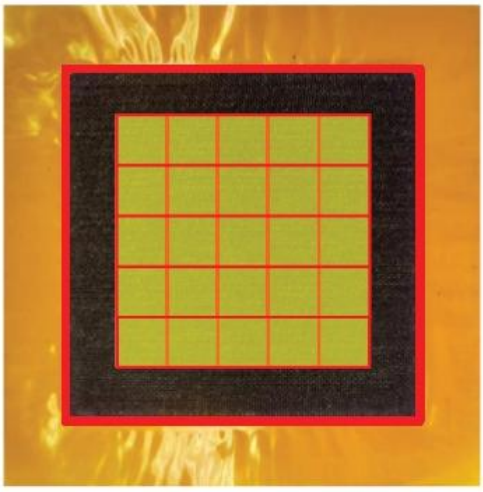


Figure 15. Division of membrane active area in 1x1 mm squared ROI.

To assess the intrinsic wettability of the GDL, water droplets were deposited using a high-precision stainless-steel medical needle. The internal capillary diameter imposed a minimum droplet size of approximately 400 µm. Multiple droplets were placed within each of the 25 mapped ROIs (see Figure 15) to ensure spatial coverage and reproducibility. Droplet geometry

and contact angles were analyzed using the previously described digital imaging setup. For each droplet, the static contact angle was determined by fitting the droplet contour at the liquid–solid–air interface. A total of 115 droplets were analyzed, covering diameters in the range 400–800 µm. As shown in Figure 16, the most probable contact angle was found to be 123° ± 2°, consistent with literature values for woven carbon cloth GDLs. The contact angle remained substantially constant across droplet size classes, confirming the homogeneity of surface wettability. The mean value of 123° was therefore used as boundary condition for wettability in subsequent CFD simulations.

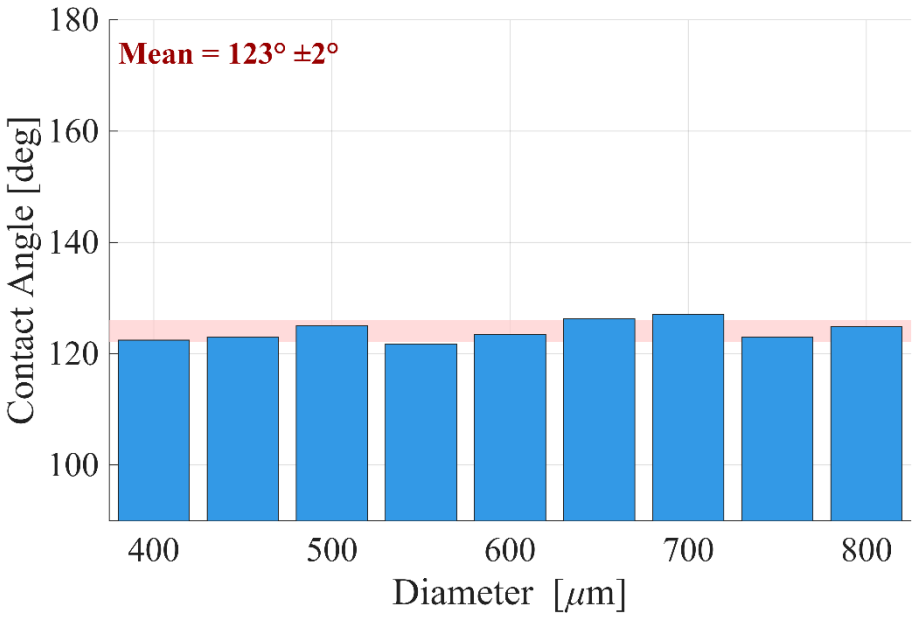


Figure 16. Measured contact angle for various droplet classes.

The transient motion of water droplets under airflow was captured using a 512 × 512 CMOS high-speed camera, capable of 5000 frames per second at full resolution. The optical system, equipped with a 25 mm objective, provided a spatial resolution of 55 µm/pixel. An exposure time was selected to achieve a frame rate of 1000 images per second, ensuring a balance between illumination intensity and temporal accuracy. The resulting image sequences were processed using a dedicated NI Vision 2020 script to extract the instantaneous droplet contour and the center of mass trajectory during motion. A 10 m/s airflow was applied tangentially to the GDL surface using a fan located 50 mm from the droplet deposition region. The local air velocity was continuously monitored with a hot-film anemometric probe, with a measurement accuracy of ±0.1 m/s. The tests were conducted under open-channel conditions, allowing free air circulation around the sample. This configuration isolated the effects of airflow and surface roughness on droplet motion, neglecting the influence of humidity or phase change, which were outside the scope of this study.

For each droplet, the position of the luminous center of mass was tracked frame-by-frame, enabling the computation of instantaneous velocity and acceleration profiles.

3.1.2. Representative geometry reconstruction

A three-dimensional geometrical reconstruction of the GDL surface was performed based on optical measurements described in the previous sub-chapter. This reconstruction aimed at defining a numerically efficient yet physically representative domain suitable for CFD simulations of multiphase flow using the VOF approach.

The reconstructed geometry was modelled in CREO Parametric 8.0, following a procedural design strategy aimed at reproducing the characteristic woven topology (Figure 14). The workflow can be summarized as follows:

a. Definition of base waveforms
   Four sinusoidal trajectories were created to represent the undulation of the warp and weft fibers. Their amplitude and wavelength corresponded to the average measured values from optical data.

b. Profile generation
   A trapezoidal cross-section was swept along each trajectory to mimic the shape of the individual yarns composing the GDL fabric. This approach allows controlling both fiber thickness and inter-fiber spacing.

c. Surface merging and trimming
   The swept surfaces were merged and trimmed at their intersections to obtain a continuous top surface describing the woven pattern. The resulting geometry defined a square elemental region, representing the smallest repeating unit of the GDL.

d. Serial repetition of the unit cell
   Using the Serial Repetition tool in CREO, the unit region was periodically replicated to build a surface of desired dimensions, suitable for later import into the CFD pre-processing environment.

This methodology is visualized in Figure 17 and provided a parametric and scalable reconstruction, easily adjustable for exploring different GDL configurations or comparing realistic and simplified topographies.

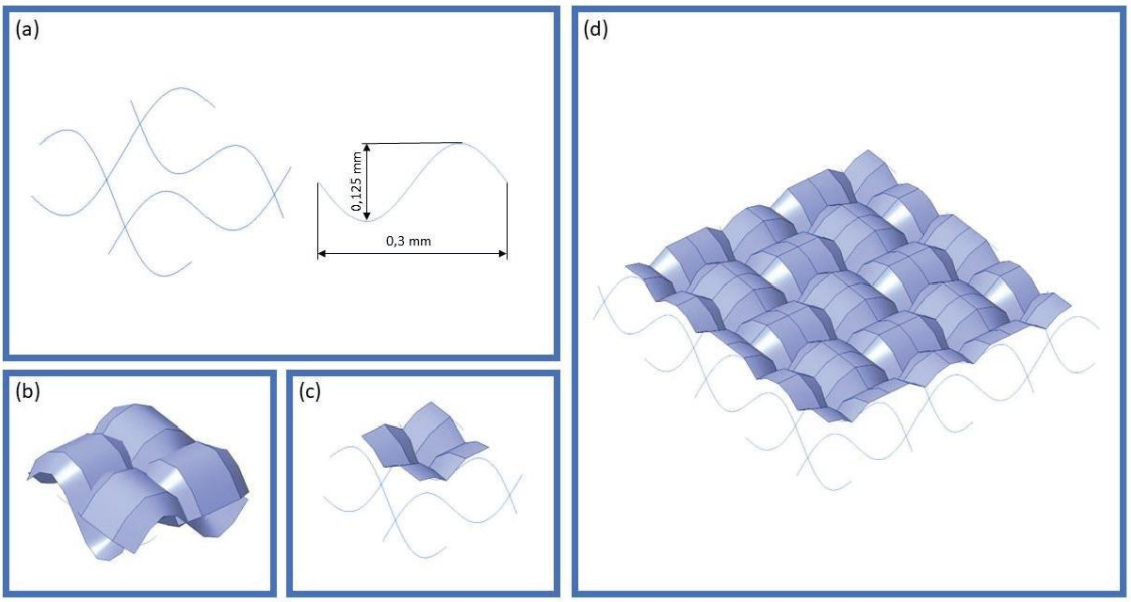


Figure 17. Step performed to model the realistic GDL

In addition to the fully realistic GDL surface, two further configurations were generated for comparison and parametric analysis:

- a completely smooth surface, representing the simplified model commonly used in literature.
- a hybrid surface, characterized by reduced amplitude (0.075 mm) and increased wavelength (0.4 mm) of the sinusoidal pattern.

These variations were implemented using the same procedural approach described above, ensuring consistency in domain dimensions and mesh topology. Such comparative models enabled evaluating the influence of surface roughness and pore morphology on droplet motion and detachment dynamics. Figure 18 shows a top view of the modelled hybrid and realistic GDLs.

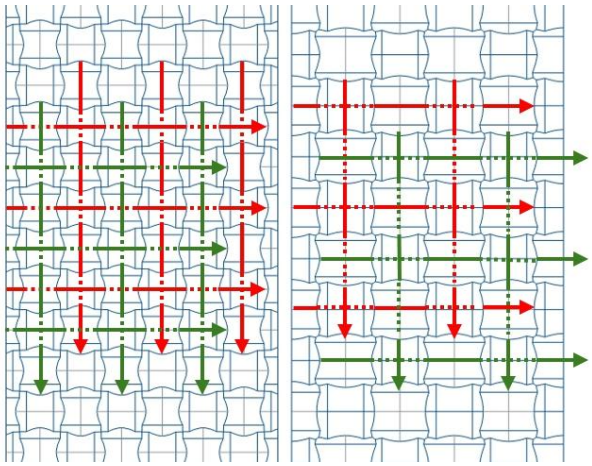


Figure 18. Visualization of the difference between the generated GDL (left: realistic; right: hybrid)

3.1.3. VOF Simulations

In the present research framework, the VOF formulation forms the numerical backbone of the simulations discussed in subsequent chapters, providing direct insight into the coupled effects of surface morphology, wettability, and airflow on droplet behavior. In the VOF approach, the interface between immiscible fluids is captured through the evolution of a scalar field, namely the volume fraction α of one of the phases (typically the liquid one in two-phase domains) within each computational cell. For any point in the computational domain:

$$\alpha = \begin{cases} 0 \\ 1 \\ 0 < \alpha < 1 \end{cases} \quad \begin{cases} pure\ primary\ phase\ (e.g.air) \\ pure\ secondary\ phase\ (e.g.liquid\ water) \\ interface\ region \end{cases} \tag{17}$$

The transport of the liquid fraction is governed by the phase continuity equation:

$$\frac{\partial \alpha}{\partial t} + \mathrm{u} \cdot \nabla \alpha = 0 \tag{18}$$

where u is the local velocity field. The effective density (ρ) and viscosity (μ) in each control volume are computed as weighted averages of the individual phase properties; for a two-phase, liquid-gas domain, we can write:

$$\rho = \alpha \rho_l + (1 - \alpha)\rho_g \tag{19}$$

$$\mu = \alpha \mu_l + (1 - \alpha)\mu_g \tag{20}$$

where subscripts l and g denote the liquid and gas phases, respectively. The momentum equation is solved for the mixture field:

$$\frac{\partial (\rho \mathrm{u})}{\partial t} + \nabla \cdot (\rho \mathrm{uu}) = -\nabla p + \nabla \cdot [\mu(\nabla \mathrm{u} + \nabla \mathrm{u}^T)] + \rho \mathrm{g} + \mathrm{f}_\sigma \tag{21}$$

Here f_σ represents the surface tension force, incorporated using the CSF model, as seen in Eq. (*14*). This term ensures that capillary effects are correctly represented at the fluid-fluid boundary.

The reconstructed GDL surface was used as the bottom wall of a computational domain with dimensions 1 × 1 × 4.8 mm³. This configuration simulated a portion of the gas channel over

the GDL, maintaining a balance between spatial representativeness and computational cost. Boundary conditions were defined as follows:

- Inlet boundary: constant airflow velocity of 10 m/s.
- Outlet boundary: pressure outlet at atmospheric pressure.
- Top and side walls: no-slip condition.
- Bottom wall: hydrophobic, impermeable GDL surface.

The base grid size was set to 0.02 mm, with three inflation layers near the GDL to accurately capture velocity and shear gradients close to the wall.

The simulation domain was initialized with still air. A liquid water droplet of controlled diameter (ranging from 200 to 600 μm) was then deposited on the GDL surface at 1 mm downstream of the velocity inlet. This boundary was set to 0 for the first millisecond to allow droplet stabilization on the surface; afterwards the airflow was set to the inlet value, and the simulation ended at the 25 ms mark. Surface tension between water and air was fixed at 0.072 N/m, and wall adhesion effects were enabled. All simulations assumed laminar flow conditions, consistent with the low Reynolds numbers typical of droplet-scale motion in fuel cell channels.

3.1.4. Results and validation

The numerical simulations and corresponding experimental observations provided a comprehensive characterization of liquid water droplet behavior on the woven carbon cloth GDL. The analyses were aimed at quantifying the influence of surface microgeometry on droplet detachment, transport velocity, and removal time, under the effect of a controlled 10 m/s air crossflow. Droplet diameters ranged from 200 to 600 μm, and in subsequent cases a pair of droplets was initialized at controlled distances to investigate mutual interaction effects.

The realistic and hybrid surfaces exhibited a longer detachment phase compared with the flat surface, while the performances were comparable with the hybrid configuration. The woven morphology introduced capillary anchoring points that delayed rolling and required a higher local airflow shear for detachment. Figure 19 shows that while droplets on the flat surface rapidly reached the outlet, those on the realistic and hybrid GDLs experienced slower motion and prolonged residence times.

As the droplet diameter increased to 600 μm, the influence of the microstructure diminished: the larger droplets spanned multiple pores, effectively averaging the surface roughness effects. As shown in Figure 20, droplets of 600 μm exited the computational domain almost simultaneously for both the realistic and hybrid geometries, while still showing a modest delay relative to the flat surface.

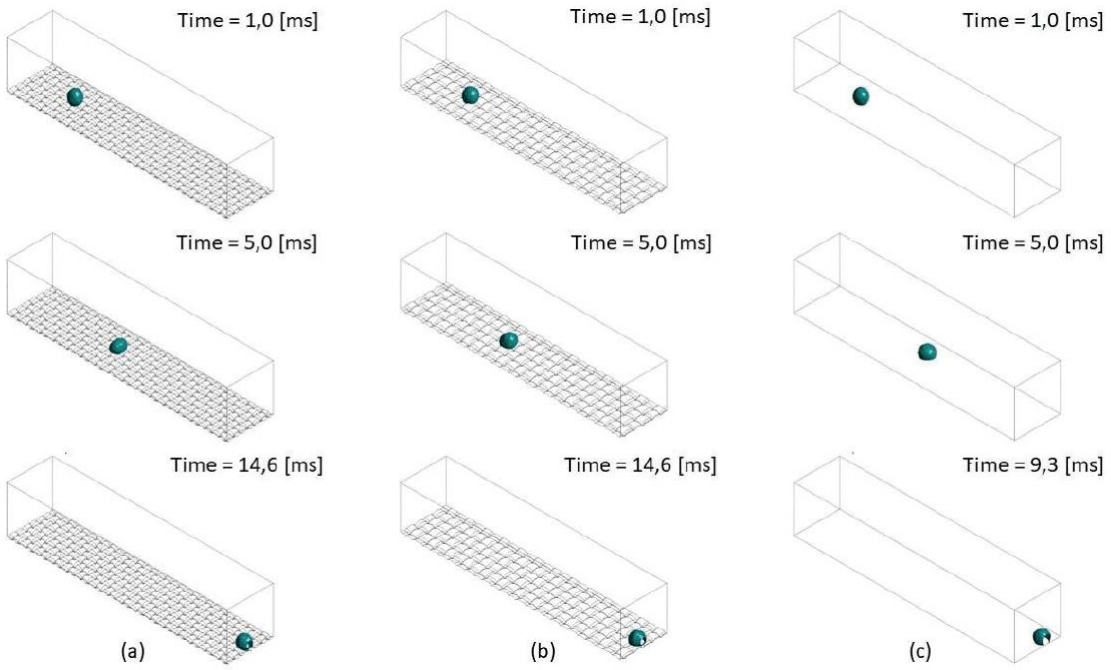


Figure 19. Water volume fraction isosurfaces (α = 0.8) for a 300 µm droplet on different GDL configurations: (a) realistic woven GDL, (b) hybrid GDL, and (c) flat reference surface.

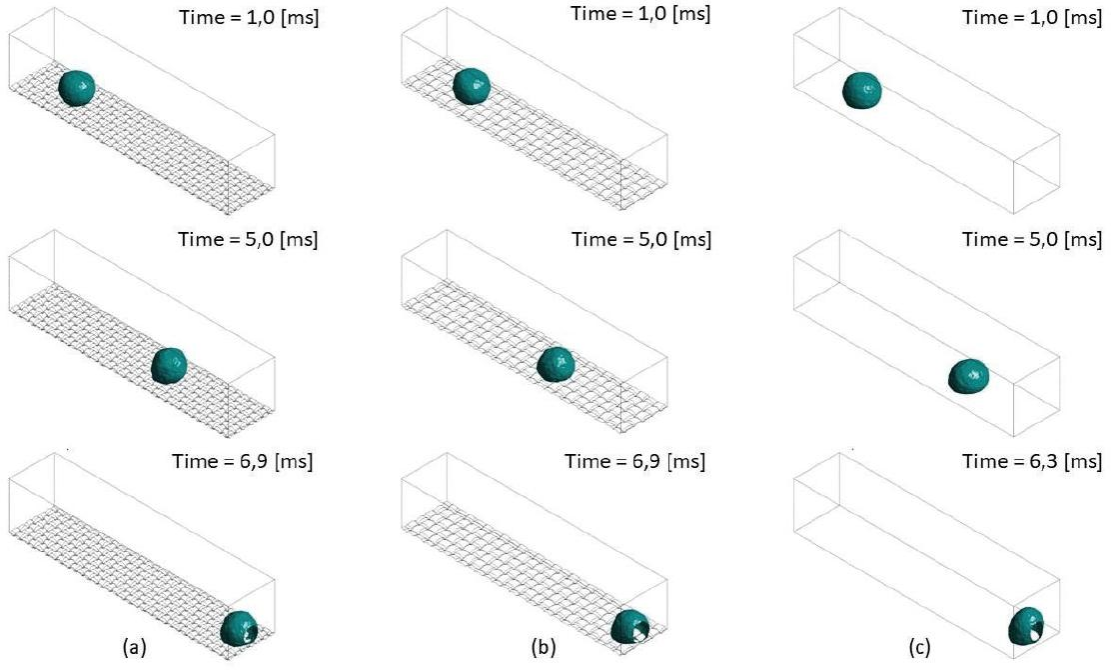


Figure 20. Water volume fraction isosurfaces (α = 0.8) for a 600 µm droplet on different GDL configurations: (a) realistic woven GDL, (b) hybrid GDL, and (c) flat reference surface.

High-speed imaging was used to validate the numerical results for a 600 ± 20 µm droplet. The recorded image sequence (Figure 21) captured the following sequence of events:

a. Deposition on the GDL surface.

b. Oscillations of the droplet contour lasting about 1 ms after the airflow was initiated, due to the balance between retention and detachment forces.
c. Onset of motion once a critical local velocity was reached.
d. Rolling displacement along the airflow direction, with negligible sliding behavior.

The rolling motion observed experimentally confirmed the suitability of using a fixed static contact angle in the simulations, without differentiating between advancing and receding angles. Further consideration will be made when dealing with higher wettability surface, where sliding behavior dominates.

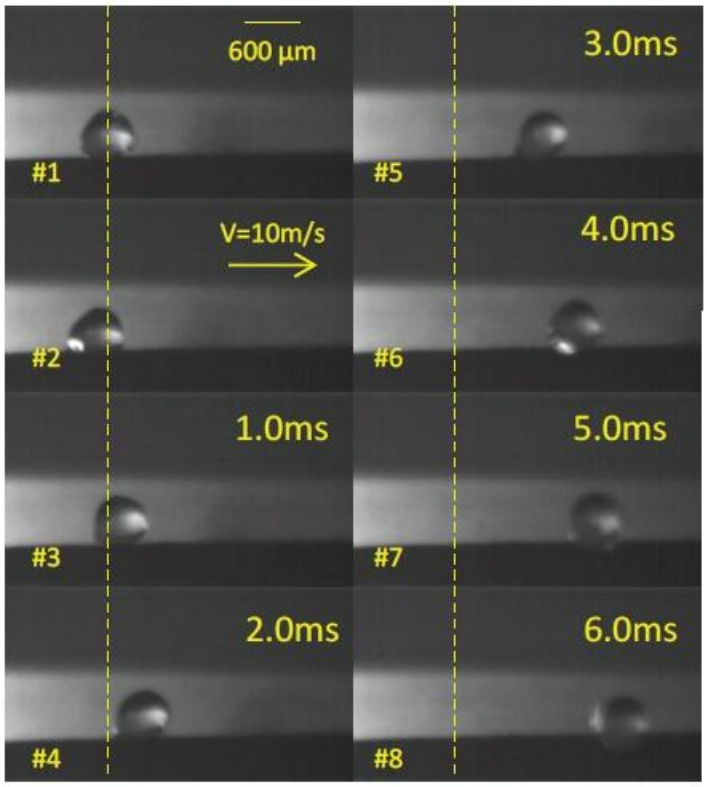


Figure 21. Experimental 600 µm droplet motion on the GDL

A direct comparison between experimental and simulated droplet trajectories showed good agreement (Figure 22). The only significant deviation occurred after approximately 5 ms, attributed to the open-channel experimental configuration, which generated lower aerodynamic pressure forces than the closed-channel numerical domain. Nevertheless, the agreement was sufficient to confirm the validity of the numerical model.

Figure 23 summarizes the mean droplet velocity for all analyzed diameters and GDL configurations. The data indicate that the effect of microstructure is more pronounced for smaller droplets, whose diameters are comparable to the pore size of the woven structure. For the smallest droplets (200–300 µm), the surface roughness significantly increased the resistance to motion, leading in some cases to complete droplet retention within the pores. Conversely, for 600 µm droplets, the mean velocity on the realistic and hybrid surfaces

remained about 8% lower than on the flat surface, confirming that even moderate surface roughness continues to influence the flow.

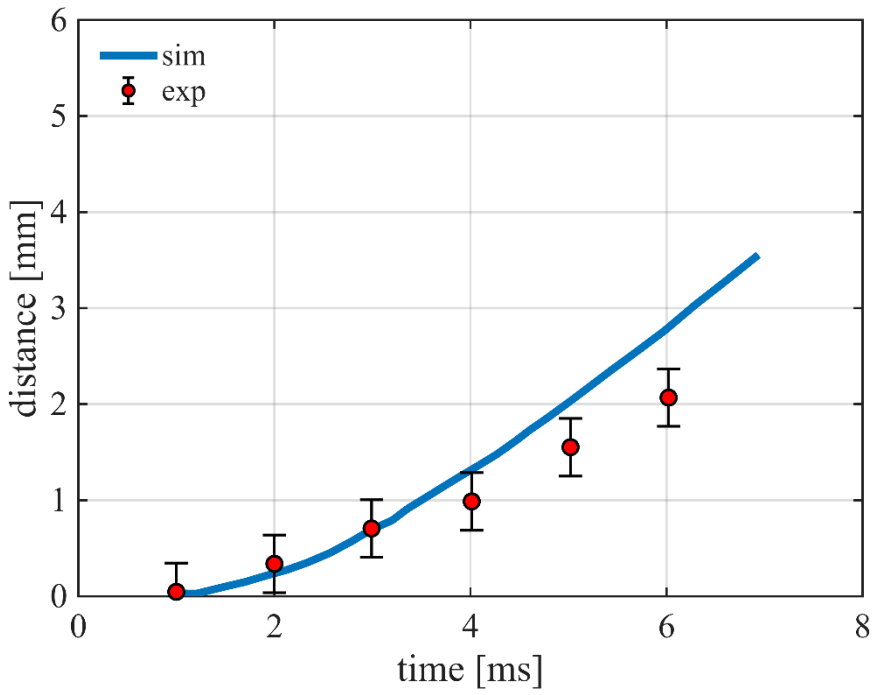


Figure 22. Droplet center of mass displacement (wrt initial deposition) over time

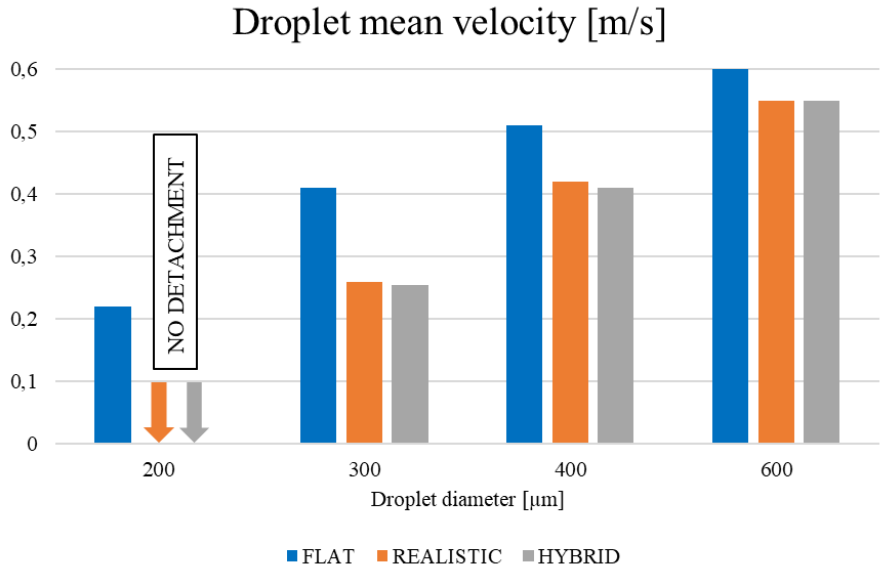


Figure 23. Droplet mean velocity for different diameter classes and considered GDL surface

The initial water removal time, defined as the instant when the leftmost water cell travel across a plane perpendicular to the airflow and passing through droplet center of mass at deposition, further highlighted the same trend (Figure 24). The flat surface yielded the shortest removal

time, while both the realistic and hybrid GDLs exhibited delayed detachment due to pore-scale obstacles that impeded initial rolling. Interestingly, despite its reduced amplitude and longer wavelength, the hybrid surface showed slightly higher resistance than the realistic one. This behavior is explained by its larger wetting area, which increases the adhesive force between the liquid and the surface, thus slowing the initiation of motion. These findings are in agreement with prior studies [73,74] that reported slower droplet dynamics on rough GDLs compared to idealized flat surfaces.

For the smallest droplets (≈200 μm), a no-detachment condition was observed in both the realistic and hybrid cases. Such droplets became trapped in the surface pores, representing a critical flooding scenario for PEMFC operation. These retained droplets can block reactant transport toward the catalyst layer and shield the airflow, delaying the removal of downstream droplets until coalescence or growth restores detachment conditions.

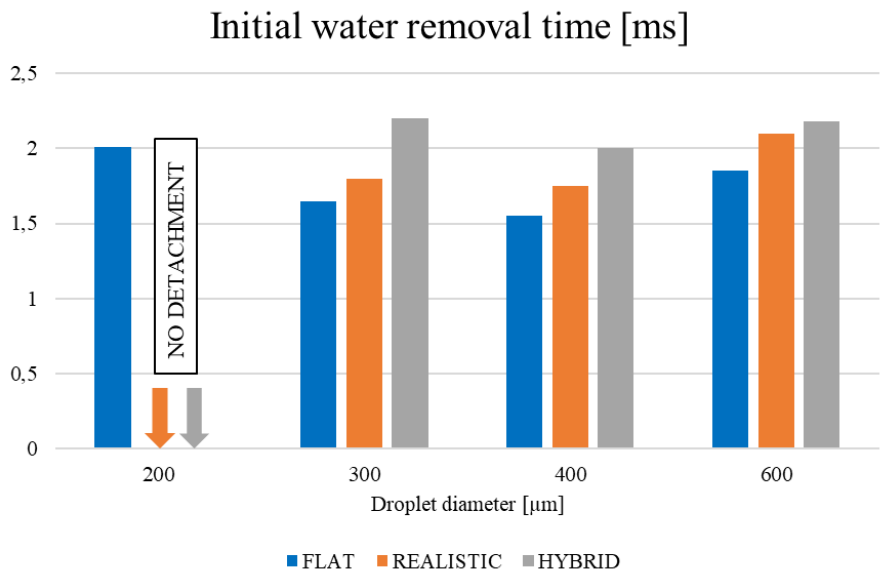


Figure 24. Droplet removal time for different diameter classes and considered GDL surface

To further investigate the combined effect of surface microstructure and droplet interaction, simulations were performed for pairs of 300 μm droplets positioned at controlled distances. The second droplet was placed several pores downstream from the first one, specifically in the fourth pore for both the realistic and hybrid geometries, and at an equivalent mean distance for the flat case (see Figure 25).

As shown in Figure 26, the mutual influence between droplets was clearly observed. On the flat surface, the leading droplet partially shielded the trailing one from the incoming airflow, resulting in different velocities and a clear velocity gradient as inter-droplet spacing decreased. On the realistic woven surface, a similar pattern emerged, though both droplets experienced lower mean velocities and longer detachment phases due to surface roughness. The hybrid surface exhibited the strongest retention effect: the second droplet became temporarily trapped

within a pore and only detached after merging with the first droplet. This configuration resulted in the longest exit time among all cases.

These results emphasize that GDL surface microgeometry not only affects single-droplet detachment but also modifies the collective behavior of multiple droplets, potentially altering the overall water removal dynamics at the channel–GDL interface. Moreover, they confirm that GDL asperities cannot be neglected in predictive models of water removal in PEMFCs; even small-scale variations in surface topography significantly influence droplet detachment dynamics, average transport velocity, and inter-droplet interactions.

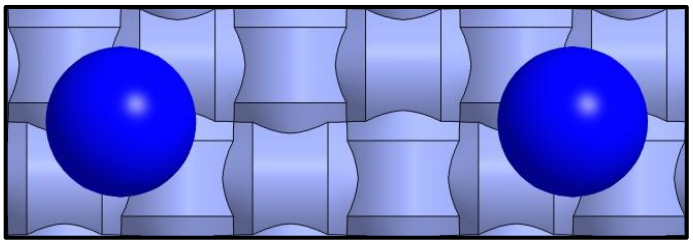


Figure 25. The second droplet was deposited in line with the first one, at the 4th pore

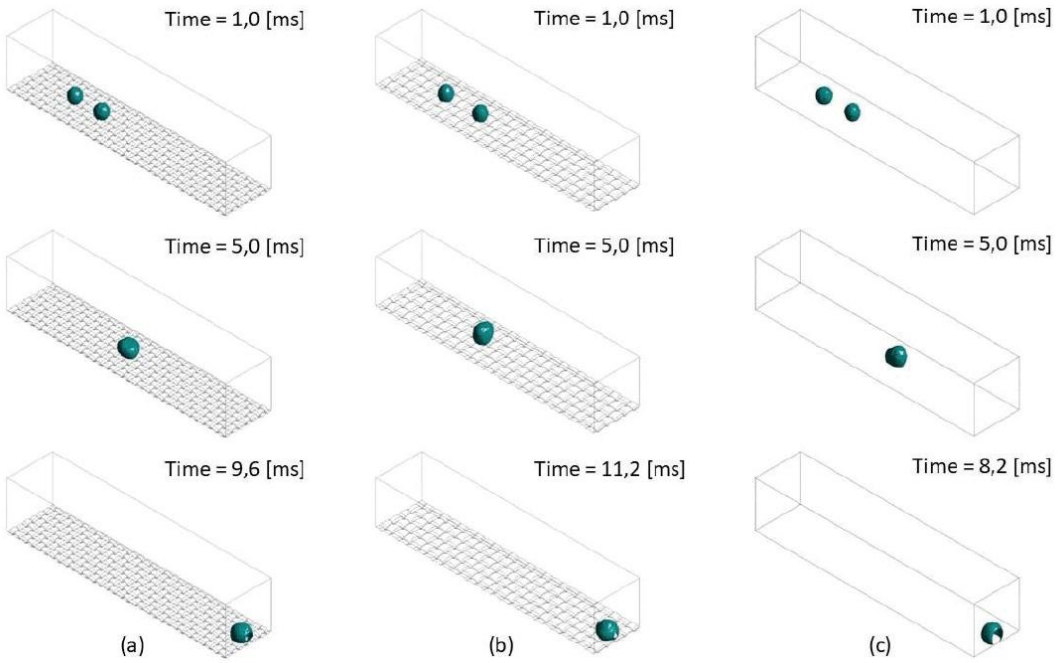


Figure 26. Water volume fraction isosurfaces (α = 0.8) for a 300-300 μm droplet pair on different GDL configurations: (a) realistic woven GDL, (b) hybrid GDL, and (c) flat reference surface.

3.2. Contact angle modelling in Fluent

As discussed in the previous chapter, rolling motion predominates on the GDL surface, which means that assuming a static contact angle does not critically affect the accuracy of the

simulations. The situation, however, is quite different for the channel walls, which are significantly more hydrophilic and where the formation of liquid films or slugs is often observed, slowly sliding toward the outlet. For this reason, three distinct boundary condition strategies were implemented and compared in ANSYS Fluent [75], two of which were developed through User-Defined Functions (UDFs):

I. Constant (Static) Contact Angle, the ready-made solution proposed by the commercial code.
II. Quasi-Dynamic Contact Angle.
III. Fully Dynamic Contact Angle, based on literature correlations.

All models were compared with optical experiments performed on hydrophilic PEMFC channel surfaces, characterized by static, advancing and receding angles of 50° ± 2°, 55° ± 2° and 45° ± 2°, respectively.

From an experimental standpoint, surface characterization followed a procedure analogous to that described for the GDL. However, in this case, the droplets assumed the shape of a spherical cap rather than a complete sphere. As shown in Figure 27, two key geometric parameters were defined: the maximum chord of the spherical cap B and its corresponding height H. Droplets of various sizes, with B ranging from 500 to 3000 μm, were deposited onto the channel surface, and their contact angle was measured, yielding an average value of 50.4°. A strong correlation ($R^2 = 0.98$) was also found between the chord length and droplet height. Consequently, once the surface is defined, the droplet morphology can be fully characterized using a single viewing angle (either front or top view). Deposition parameters for numerical initialization are B, H and θ; however, under the spherical cap assumption, only two of those parameters are independent (see Figure 28).

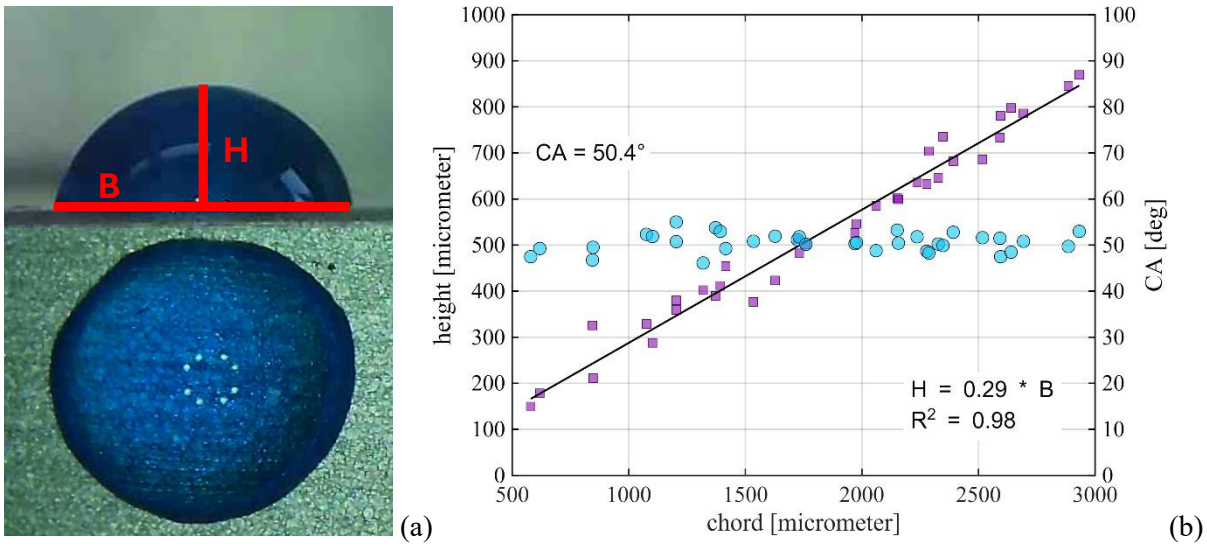


Figure 27. (a) Visualization of droplet deposited on channel and (b) morphological parameters (CA and H over B) for several droplet spanning over different chords classes.

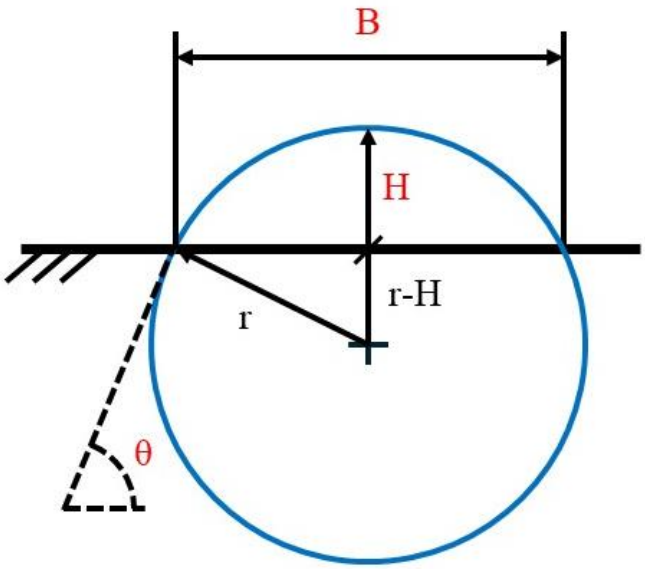


Figure 28. Droplet is deposited as a spherical cap in the domain

For this subset of simulations, the considered domain is a 2D channel, with a velocity inlet, ambient pressure outlet, and upper and lower walls with the appropriate contact angle formulation.

3.2.1. Static contact angle

The simplest approach available in Fluent assigns a fixed static contact angle to the wall boundary. This constant value is used to adjust the local surface normal at the interface between phases according to Eq. (15). In the present study, the experimentally measured static equilibrium contact angle of 50° was used. Although this simplification ensures computational stability, it does not distinguish between advancing and receding interfaces and completely ignores CAH. Consequently, it tends to overestimate the water removal capability of the surface, predicting premature detachment and higher droplet velocities.

3.2.2. Quasi-dynamic contact angle UDF

To introduce a more realistic representation of the moving contact line, a UDF-based quasi-dynamic model was implemented. This approach differentiates between advancing and receding contact lines depending on the local interface velocity U_{cl}:

$$\theta = \begin{cases} \theta_{adv} = 55° & if\ U_{cl} > 0 \quad (advancing\ front) \\ \theta_{rec} = 45° & if\ U_{cl} < 0 \quad (receding\ front) \end{cases} \tag{22}$$

Within the simulation domain, the contact line velocity corresponds to the velocity of the interface cell, that is, the cell located adjacent to the wall. As the droplet travels along the solid surface, the contact line effectively moves over a no-slip boundary. To circumvent this limitation, it is common practice to introduce slip or partial slip conditions near the contact line [76,77,78]. In the present work, the tangential component of the center of mass velocity was used as an approximation of the contact line velocity. The scalar product between the velocity vector and the VOF gradient was used to distinguish advancing from receding regions.

This formulation incorporates the experimentally observed contact angle hysteresis yet still neglects the continuous variation of the contact angle with capillary number; it therefore provides intermediate accuracy: an improvement over the static model but insufficient when dynamic wetting effects dominate.

3.2.3. Dynamic contact angle UDF

A more physically consistent Dynamic Contact Angle boundary condition was implemented through separate UDF and two different correlations, one for the advancing and one for the receding interface. Both models express the dynamic angle as a function of the capillary number Ca, evaluated as per Eq. (*16*). The advancing dynamic angle was computed following Kistler's correlation [79], which relates it to the static angle via the empirical Hoffman function f_H [80]:

$$\theta_{adv,dyn} = f_H\ (f_H{}^{-1}(\theta_{adv}) + Ca) \tag{23}$$

where the Hoffman function is defined as:

$$f_H(x) = \cos^{-1}\left(1 - 2\tanh\left(5.16\left(\frac{x}{1+1.31\,x^{0.99}}\right)^{0.706}\right)\right) \tag{24}$$

and its inverse can be approximated using the Hoffman–Voinov–Tanner relation:

$$f_H{}^{-1}(\theta_{adv}) = \frac{\theta_{adv}^3}{72\,Ca} \tag{25}$$

This formulation captures the increase in the advancing angle with higher contact line velocities, consistent with experimental wetting observations.

The receding dynamic contact angle was computed following the Tanner correlation [81], later reformulated by Nichita et al. [82], as:

$$\theta_{rec,dyn} = (\theta_{rec} - 72\ Ca)^{1/3} \quad (26)$$

This expression predicts a decrease in the receding angle with increasing capillary number, effectively modelling the contact line elongation during droplet motion.

3.2.4. Results and validation

The model validation was conducted using data from the experimental campaign. For this purpose, the droplet with a 930/260 (=B/H) ratio was selected, and the motion of its rightmost water cell in the simulation was compared to the corresponding position observed experimentally. As anticipated during the modeling phase, Figure 29 clearly demonstrates that using a static contact angle leads to faster water removal, as it completely ignores the evolution of advancing and receding angles and the resulting increase in adhesion force on the droplet. Incorporating a quasi-dynamic contact angle improves the simulation's accuracy; however, only the dynamic contact angle approach consistently produces results within the experimental error margins. Both the implemented correlations rely on contact line velocity, which in our simulations is approximated using the droplet's center of mass speed. Due to the relatively large cross-sectional area of the droplet, the associated velocities are significant, leading to noticeable differences between quasi-dynamic and dynamic contact angles. It should also be noted that the experimental measurements were obtained from a top view, while the simulations were conducted from a 2D frontal perspective. Nevertheless, as shown in Figure 27, droplet height H correlates well with chord B for sizes ranging from 500 to 3000 microns.

Figure 30 further explores model differences for the 930/260 case. Up to 1.0 ms, the interfaces of the droplet in different simulations nearly overlap, as the droplet is just beginning to bend in response to the airflow. At 3 ms, the impact of the applied contact angle becomes apparent when comparing the receding angles: the grey curve (static model) and the blue curve (dynamic model) diverge, reflecting the dynamic adjustment of contact angle in the simulation according to the center of mass velocity. By 5 ms, droplets have reached an elongated equilibrium shape, highlighting differences in advancing angles between the dynamic (blue) and static (grey) models, while the red curve approximates the experimentally measured θ_{adv}.

Figure 31 compares the center of mass displacement for the three contact angle strategies across all droplet classes. In all cases, the constant contact angle leads to faster water removal, whereas quasi-dynamic and dynamic approaches slow down droplet motion. As already mentioned, differences are negligible up to 1.0–1.5 ms, when droplets remain nearly stationary. Once inlet velocity increases, the evolving contact angle influences the center of mass behavior. For larger droplets, the difference between static and modeled contact angles decreases, indicating that the effect of contact angle evolution has less influence on the overall force balance for bigger

droplets. This suggests that accurately capturing the dynamics of smaller droplets requires careful modeling of contact angle variation.

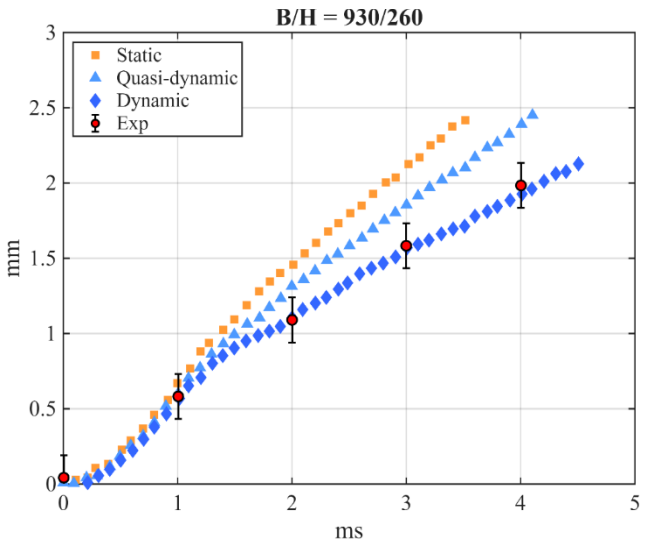


Figure 29. Validation of the dynamic contact angle model and comparison with the proposed approaches

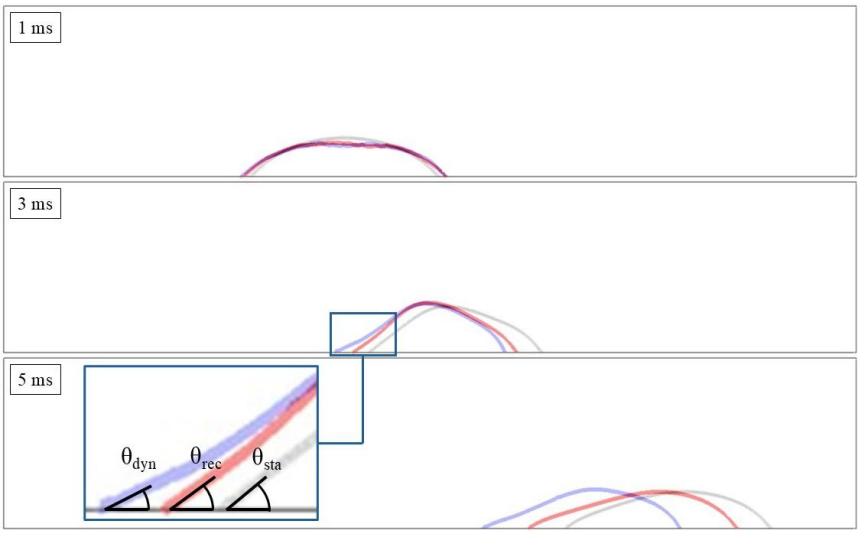


Figure 30. Comparison of 0.5 isosurface for static contact angle (grey), quasi-dynamic contact angle (red) and dynamic contact angle (blue); close up on the receding front to highlight the difference in the applied angle at the interface.

Interestingly, as droplet size decreases, the difference between the quasi-dynamic and dynamic approaches becomes progressively smaller. This behaviour can be physically explained by considering the underlying force balance: smaller droplets experience lower drag forces due to their reduced exposed area to the airflow. Consequently, the contact-line velocity decreases, leading to a lower Ca. Since Ca determines how much the advancing and receding angles deviate from their limiting values, low numbers implies that the dynamic contact angles remain close to the experimentally defined CAH range.

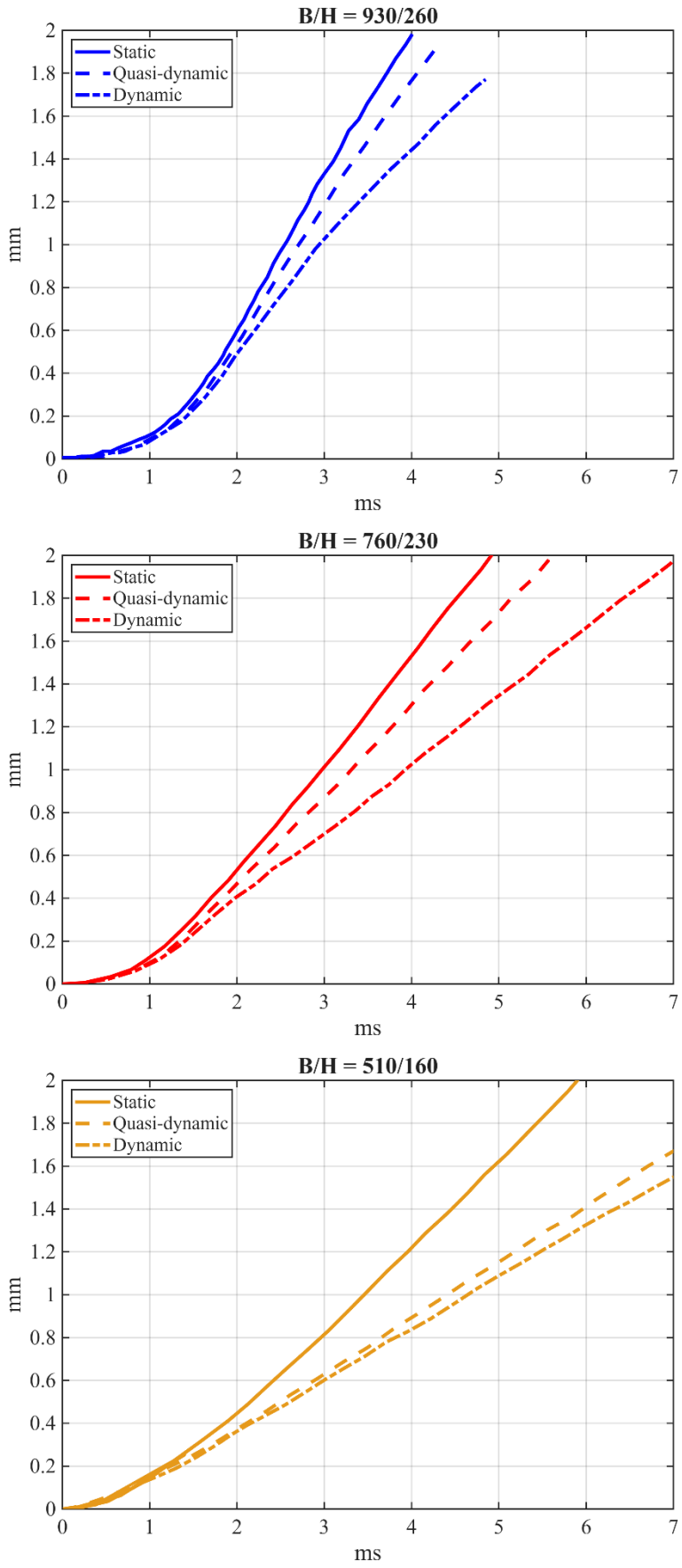


Figure 31. Comparison of centre of mass position for different contact angle modelling strategies and three droplet classes.

3.3. Force-Based Dynamic Contact Angle

This section represents the final and most advanced stage of the model development. The algorithm initially implemented in ANSYS Fluent was ported to OpenFOAM, with two main objectives. First, to enable the handling of multiple droplets simultaneously within the computational domain; and second, to allow the evaluation of the dynamic contact angle even under adhesion conditions, i.e., within the CAH range.

In previous implementations, whenever a droplet was pinned to the surface (adhesion regime), the static contact angle was directly imposed. Although computationally simple, this approach could induce numerical instabilities during the transition phase between adhesion and motion, as the contact angle would change abruptly rather than smoothly. The present formulation addresses this issue by introducing a force-based coupling between the local force balance and the instantaneous contact angle evolution, ensuring a physically consistent and continuous transition between regimes. Figure 32 provides a schematic block diagram to illustrate the extended logic, integrating the modifications with respect to the previous quasi-dynamic framework.

The following subsections describe the model implementation in detail and subsequently present its experimental validation. The analysis begins with droplet deposition on surfaces of different wettability, followed by the evaluation of drag-induced motion for a suspended droplet (i.e., not in wall contact) and on an inclined plane within a representative channel geometry.

It should be noted that, at the current stage of development, the model is fully functional only on orthogonal meshes, as the interface reconstruction for non-orthogonal grids has not yet been implemented.

To avoid excessive detail within the main text, the pseudocode of the implemented routines is reported in

Appendix A - Force-Based Dynamic Contact Angle **Pseudocode**.

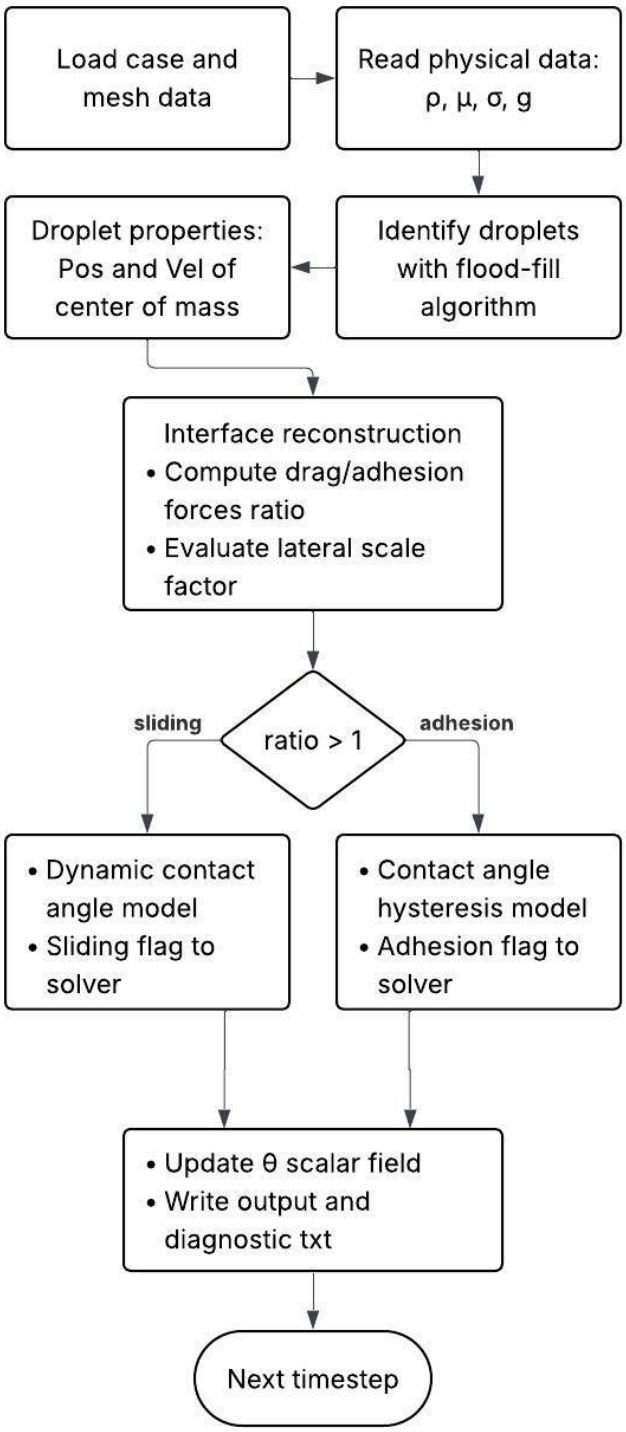


Figure 32. Flowchart of the proposed Force Based Dynamic Contact Angle

3.3.1. Boundary developement

This section describes the implementation strategy adopted for the development of the custom boundary condition and its integration within the OpenFOAM computational framework. The goal of this chapter is to provide a detailed overview of the code architecture, the additional functions introduced, and the modular design philosophy that guided the implementation. The overall approach was conceived to make the boundary condition as modular and flexible as possible, such that it could be easily modified or extended without affecting the behaviour of other parts of the solver. This modularity was essential for two reasons:

1. the present work represents a continuously evolving research effort, in which the logic of the boundary condition has been repeatedly refined to meet the changing requirements of different numerical experiments.
2. it allows future developers and researchers, who may wish to continue or adapt this work, to interact with the code more intuitively, focusing only on specific sections of interest without the need to alter the entire solver structure.

From a structural point of view, the implemented boundary condition was initially based on the dynamicAlphaContactAngle class provided by the OpenFOAM library. However, substantial modifications were introduced to extend its capabilities and to enable a more complete description of force-based and dynamically correlated contact angles. The modified version includes additional code blocks and custom functions that were progressively integrated and tested to ensure numerical stability and physical consistency.

For clarity and reproducibility, the custom implementation has been logically divided into a series of modular header files, each of which is responsible for a specific functionality within the boundary condition. The files are organized as follows:

1. readProperties.H
2. findDrop.H
3. dropG.H and dropVel.H
4. printGposition&velocity.H
5. forcesOnDroplet.H
6. printForces.H
7. lateralScaleFactor.H
8. contactAngleHysteresis.H
9. contactAngleCorrelations.H

Although some of these elements have been previously discussed in earlier sections of the thesis, they will be reintroduced and described again here for the sake of completeness and clarity, with a focus on their implementation details and their role in the overall numerical framework. Each of the aforementioned code modules is treated in a dedicated subsection of this chapter, and for each file the following aspects will be discussed:

- its specific purpose and functionality,
- the data exchange mechanisms between it and the main solver,
- the interaction with the Volume of Fluid (VOF) phase fields,
- the physical meaning of the parameters and variables managed within it.

3.3.1.1. readProperties.H

The file readProperties.H is responsible for loading from simulation folder all the physical quantities required by the boundary condition during its initialization. The routine first accesses the mesh associated with the current boundary patch and then reads two input dictionaries from the constant directory of the case:

- transportProperties file, which contains the thermophysical data of the working fluids
- uniform vector field g, representing the gravity acceleration.

Within the transportProperties dictionary, the sub-dictionaries water and air are used to extract the following quantities:

- dynamic viscosity of liquid water (μ)
- densities of air and water (ρ)
- surface tension coefficient (σ)

A default value (def = -999) is defined to verify the successful reading of each property; if any of them is missing, the execution is immediately stopped, and an explicit error message is issued. This safety check prevents the simulation from continuing with undefined or inconsistent physical data.

In the standard implementation of dynamic contact angle boundary condition, such physical properties are not usually required directly by the boundary itself, since they are already available to the solver through the transport models. However, in this work the boundary condition must evaluate the forces acting on the droplet, and for that reason it needs access to the main fluid properties and to the gravity field. These quantities are later used to compute the terms contributing to the overall force balance on the liquid phase.

In summary, readProperties.H serves as the initialization block that guarantees that all fundamental material and gravitational parameters are available to the boundary condition before any further calculations are performed.

3.3.1.2. findDrop.H

The findDrop.H module is responsible for identifying all liquid regions within the computational domain and assigning a unique numerical label to each of them. This operation enables the solver to distinguish between separate droplets, track their motion over time, and provide consistent information to subsequent routines that compute each droplet's position, velocity, and force balance. The algorithm is based on a flood-fill search procedure applied to the volume fraction field of the primary liquid phase (alpha.water), adapted from the method proposed by Heinrich and Schwarze for spray atomization particle tracking [83].

At each time step, the alpha1 field is read directly from the mesh database. A small threshold value (set here to 0.01) is used to discriminate between gas and liquid cells: only cells where alpha1 > 0.01 are considered part of a liquid body. A new scalar field, named vofID, is created

to store the identification number of each droplet. Initially, all cells are assigned a value of zero. The algorithm then iterates over all mesh cells: whenever it encounters a liquid cell ($\alpha >$ threshold) that has not yet been labelled, a new unique global identifier is generated. This identifier is then propagated to all neighbouring liquid cells connected to the original one, so that all the cells belonging to the same continuous liquid region receive the same ID.

The propagation is performed using recursive flood-fill logic, implemented through dynamic lists of neighbouring cells. Each newly discovered liquid neighbour is added to a temporary list, which is then expanded iteratively until no further connected cells remain. This ensures that all parts of a droplet, regardless of its shape or topology, are consistently labelled with a single identifier.

Because the code is designed to run efficiently in parallel environments, particular attention is devoted to communication between processors. In a decomposed mesh, droplets may span multiple subdomains, with their cells distributed across different processors. To handle this situation, the algorithm introduces a global indexing system to prevent inconsistencies in cell identifiers across processors. After the initial labelling phase, a parallel correction stage begins, ensuring that droplets crossing processor boundaries are not erroneously counted as distinct entities. During this stage, the code examines all coupled patches (i.e., processor-processor interfaces) and compares the IDs assigned to adjacent cells on opposite sides of the boundary. If two connected cells across the interface possess different positive identifiers, the higher value is replaced by the lower one. The correction then propagates recursively through the local neighbourhood until all connected cells share the same label. This correction-propagation process is repeated iteratively, with a synchronization step (“reduce” command) between processors at the end of each cycle, until no further corrections are required. In this way, every droplet spanning multiple processors receives a consistent and unique global label throughout the parallel domain. Once the labelling process is complete, the vofID field is written to disk and synchronized across all processors. A subsequent stage counts the total number of droplets and generates a list of all unique droplet IDs present in the domain.

For parallel runs, each processor first builds its local list of unique identifiers, which is then gathered by the master process using OpenFOAM’s collective communication utilities (Pstream::gatherList). The master process merges these lists, removes duplicates, and determines the final number of distinct droplets. The result is then broadcast back to all processors to ensure consistency across the domain. In serial runs, the same loops are executed locally, without the need for parallel communication.

At the end of the routine, the updated droplet information is written to external text files. Specifically, the code generates a file named dropNumber.txt, containing the current number of detected droplets, and a series of files droplet_i_label.txt, each storing the label of the i-th droplet. These files are updated only when the number of droplets changes, such as during coalescence or breakup events. This mechanism provides a lightweight interface between the solver and external post-processing scripts, allowing straightforward monitoring of droplet evolution without the need to read full field data.

In summary, findDrop.H implements a robust and parallel-consistent procedure for identifying and labeling all continuous liquid regions within the computational domain. It ensures that each droplet receives a unique and globally consistent identifier, forming the foundation for the

subsequent routines that compute droplet-level quantities such as position, velocity, and force balance.

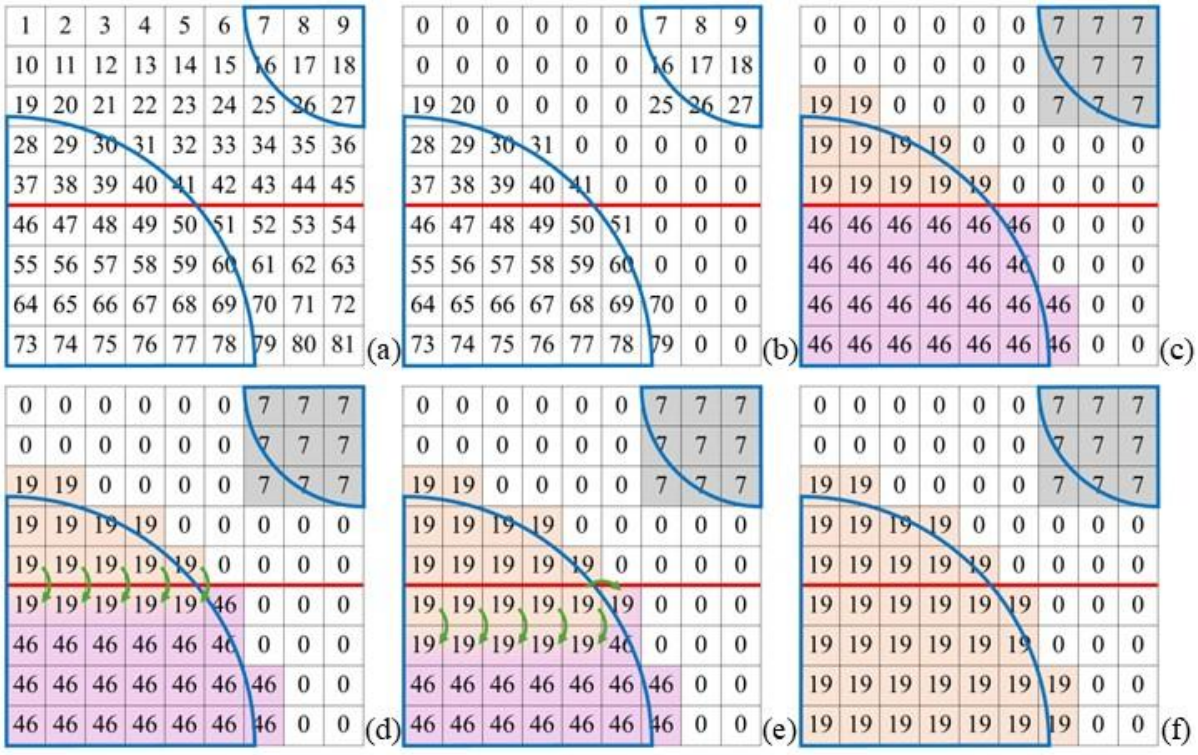


Figure 33. Visualization of the implemented Flood-fill algorithm. After the initial assignment of unique global indices to all cells (a), only cells with a secondary-phase volume fraction above the defined threshold retain their vofID field values (b). When the droplet spans multiple processors (c), the algorithm compares neighbouring cells across the interface and shares the lowest index value (d). During the subsequent iterations, this updated index is propagated through all connected cells (e) until full consistency is achieved (f).

3.3.1.3. dropG.H and dropVel.H

The purpose of these two modules is to determine, for each droplet identified in the domain, its center of mass, mass, and mean velocity. They operate sequentially:

- dropG.H evaluates the geometrical and physical characteristics of each droplet
- dropVel.H extracts the representative velocity from the flow field.

The information produced here is later used by the boundary condition to estimate dynamic effects and local force balances.

The routine begins by creating three data structures for every detected droplet: a vector storing the mass, a vector containing its center-of-mass coordinates, and a static vector used to retain the values from the previous time step to check label consistency between successive simulations. For each droplet ("dropIdx" define its identifier in the vofID field), the code loops through all computational cells. If a cell satisfies both conditions (i.e., its volume fraction alpha1 exceeds the liquid threshold and its vofID matches the label assigned to that droplet),

the cell is included in the droplet subdomain. The algorithm accumulates the coordinates of all such cells and computes the droplet's total mass as

$$m_{d,i} = \rho_{\text{water}} \sum_j V_j \, \alpha_j \tag{27}$$

where $\rho_{\text{water}}$is the liquid density, V_j is the cell volume, and α_j the local volume fraction. The number of contributing cells and the sum of their coordinates are both reduced across processors using OpenFOAM's collective operations, ensuring that all partitions contribute to a single consistent result. This synchronization step guarantees that droplets spanning multiple processor subdomains have the same total mass and centroid, regardless of mesh decomposition or load balancing. Once all contributions have been gathered, the center of mass is computed as the mean of the participating cell coordinates $N_{\text{cells,i}}$:

$$\mathrm{G}_{\text{liq,i}} = \frac{1}{N_{\text{cells,i}}} \sum_j \mathrm{x}_j \tag{28}$$

Both the total mass and the center-of-mass vector are then stored in the global arrays for later use.

The subsequent block, dropVel.H, reads the velocity field U from the solver database and computes, for each droplet, its mean velocity vector. The procedure mirrors the previous one, and finally we can write:

$$\mathrm{U}_{\text{mean,i}} = \frac{1}{N_{\text{cells,i}}} \sum_j \mathrm{U}_j \tag{29}$$

This average velocity serves as the representative motion of the droplet within the flow field and is later used to approximate the contact-line velocity and related dynamic effects.

Overall, these two routines provide the geometrical (position) and kinematic (velocity) descriptors of each droplet, ensuring that subsequent calculations are based on consistent, physically meaningful quantities even when the simulation is executed in parallel mode.

3.3.1.4. printGposition&velocity.H

This routine handles the writing of diagnostic output files during the simulation. Its purpose is purely to monitor the droplet evolution in real time, by recording the current time, the center-of-mass coordinates, and the corresponding mean velocity for each identified droplet. The writing operation is performed only by the master processor, ensuring that duplicated output is

avoided in parallel runs. A static variable keeps track of the last written time step, preventing redundant file access within the same output interval. Data is written only when the simulation time corresponds to an output time defined in the OpenFOAM control dictionary, or at the initial time step.

This simple yet essential mechanism provides a lightweight diagnostic output to track droplet motion and to verify that the labeling, tracking, and averaging routines are working correctly throughout the simulation.

3.3.1.4.1. forcesOnDroplet.H

The forcesOnDroplet.H module constitutes one of the key parts of the boundary condition, as it performs the evaluation of all the hydrodynamic forces acting on each identified droplet. The algorithm combines local field data from the multiphase simulation (such as the velocity and pressure gradients, the surface normal vectors, and the droplet geometry) to compute viscous, pressure, gravitational, and adhesion forces. The ratio between the total drag and the adhesion is then used to classify the droplet state, i.e. adhesion or sliding condition.

Interface Identification and Local Force Evaluation

The core of the routine loops over all computational cells, selecting only those that belong to the current droplet and that satisfy

$$\text{threshold} < \alpha < (1 - \text{threshold}) \tag{30}$$

thus, identifying the interfacial region between liquid and gas. For each found cell, the algorithm computes the local interface orientation from the gradient of the volume fraction field and reconstructs the cutting plane that approximates the gas–liquid surface within the cubic control volume. The plane is expressed in the general form

$$a\,x + b\,y + c\,z = D \tag{31}$$

where a, b, c are the components of the local normal vector to the interface, as per Eq. (15). D is the intercept determined iteratively through auxiliary functions (findInterfacePlane, findIntersection, and intersectionArea). These functions locate the proper value of D corresponding to the target local liquid fraction α, identify the intersection of the plane with the cube edges, and compute the interface area dA enclosed by the intersection polygon. Separate implementations handle the 2D and 3D cases, with internal consistency checks to flag incorrect intersection counts via warning files.

Once the local interface geometry is known, the code evaluates the hydrodynamic forces acting on the droplet at a small offset distance from the interface. This offset (equal to 3 cell lengths) allows sampling of flow quantities in the adjacent gas region, avoiding numerical noise at the interface.

The local viscous stress tensor is then computed as

$$\tau = \mu_{water} (\nabla U) \tag{32}$$

and the dynamic pressure is evaluated as

$$p_{dyn} = \frac{1}{2} \rho_{air} \ | U |^2 \tag{33}$$

The corresponding pressure drag and viscous drag contributions are:

$$dF_p = p_{dyn} n_i \, dA \tag{34}$$

$$dF_\mu = (\tau \cdot n_i) \, dA \tag{35}$$

These local forces are integrated over all interface cells belonging to the droplet, yielding the total pressure and viscous drag forces. Their sum represents the total aerodynamic drag acting on the droplet surface.

Note: the auxiliary functions findInterfacePlane, findIntersection, and intersectionArea are discussed in the Appendix B - Auxiliary Geometric Functions.

Adhesion and Gravitational Forces

To evaluate the adhesion force, the code estimates the contact area between the droplet and the wall by looping over all patch faces associated with the boundary condition. Each face belonging to the current droplet is assumed to have a square projection with area $A = l^2$, where l is the local cubic cell edge. The sum of these areas gives the total wetted surface A_{adh}, from which an equivalent circular diameter is derived:

$$D_{adh} = \sqrt{\frac{4A_{adh}}{\pi}} \tag{36}$$

The maximum adhesion force is then evaluated following the empirical correlation proposed by ElSherbini and Jacobi [84]:

$$F_{adh} = k_k \frac{24\,\sigma\,D_{adh}}{\pi^3}[\cos(\theta_{rec}) - \cos(\theta_{adv})] \quad (37)$$

where k_k is a correction factor. The need for this additional factor (not present in the cited source) arises from the fact that the employed correlation assumes an ellipsoidal contact area for the droplet. This area is uniquely defined in three-dimensional simulations; however, in two-dimensional cases, an infinite number of possible geometries can be derived from a single ellipse chord.

The gravitational force component acting along the streamwise direction is computed as:

$$F_g = m_d\, g_x \quad (38)$$

Finally, the force ratio

$$R = \frac{|F_{drag} + F_g|}{F_{adh}} \quad (39)$$

is used as a non-dimensional indicator of the droplet regime: if $R < 1$, adhesion dominates and the droplet remains pinned; if $R > 1$, drag exceeds adhesion and the droplet is expected to slide or roll.

3.3.1.5. printForces.H

This section handles the writing of diagnostic output files containing the instantaneous forces acting on each droplet during the simulation. As in the previous routine, execution is limited to the master processor to prevent redundant data output in parallel runs. A static variable is used to record the last written time step, ensuring that files are updated only once per output interval.

Beyond its role as a runtime diagnostic tool, this output also provides a convenient source of data for post-processing analyses, such as examining the temporal evolution of the total drag force, identifying the onset of sliding or detachment events, and so on.

3.3.1.6. lateralScaleFactor.H

This section introduces a local lateral scaling factor applied to the dynamic contact angle field in order to account for the geometric asymmetry of a droplet moving over a solid surface. In practical terms, this correction adjusts the effective contact angle along the droplet to reflect the natural difference between the advancing and receding regions of the front. Neglecting this effect would produce an unrealistic uniform angle distribution, with cusps at the lateral wall.

The implemented approach provides a linear correction of the local contact angle based on the projected orientation of the liquid-gas interface with respect to the main flow direction (here, the x-axis). Two scalar fields are introduced:

- factor: a scaling coefficient applied to the dynamic contact angle at each wall face;
- uwall: the local wall velocity, representing the instantaneous contact-line velocity for the corresponding portion of the droplet.

At each time step (except the first), the algorithm performs the following operations.

First, it searches for the maximum and minimum contact angles previously applied on the wall, corresponding respectively to the advancing and receding regions of the droplet. These extreme values are then used to compute the maximum and minimum x-components of the interface normal vector, which serve as geometric references for the scaling.

For each wall cell belonging to a given droplet:

- If the x-component of the interface normal is negative, the local interface is classified as advancing, meaning the liquid front is entering the cell volume. The scaling factor is thus computed as

$$\text{factor} = |n_x / n_{x,\max}| \tag{40}$$

- Conversely, if $n_x > 0$, the interface is receding, and

$$\text{factor} = |n_x / n_{x,\min}| \tag{41}$$

To avoid unphysical amplification, the scaling factor is limited to unity (factor ≤ 1).

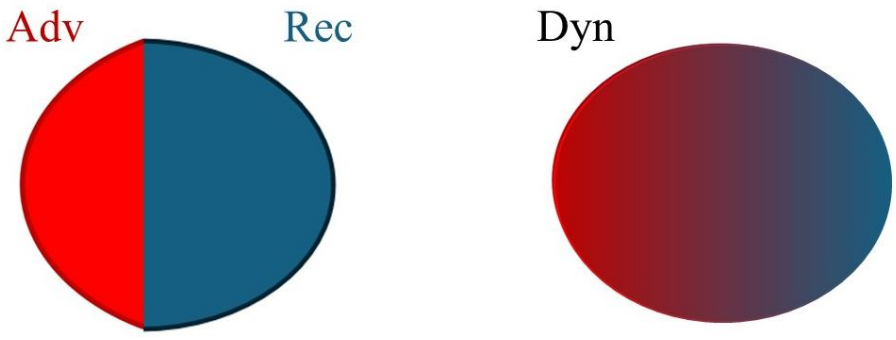


Figure 34. To avoid the presence of cusps at the contact line (left), a linear scalar factor is implemented to smooth out the trend from advancing-receding angles (right)

This formulation effectively introduces a spatial modulation of the contact angle, allowing the model to reproduce the asymmetry observed during droplet motion. Although the trend is presently linear, the structure of the routine is sufficiently general to accommodate more complex, nonlinear dependencies in future developments.

3.3.1.7. contactAngleHysteresis.H

This routine defines the local dynamic contact angle by accounting for contact angle hysteresis and for the balance between adhesion and drag forces acting on each droplet. The procedure bridges the gap between the purely geometric definition of the contact angle and its physical dependence on the droplet's motion, enabling the model to continuously transition between static, advancing, and receding regimes.

At the beginning of the routine, a temporary scalar field is initialized to the equilibrium angle and used as the output field for the boundary condition. An auxiliary vector of Boolean flags tracks whether each droplet is already in motion, allowing the solver to distinguish between droplets in a static adhesion state and those undergoing sliding.

For each identified droplet, the code evaluates the ratio between total drag and adhesion forces. If this ratio is less than or equal to one, the adhesion force dominates, and the droplet remains attached to the surface. In this condition, two text files are generated by the master processor:

- Adhesion_Check_i.txt, which acts as a flag for the solver to indicate that the droplet is in the adhesion regime;
- dropDeceleration_i.txt, which stores the instantaneous deceleration resulting from the imbalance between drag and adhesion forces.

If the droplet was previously sliding, the computed deceleration value is written to the file; otherwise, a zero value is recorded, preventing spurious motion of droplets that remain pinned.

For all wall faces belonging to the droplet, the local contact angle is then computed as a linear function of the droplet's velocity and force ratio R, following a simplified hysteresis law:

$$\theta_{dyn} = \begin{cases} \theta_0 + R \cdot (\theta_A - \theta_0) \cdot \text{factor}, & \text{if } u_{wall} < 0 (\text{advancing front}) \\ \theta_0 - R \cdot (\theta_0 - \theta_R) \cdot \text{factor}, & \text{if } u_{wall} \geq 0 (\text{receding front}) \end{cases} \quad (42)$$

where:

- θ_0 is the static contact angle,
- θ_{adv} and θ_{rec} are the advancing and receding limits, respectively,
- ratio represents the balance between drag and adhesion forces,

- factor is the geometric scaling parameter previously computed.

This formulation ensures that the local angle evolves smoothly between its limiting values as a function of both hydrodynamic forcing and interface geometry. At the end of the loop, the computed angle field is stored to be used in the next timestep.

Overall, this section provides the core physical closure of the boundary condition: it dynamically links the droplet's motion, force balance, and contact-line geometry into a unified, time-resolved model for wettability hysteresis.

3.3.1.8. contactAngleCorrelations.H

This final section of the boundary condition defines the dynamic contact angle correlations used when the droplet transitions from the adhesion regime to actual sliding motion. Once the drag-to-adhesion ratio exceeds unity (R > 1), the adhesion force is no longer sufficient to pin the droplet, and the dynamic evolution of the contact angle becomes governed primarily by hydrodynamic effects. To capture this behaviour, the model employs semi-empirical correlations that relate the dynamic contact angle to the local capillary number Ca, which quantifies the relative importance of viscous and capillary forces. The capillary number is computed for every face on the wall patch, providing a local measure of interfacial dynamics.

Two regimes are then distinguished depending on the sign of the wall velocity U_{cl}:

- Advancing front ($U_{cl} < 0$)

  The contact angle increases as the droplet edge advances over the dry surface. In this case, the Kistler correlation is used, a semi-empirical formulation derived from the Hoffman function, which relates the advancing angle to Ca through a non-linear hyperbolic tangent expression. The local value is then scaled linearly using the geometric factor previously defined, according to:

$$\theta_{dyn} = \theta_0 + (\theta_A^* - \theta_0) \cdot \text{factor} \quad (43)$$

- Receding front ($U_{cl} \geq 0$)

  The local contact angle decreases as the liquid front retracts. The model adopts a cubic relation based on the Cox–Voinov law, where the cube of the contact angle is linearly related to the capillary number. The updated contact angle is computed as:

$$\theta_{dyn} = \theta_0 - (\theta_0 - \theta_R^*) \cdot \text{factor} \quad (44)$$

θ_A^* and θ_R^* are the contact angle evaluated following the experimental correlation already described in chapter 3.2.3 (see Eq. 23 to 26).

At the end of the loop, the flag is set to true for the corresponding droplet, indicating that it has entered the sliding regime and will henceforth follow the hydrodynamic correlation rather than the hysteresis-based model.

This final step therefore completes the physical closure of the dynamic contact-angle boundary condition. The implementation allows the contact angle to evolve continuously and consistently from the static (adhesion) condition to the fully dynamic regime, reproducing both advancing and receding behaviours observed experimentally for droplets under motion.

3.3.2. Solver modification

The dynamic contact-angle boundary condition developed for this work requires a minimal yet essential modification to the main interFoam solver. Specifically, this addition enables a two-way coupling between the solver and the custom boundary, allowing the contact-angle model to directly influence the droplet motion during runtime.

The new code segment, inserted after the solution of the momentum and volume-fraction equations, performs three main tasks:

1. Read droplet information generated by the boundary.
   The boundary condition writes several auxiliary files during each output step:
   - dropNumber.txt, the number of droplets currently detected in the domain;
   - droplet_i_label.txt, an integer label identifying each droplet within the volume fraction field (vofID);
   - Adhesion_Check_i.txt and dropDeceleration_i.txt, diagnostic files specifying whether a droplet is in adhesion or sliding condition, and its corresponding deceleration value.

   These files are read at each iteration of the solver to ensure synchronization between the physical model (evaluated in the boundary) and the flow field evolution.
2. Adjust the velocity field according to droplet state.
   Once the information is imported, the solver iterates over all detected droplets.
   - If the adhesion flag file is found and the computed deceleration is zero, the droplet is considered pinned to the surface. In this case, the velocity of all fluid cells belonging to that droplet at the boundary is set to zero, effectively freezing its motion while maintaining the correct interfacial shape.
   - Conversely, if a nonzero deceleration is read, the droplet is in a sliding condition. The solver then applies a deceleration term to the velocity field along the direction of motion (here, the x-axis), scaling it by the simulation time step.

This mimics the retarding effect of adhesive forces during partial sliding, providing a physically realistic slowdown of the droplet's center of mass.

3. Clean temporary data
   At the end of each iteration, the solver removes the auxiliary vofID field from non-output timesteps to prevent unnecessary storage and maintain consistency across parallel runs.

This additional routine establishes a feedback loop between the boundary-level contact-angle dynamics and the bulk momentum equations of the solver. By dynamically freezing or decelerating the droplet motion based on adhesion strength, the solver can accurately reproduce both static pinning and partial-sliding regimes, without requiring external intervention or user-defined motion constraints.

3.3.3. Results and validation

The first test performed after completing the implementation of the boundary condition concerns the deposition of water droplets on surfaces with different wettability. This preliminary validation aims to ensure that the boundary does not introduce unintended effects and that it does not alter the prescribed static contact angle in the absence of any aerodynamic forces (i.e., when the droplet is at rest and not exposed to airflow). As shown in Figure 35, the same droplet with a volume of 0.02 mm^3 was deposited on three surfaces with increasing hydrophobic behaviour, in order to verify whether the imposed static contact angle was correctly reproduced.

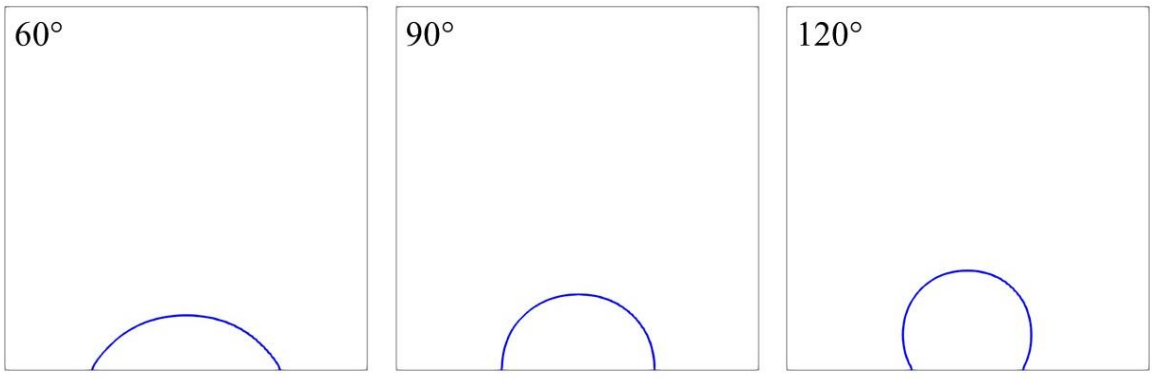


Figure 35. The same 0.02 mm^3 droplet was deposited on surface with increasing hydrophobic behaviour

Since one of the primary objectives of developing this code was to capture the mutual aerodynamic interaction between droplets (e.g., the shielding effect of upstream droplets on downstream ones within PEMFC channels) two droplets were placed on the BP with a static contact angle of approximately 50°. The system was then exposed to a constant airflow of 15 m/s. The deposited droplets correspond to classes B600 and B400 (as defined in Figure 28),

initially separated by 1 mm. The same configuration was simulated using the updated solver and boundary condition. Figure 36 shows that the evolution of the distance between the two centres of mass is satisfactorily predicted by the model, in good agreement with experimental observations. It is worth mentioning that, at this step, the experimental apparatus and data processing is the same described in 3.1.1.

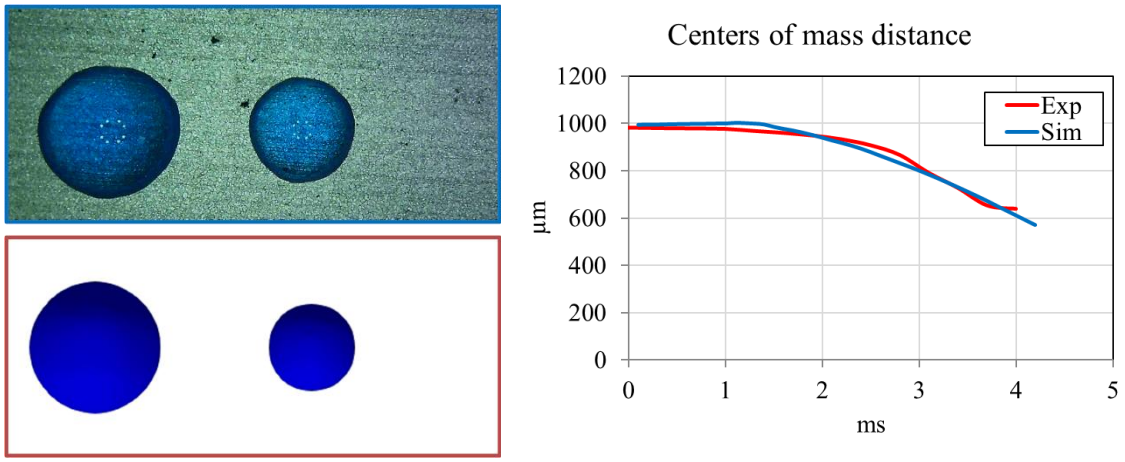


Figure 36. A B600 and B400 droplet were deposited at 1 mm distance (left), and their centers of masses distance was monitored (right); simulation predictions align quite well with experimental visualization.

A further validation step involved comparing the computed drag forces with analytical correlations available in the literature. At this point, a clarification is required. The literature presents a wide variety of drag models, applicable to a broad range of particle sizes, flow regimes, and interface shapes. Specifically, for droplet removal modelling in PEMFC channels, Cho et al. [85] performed a theoretical and numerical study of a single droplet within a rectangular channel, showing that drag increases with the interaction with the core flow as the droplet height or equivalent diameter increases, and that the detachment mechanism depends on both the inlet location and flow development regime. They provide useful relationships and scaling laws with U^2, channel height H, and droplet size. Similarly, Li et al. [86] conducted an ex-situ investigation on the effect of channel height and contact-angle hysteresis, showing that reducing H facilitates detachment by lowering the effective hysteresis at the onset of motion, producing valuable H–U–θ maps.

However, a common limitation among these “geometric” models lies in the definition of an equivalent droplet diameter, which is straightforward for droplets on highly hydrophobic GDLs (with contact angles around 120°), but becomes impractical for droplets adhering to the side walls of the channel. In these cases, the droplet is essentially spread along the wall, forming a spherical cap, and thus lacks a clear geometric definition of its equivalent diameter. To overcome this limitation, the interface reconstruction method previously described was employed, enabling the direct evaluation of drag and adhesion forces acting on the droplet.

To further assess the boundary behaviour, the numerical output was compared with the general expression for the drag force:

$$F_{drag} = \frac{1}{2} \cdot \rho \cdot U^2 \cdot C_D \cdot A \quad (45)$$

where C_D and A are the drag coefficient and the frontal area of the obstacle, respectively. Using this formulation for liquid droplets poses two main challenges:

1. The drag coefficient $C_D$is not constant but varies with the Reynolds number and the droplet shape, which changes as deformation occurs.
2. The droplet shape also affects the effective frontal area exposed to the flow.

To address these issues, the drag force experienced by a droplet placed at the centre of a cubic domain (3×3×3 mm³) was computed under a steady airflow, assuming that the exposed area remained approximately constant, equal to the frontal area of a sphere of equivalent diameter. The value of $C_D$was taken from the literature (see Figure 37, adapted from Schlichting and Gersten [87]) and set to a constant 0.5, which is consistent with the velocity range considered.

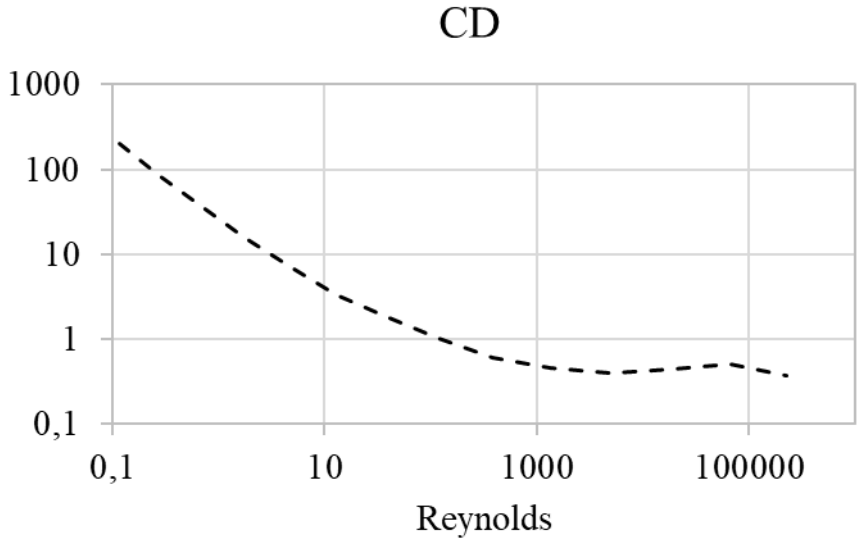


Figure 37. Drag coefficient over Reynolds number, adapted from Schlichting and Gersten [87]

Simulations were carried out for droplets of 300 μm and 400 μm in diameter, subjected to constant airflows of 7.5 m/s, 15 m/s, and 20 m/s, after an initial rest period of 0.1 ms and for a total of 2 ms.

Figure 38 reports the computed results organized by velocity classes. The dashed lines correspond to the theoretical drag forces evaluated using Eq. 45, while the solid lines represent the forces estimated through the implemented numerical algorithm. The black curve refers to the 300 μm droplet, and the grey curve to the 400 μm droplet.

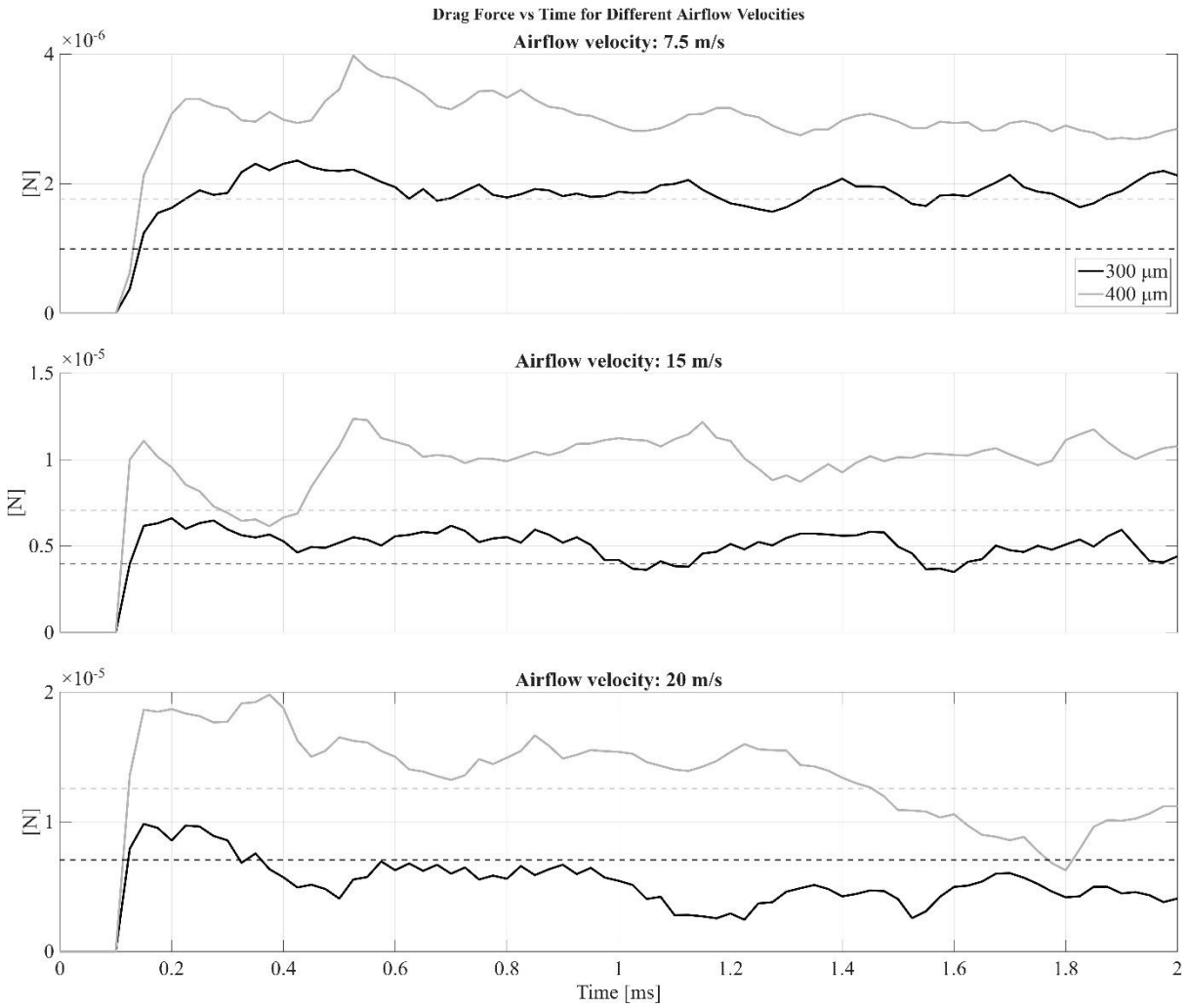


Figure 38. Comparison between boundary evaluated drag force and Eq. 45

A first observation is that the predicted drag forces exhibit a consistent agreement with the theoretical trend, with discrepancies that remain within a physically acceptable range. The overall magnitude and scaling with velocity are accurately reproduced, confirming the soundness of the numerical formulation and its correct integration within the solver. The deviations observed between the computed and theoretical curves can be attributed to several factors.

First, the comparison is performed assuming a constant drag coefficient C_D across all test conditions. In reality, the drag coefficient depends on the Reynolds number, which varies with droplet size and flow velocity; at lower velocities, the reduced Reynolds number would correspond to slightly higher C_D values, leading to an overestimation of the analytical drag.

Second, the droplet deformation induced by the aerodynamic load is not included in the analytical expression. Under the imposed airflow, the initially spherical droplet elongates in the flow direction, effectively altering both the frontal area and the corresponding C_D. These geometric variations, although moderate, are sufficient to explain the observed divergence from the idealized model.

Moreover, the numerical simulations capture unsteady fluctuations in the drag signal. Such oscillations originate from vortex shedding and wake instabilities developing downstream of the droplet. The interaction between these local recirculation regions and the liquid-air interface periodically modifies the effective aerodynamic load, producing the oscillatory behaviour visible. Overall, the comparison demonstrates that the implemented force reconstruction algorithm is capable of reproducing both the average magnitude and the physical variability of the drag force. Despite the simplifying assumptions, the predicted values remain within a realistic range, validating the robustness and reliability of the developed boundary condition.

Particular effort was then dedicated to assessing the reliability of the adhesion formulation. A practical benchmark for this purpose consists in reproducing the dynamics of a sessile droplet placed on an inclined surface, where gravity represents the only external driving force. Previous experimental investigations (e.g., Podgorski [88]; Le Grand et al. [89]) have demonstrated that droplet motion on tilted substrates is governed by a threshold inclination angle. This critical angle is reached when the gravitational component parallel to the surface overcomes the contact-angle hysteresis interval ($\theta_{adv}-\theta_{rec}$), leading to depinning and subsequent sliding.

The droplet examined in this study has a constant volume of 6 mm³ (6 µL), whereas its physical properties depend on the specific silicone oil reported in literature data. The numerical domain is represented by a rectangular box measuring 6.75 × 2.25 × 2.25 mm³. One boundary corresponds to the inclined wall, another represents a symmetry plane that divides the droplet into two identical halves, while the remaining faces are treated as periodic boundaries in order to reduce computational requirements. The substrate rotation is imposed gradually so that the droplet behaviour can be regarded as quasi-static. After deposition, the droplet is allowed to relax for 0.05 s before the inclination is progressively increased according to

$$\alpha(t) = \pi \frac{(t-t_0)}{T} \tag{46}$$

where $t_0$denotes the initial relaxation time and $T = 0.5$s represents the rotation duration. To describe the droplet geometry, the volume of an equivalent spherical droplet with radius a,

$$V = \frac{4\pi a^3}{3} \tag{47}$$

is equated to the volume of the spherical cap formed once the droplet reaches equilibrium on the solid surface,

$$V = \frac{\pi H^2(3R-H)}{3} \tag{48}$$

Figure 39 schematically illustrates both the initially deposited droplet and the resulting spherical-cap configuration after wetting equilibrium is established.

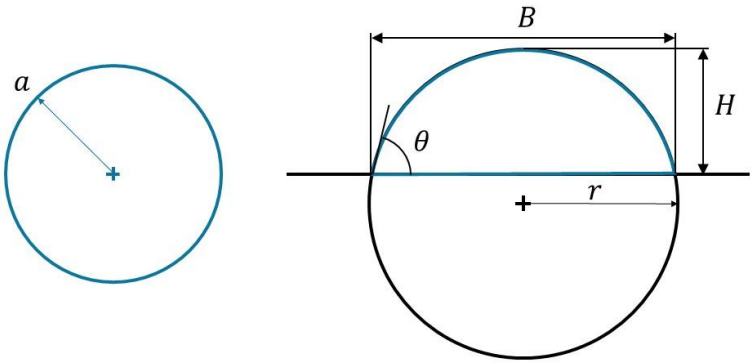


Figure 39. Initial drop has a spherical shape (left); once it is deposited on a surface with given contact angle, it reaches a stable condition as a spherical cap (right)

By combining the above relationship with the geometrical identities associated with spherical caps characterized by contact angle θ, the wetted diameter can be derived as

$$B = 2R\sin\,\theta = 2\sin\,\theta\sqrt[3]{\frac{4a^3}{3(1-\cos\,\theta)^2-(1-\cos\,\theta)^3}} \quad (49)$$

This parameter represents the effective width of the adhesion region. Under quasi-static assumptions, it is reasonable to approximate the adhesion diameter D_{adh} appearing with the wetted diameter B (see Figure 40).

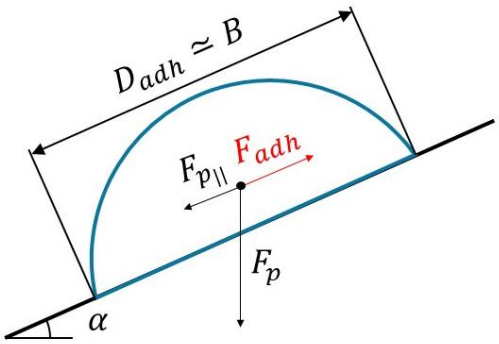


Figure 40. Forces acting on a droplet deposited over an inclined plane

Droplet motion is then initiated when the tangential component of the gravitational force balances the resisting adhesion force, namely

$$\rho V g\sin\,\alpha_{cr} = \frac{24}{\pi^3}\sigma B(\cos\,\theta_{rec} - \cos\,\theta_{adv}) \quad (50)$$

Rearranging this expression provides the following relation for the critical inclination angle:

$$\sin\ \alpha_{cr} = \frac{24a^2B}{\pi^3 V}(\cos\ \theta_{rec} - \cos\ \theta_{adv})Eo^{-1} \quad (51)$$

Here,

$$Eo = \frac{\rho g a^2}{\sigma} \quad (52)$$

denotes the Eötvös number, which quantifies the relative contribution of gravitational and capillary forces.

The coefficient $\frac{24a^2B}{\pi^3 V}$ depends exclusively on droplet size (through a and V) and surface wettability (through B, hence θ). For a droplet volume of 6 mm³ and an equilibrium contact angle of 50°, this factor is approximately equal to 0.64, which is consistent with the value of about 0.6 reported by Maglio and Legendre [90]. Consequently, when $(\cos\ \theta_{rec} - \cos\ \theta_{adv})Eo^{-1} \gtrsim 1.55$, droplet sliding cannot occur at any inclination angle, as either hysteresis effects or capillary forces dominate over gravitational driving.

The experimental datasets used for model validation are summarised in Table 1, whilst Figure 41 compares experimental measurements from Podgorski [88] and Le Grand et al. [89] with both CFD simulations and the analytical correlation developed in this work. The proposed model captures the overall trend of the experimental observations with satisfactory accuracy, although a slight overestimation of the critical inclination angle is detected. This discrepancy is likely related to the approximation $D_{adh} \approx B$, which becomes less reliable in the presence of strongly deformed droplets characterised by large hysteresis.

Table 1. Literature data [88,89] for adhesion force model validation

Fluid	Viscosity [cP]	Density [kg/m3]	Surface tension [mN/m]	Equilibrium [deg]	Advancing [deg]	Receding [deg]
47V10	10	936	20,1	50	50,5	45,5
47V100	103,7	964	20,9	50	52,9	42,7
47V1000	1035	970	21,1	50	58	47

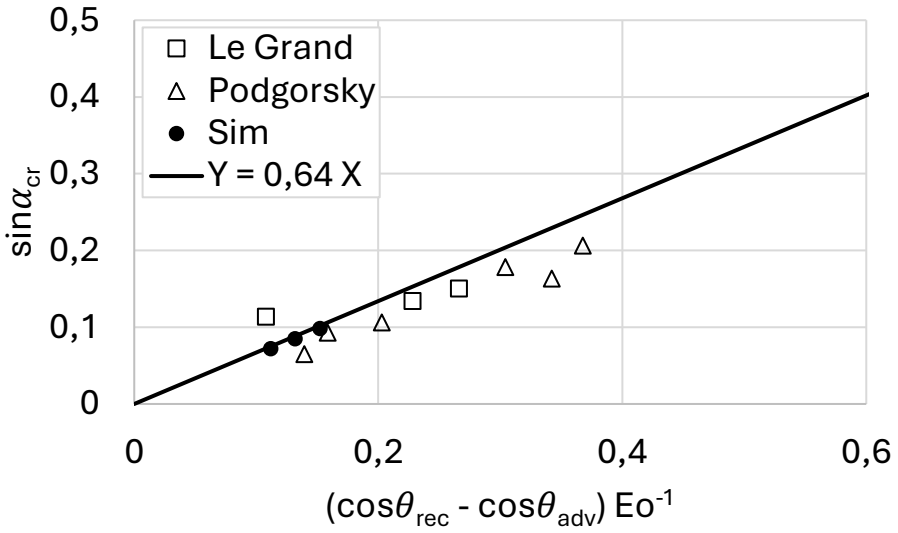


Figure 41. Experimental data from Podgorsky [88] and Le Grand et al. [89], numerical simulations and analytical trend from Eq. 51

3.3.3.1. Single drop

The simulations were carried out in a three-dimensional rectangular domain of 7 × 1 × 1 mm³, representing a simplified airflow channel section. The computational mesh is orthogonal, consistent with the current boundary implementation requirements. Air enters the domain through a velocity inlet and exits through a pressure outlet. A uniform inlet velocity of 15 m/s is imposed after an initial 1 ms stabilization phase at zero velocity, allowing the droplets to reach equilibrium before flow onset. All walls are treated as no-slip boundaries, and the bottom surface hosts the droplet deposition sites, located 3 mm downstream of the inlet. Gravity is considered, even though negligible with respect to involved aerodynamics forces for the dimension of selected droplets.

Three droplets (B400, B500, and B600) are considered, corresponding to increasing liquid volumes (0.008 µL, 0.016 µL, and 0.027 µL, respectively). The initial static contact angle is 50°, with advancing and receding angles of 55° and 45°, respectively. The results are compared with an equivalent set of droplets (B400eq, B500eq, B600eq) placed on a more hydrophobic surface (120°, 125°, 115°) but maintaining the same liquid volumes, to assess the influence of wettability on droplet motion and aerodynamic drag.

Figure 42 shows the temporal evolution of the droplet center of mass (top) and the total drag force in the longitudinal direction (bottom). For the more hydrophilic surface, the smallest droplet (B400) remains pinned throughout the simulation since the aerodynamic drag never overcomes the adhesion force. It deforms slightly in the direction of the airflow but it does not move downstream. Conversely, larger droplets (B500 and B600) exhibit progressive displacement once drag exceeds adhesion. After the transient phase associated with flow onset, the drag force stabilizes, and its magnitude scales consistently with the droplet size and exposed frontal area, confirming the correct force balance predicted by the boundary model. The

comparison with Figure 43 highlights the differences when looking at an hydrophobic surface. The higher contact angle results in a reduced wetted area and weaker adhesion, while the "elevated" droplet shape increases frontal exposure to the airflow. As a consequence, all droplets experience higher drag and faster motion, reaching the outlet (x = 7 mm) within a few milliseconds. The stars in the plots indicate the moment each droplet reaches the exit boundary. These results highlight how the implemented force-based dynamic contact angle formulation accurately reproduces the coupled influence of wettability and droplet geometry on aerodynamic response.

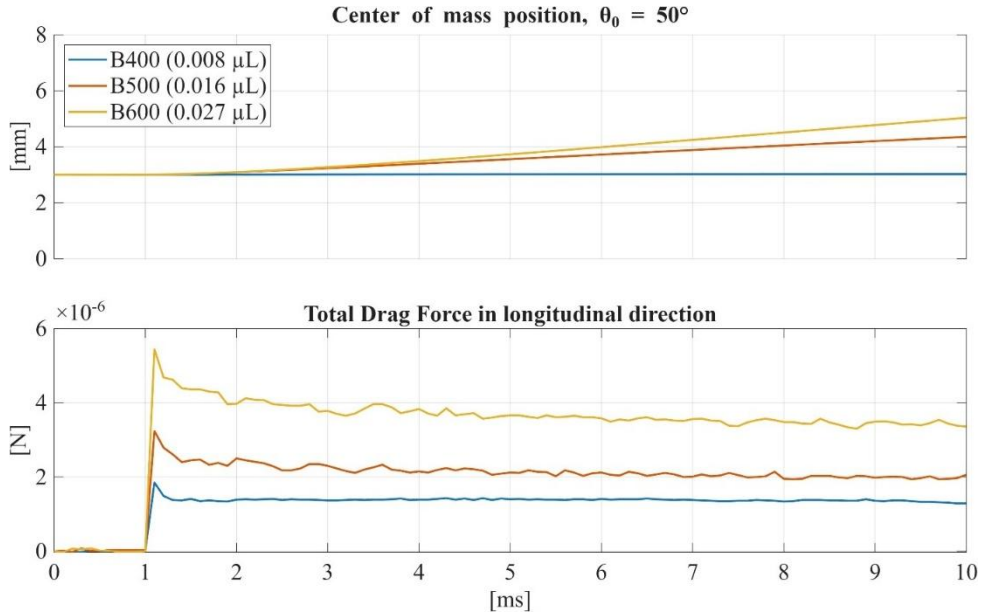


Figure 42. Center of mass position (top) and longitudinal drag force (bottom) for droplets B400, B500, and B600 on a hydrophilic surface under a 15 m/s airflow

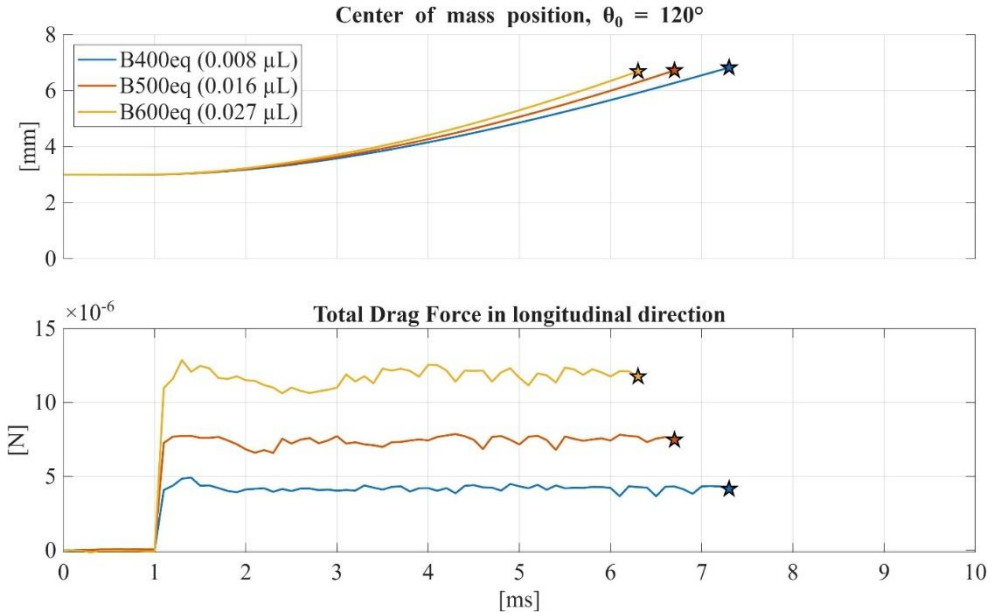


Figure 43. Center of mass position (top) and longitudinal drag force (bottom) for droplets B400eq, B500eq, and B600eq on a hydrophobic surface under a 15 m/s airflow

Figure 44 presents the temporal evolution of the droplet interface through isosurfaces at α = 0.5 for the hydrophilic case, allowing a direct visualization of deformation patterns and contact line behaviour. Figure 45 reports the corresponding evolution for the hydrophobic surface. In this latter case, the final frame is shown at t = 6.4 ms, corresponding to the instant at which the largest droplet reaches the outlet, as previously indicated by the yellow star in Figure 43. The comparison between the two figures further emphasizes the role of wettability in governing droplet spreading, elongation, and detachment dynamics under identical aerodynamic conditions.

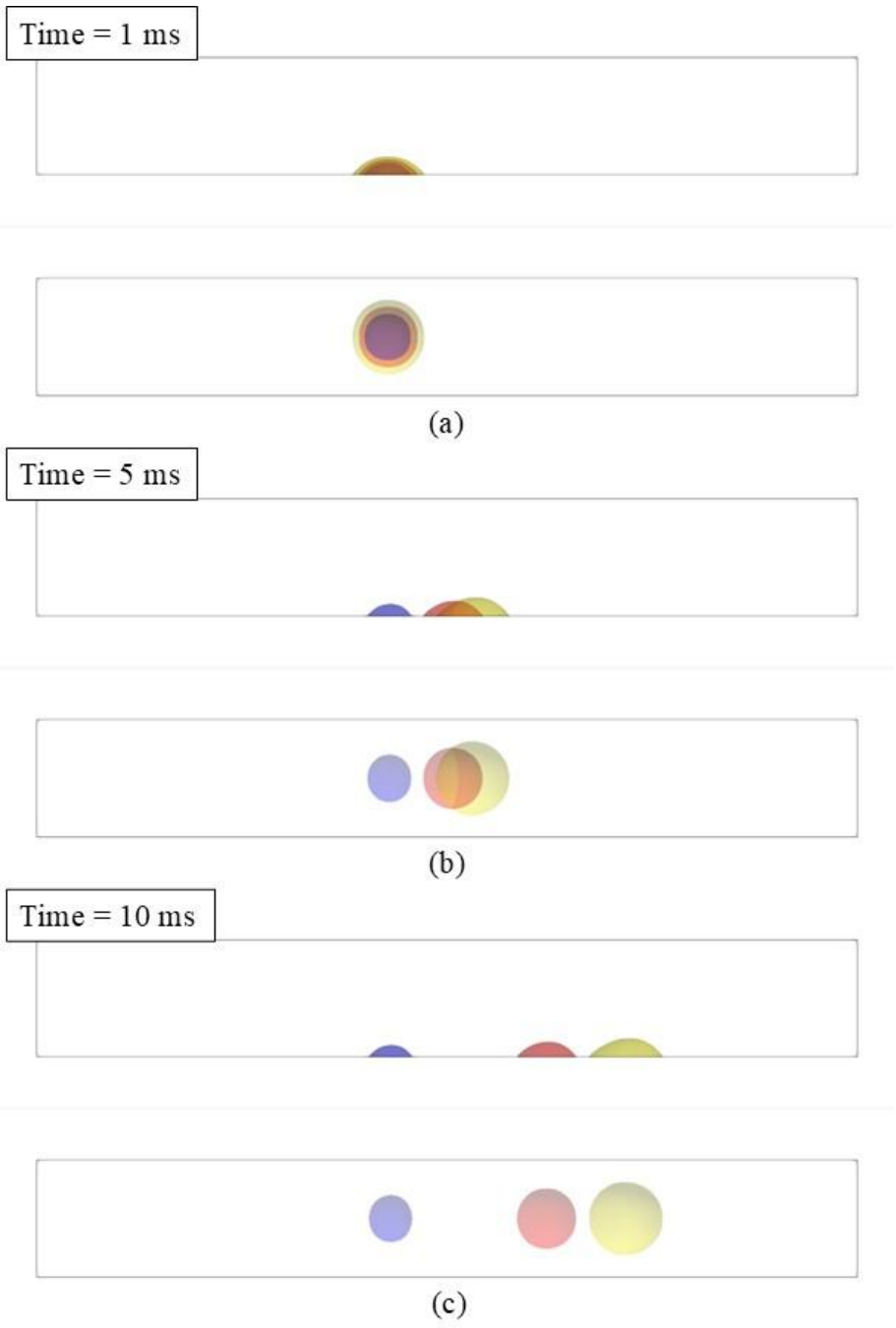


Figure 44. Front and top view of isosurfaces (α = 0.5) for a B400 (blue), B500 (red) and B600 (yellow) droplet deposited on an hydrophilic surface at different timestamp

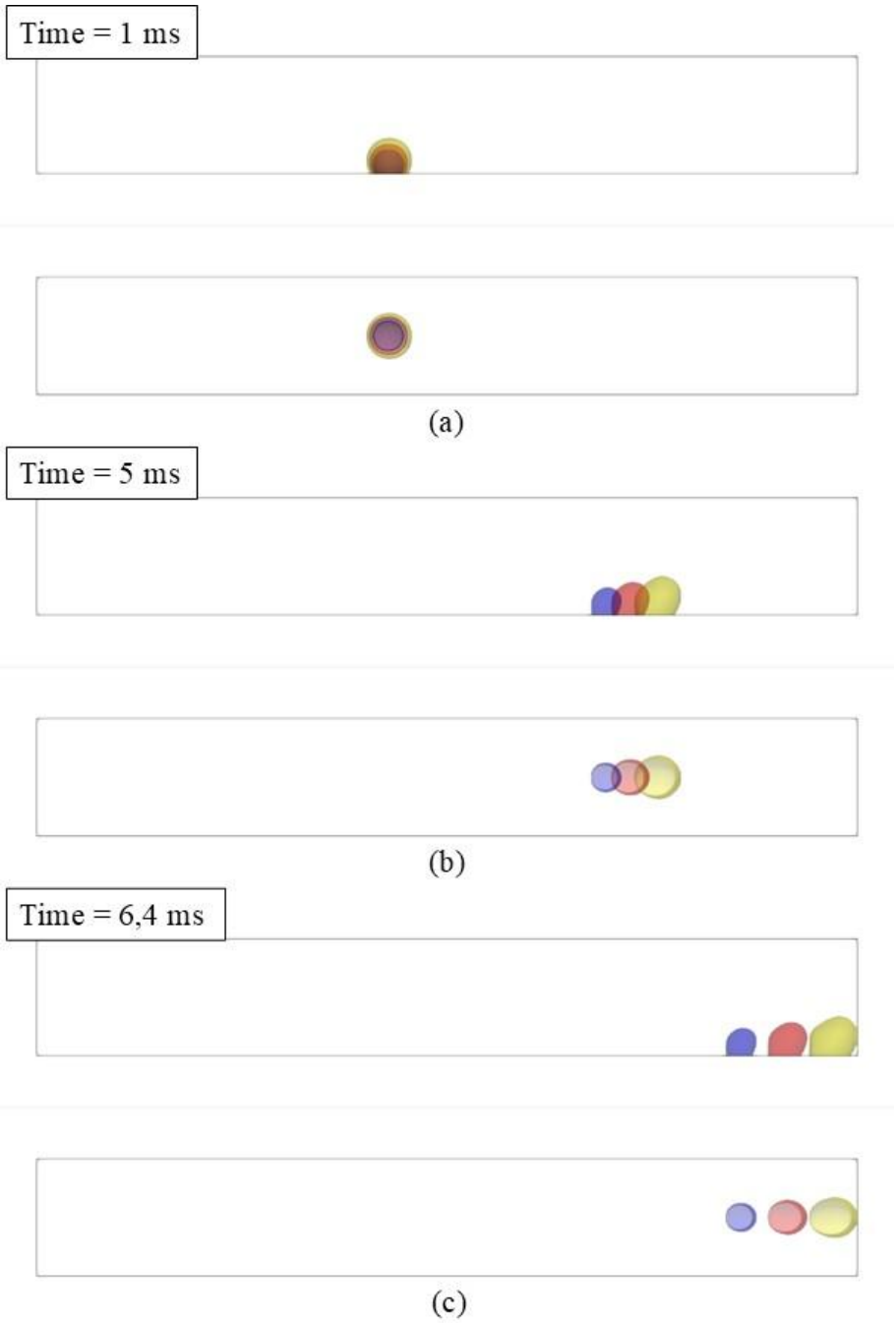


Figure 45. Front and top view of isosurfaces (α = 0.5) for a B400eq (blue), B500eq (red) and B600eq (yellow) droplet deposited on an hydrophobic surface at different timestamp

3.3.3.2. Double drop

To further assess the predictive capabilities of the force-based boundary condition, additional simulations were performed with two interacting droplets under the same flow conditions described previously. The computational domain and boundary conditions remain unchanged. The first droplet is always deposited on the wall at x = 3 mm, while the second droplet is positioned downstream at increasing center-to-center distances.

The parametric study considers different configurations, varying (i) the inter-droplet distance, (ii) the droplet size order (large–small vs. small–large), and (iii) a possible lateral offset between droplets, all phenomena that realistically occurs in PEMFC channels. In all following plots, the circle marks the coalescence event between the two droplets, while the star indicates the exit of the merged droplet from the computational domain. Figure 46 shows a scheme of the possible simulated layout.

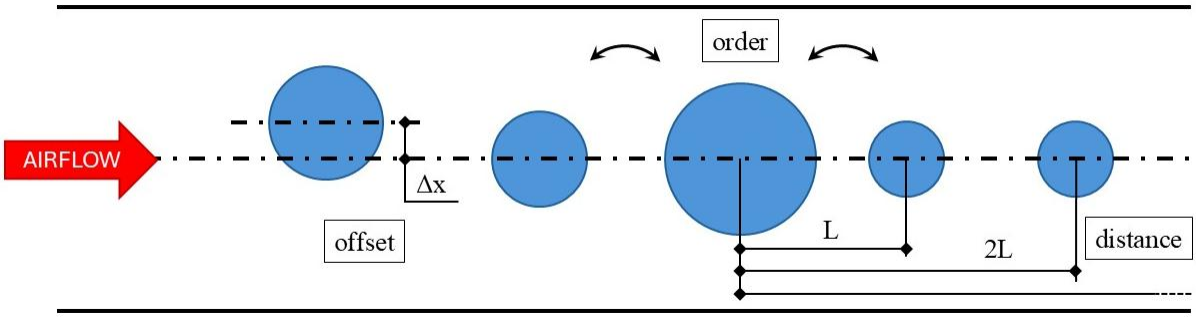


Figure 46. The parametric study investigated the reciprocal distance (right), the droplet order and shielding effect (middle) and offset (left)

Figure 47 reports the time evolution of the total longitudinal drag force for a pair of identical B600 droplets at three initial spacings. The solid lines correspond to the leading droplet, directly exposed to the airflow, while the dashed lines represent the trailing droplet partially shielded by the upstream one. At short distance (D = 0.8 mm), the rear droplet experiences a strong reduction in aerodynamic load due to the wake generated by the first droplet. The resulting velocity difference leads to rapid coalescence around the 2.9 ms mark, followed by a transient rise in drag as the merged droplet accelerates and moves downstream. For larger separations (D = 1.6-2.4 mm), the shielding effect weakens, the coalescence is delayed, and the drag profiles stabilize at lower values with limited oscillations. The star symbols denote the instant when the droplets (or merged droplet) exit the domain.

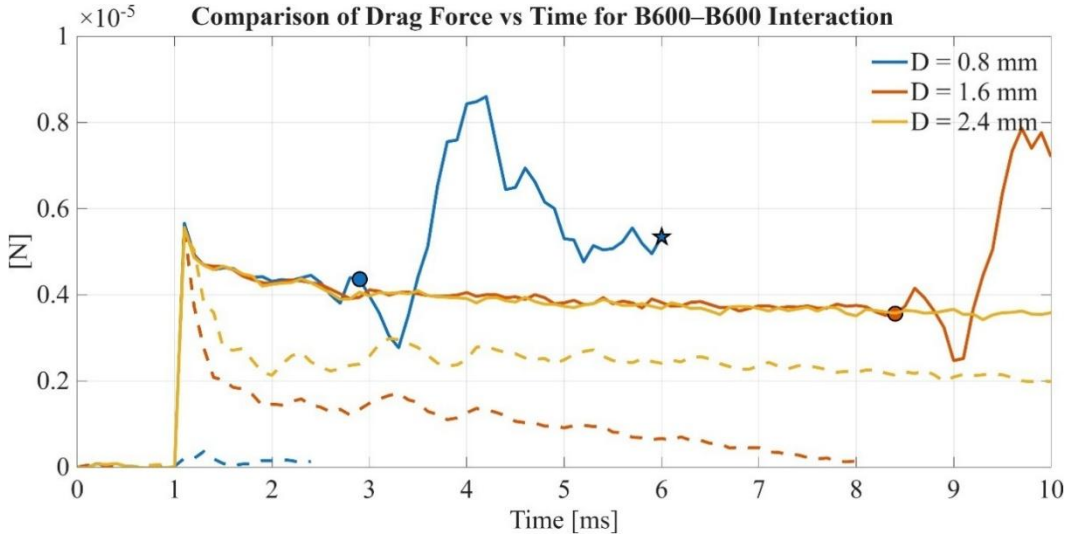


Figure 47. Longitudinal drag force for B600-B600 droplets pair on a hydrophilic surface under a 15 m/s airflow

The second configuration investigates the interaction between two droplets of different sizes, B600eq and B400eq, deposited on a more hydrophobic surface. The droplets maintain the same liquid volumes as the previous hydrophilic cases but exhibit higher mobility due to reduced adhesion and a more elevated shape. Figure 48 illustrates the longitudinal drag evolution for three initial spacings. The larger leading droplet (solid lines) dominates the flow field, producing a pronounced wake region that substantially lowers the drag acting on the trailing, smaller droplet (dashed lines). For D = 1.0 mm, this wake shielding quickly induces coalescence between the two droplets (blue circle), followed by a sharp drag increase as the merged droplet accelerates downstream and eventually exits the domain (blue star). At greater distance, the interaction weakens; however the smaller droplet is still affected and experience and oscillating and overall reducing drag force.

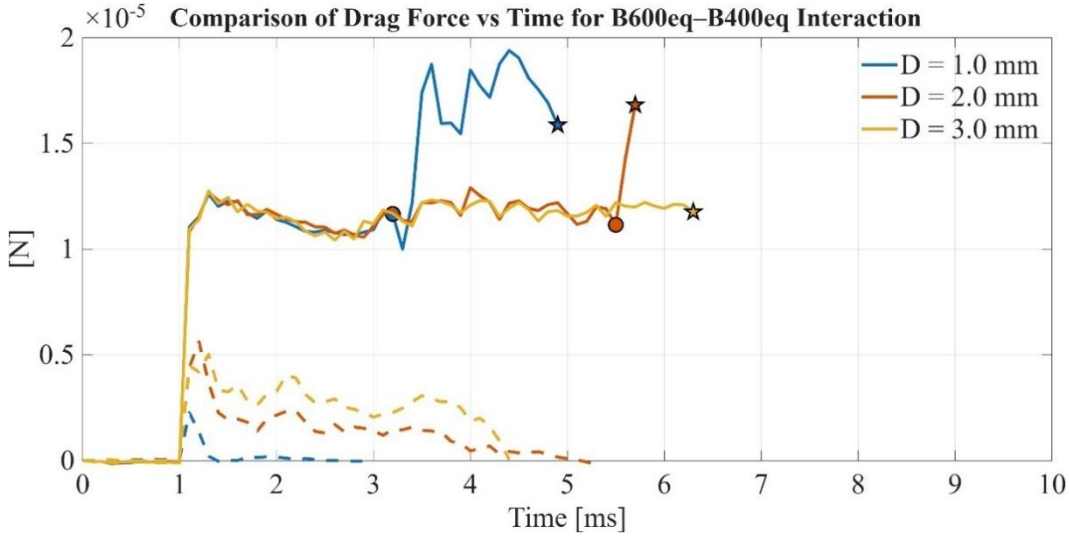


Figure 48. Longitudinal drag force for B600-B600 droplets pair on a hydrophobic surface under a 15 m/s airflow

To assess the influence of droplet sequencing, two configurations were simulated with the same droplet pair (B400 and B600) but opposite deposition order. Both cases were subjected to the same 15 m/s airflow and inter-droplet spacings of D = 1.0, 2.0, and 3.0 mm. As shown Figure 49, when the smaller droplet (B400) precedes the larger one (B600) (top plot), the upstream droplet generates only a limited wake, insufficient to strongly alter the aerodynamic field acting on the trailing, larger droplet. The drag on the downstream droplet (solid lines) thus remains relatively stable, while the smaller leading droplet (dashed lines) experiences a rapid stabilization after the initial transient. Only for the smallest spacing (D = 1.0 mm) does partial interaction occur; the other cases basically overlap with the single droplet layout.

The opposite configuration (B600-B400, bottom plot) presents a markedly different behaviour. Here, the large leading droplet strongly shields the smaller one, reducing its aerodynamic load by nearly 50%. The resulting velocity gradient quickly induces coalescence at short separations (D = 1.0–2.0 mm, blue and red circles respectively), followed by a characteristic drag spike associated with droplet merging. At larger spacing (D = 3.0 mm), the interaction weakens, but the general downwards trend for B400 droplet is still clear.

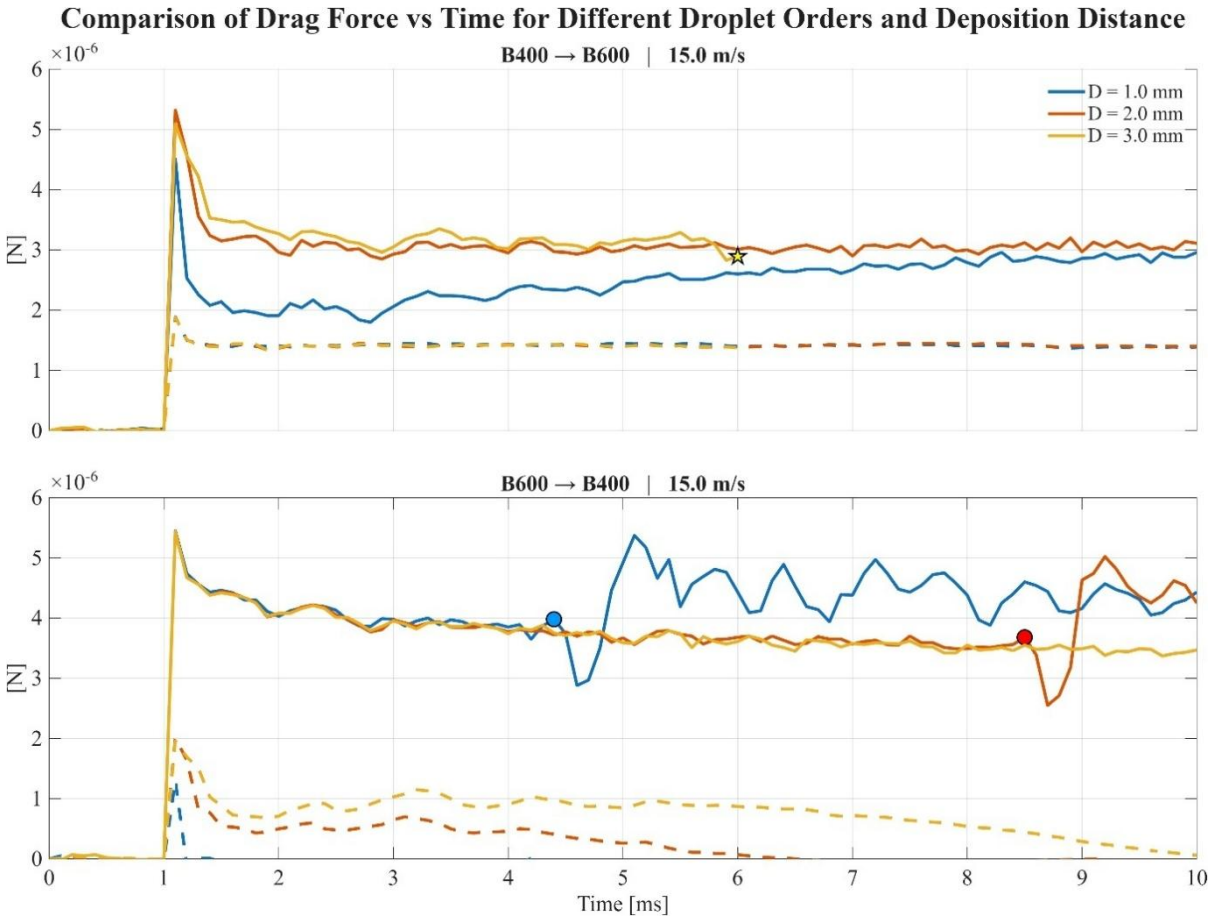


Figure 49. Longitudinal drag force for B400-B600 (top) and B600-B400 (bottom) droplets pair on a hydrophilic surface under a 15 m/s airflow

In realistic PEMFC channels, droplets are rarely aligned along the centerline. Small deviations in deposition position can significantly modify local flow structures and, consequently, the

aerodynamic interaction between droplets. To assess this effect, two misaligned configurations were tested with a lateral offset of Δx = 150 µm, using droplet pairs B400-B600 and B500-B600. All cases share the same inlet velocity of 15 m/s and inter-droplet distances of D = 1.0, 2.0, and 3.0 mm.

As shown in Figure 50, the introduction of lateral misalignment noticeably alters the drag distribution between droplets. In the B400-B600 configuration (top plot), the smaller upstream droplet generates a wake that no longer fully covers the trailing one; as a result, the shielding effect is weakened, leading to smoother drag profiles. The drag levels of both droplets become more uniform, especially for larger separations (D ≥ 2.0 mm), indicating a partial aerodynamic decoupling; this effect is particularly visible in the comparison with Figure 49, where the solid blue line experienced a steeper drop, before reaching a stable value as it approaches the outlet.

In the B500-B600 configuration (bottom plot), where droplet sizes are more comparable, the wake asymmetry causes minor oscillations in the drag signal, but no coalescence occurs. The lateral offset induces a redistribution of the airflow around the droplets, effectively reducing the wake recirculation intensity and stabilizing the drag over time.

These results highlight the sensitivity of droplet interaction to spatial alignment and suggest that even small geometric deviations can suppress coalescence phenomena typically observed in perfectly aligned configurations.

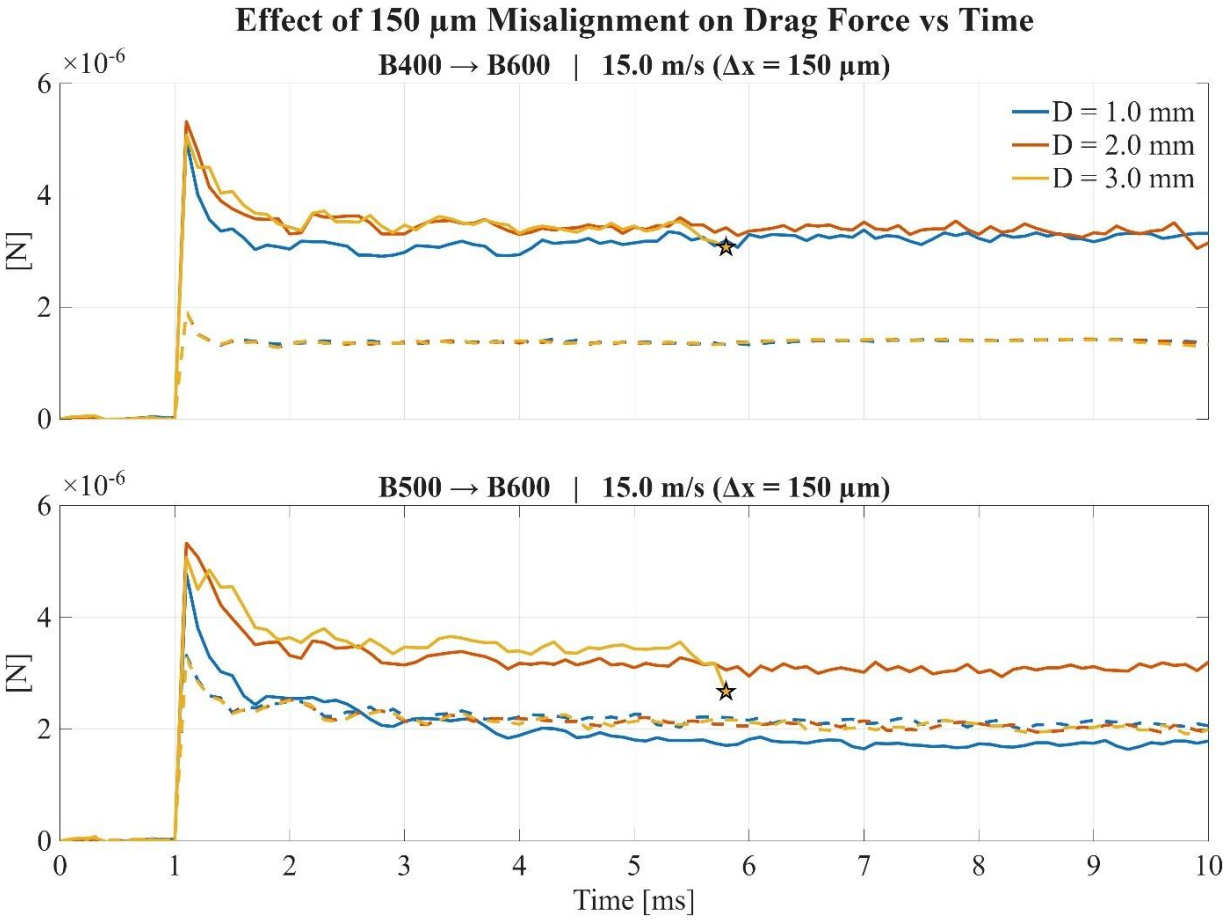


Figure 50. Longitudinal drag force for B400-B600 (top) and B500-B600 (bottom) droplets pair on a hydrophilic surface under a 15 m/s airflow and starting with 150 microns offset

4. Conclusion and future works

As outlined in the introduction, the work presented in this thesis originated from the need to better understand the mechanisms governing water droplet removal within the gas channels of polymer electrolyte fuel cells. Over time, this initial objective evolved into a broader investigation of multiphase flow modelling, leading to the development of a general computational framework for droplet-surface and droplet-flow interactions.

The implemented force-based dynamic contact angle boundary condition, fully integrated into the interFoam solver of OpenFOAM, allows the independent evaluation of adhesion and aerodynamic drag forces acting on each droplet in the computational domain. Each droplet is identified through a unique scalar ID field, which enables the localized calculation of interfacial forces and the consistent application of either adhesion or sliding behaviour according to the local force balance. This formulation maintains full compatibility with OpenFOAM's native VOF framework, ensuring numerical stability and modular integration.

The model was validated through a progressive series of simulations. Single-droplet tests demonstrated that the implemented boundary accurately reproduces static wetting conditions and the transition from adhesion to motion under aerodynamic loads. The comparison with analytical drag formulations confirmed the correct scaling of the computed forces with droplet size, velocity, and surface tension. Multi-droplet configurations further validated the model's ability to capture realistic interaction phenomena such as wake shielding, coalescence, and aerodynamic coupling, with results consistent with experimental visualizations. Moreover, sensitivity analyses revealed that even small lateral misalignments between droplets can significantly alter the local flow field, reducing coalescence likelihood and stabilizing drag fluctuations.

Beyond the solver-level validation, the reconstructed PEMFC channels served as an important physical benchmark to evaluate the influence of surface wettability and motion regime on droplet dynamics. The results confirmed that when rolling motion dominates, typically for surfaces with reduced adhesion and higher contact angles, the static contact angle is a sufficiently accurate representation of interface behaviour, as the adhesion contribution to the global force balance is minor. Conversely, for sliding-dominated motion, where adhesion forces are comparable to or exceed aerodynamic drag, modelling the evolution of the contact angle becomes essential. In these cases, the implemented force-based dynamic contact angle approach enables a physically consistent representation of advancing and receding angles, allowing smooth and realistic transitions between pinned and moving states.

From a numerical standpoint, the boundary proved stable and robust, exhibiting no artificial oscillations or non-physical interface deformation. The results demonstrated not only quantitative agreement with theoretical models but also the correct temporal evolution of aerodynamic loads, including the unsteady fluctuations associated with vortex shedding and local flow recirculation. Nevertheless, several aspects could be further improved or extended in future developments:

I. Performance optimization.

The current implementation relies on a PLIC-like interface reconstruction procedure, which ensures geometric accuracy but requires orthogonal meshes and involves relatively high computational cost. Future work should focus on extending the approach to non-orthogonal meshes, possibly by adopting local gradient-based reconstruction or adaptive stencil methods to improve both efficiency and generality.

II. Improved coalescence management.

While the present framework correctly identifies and merges droplets upon contact, the dynamic update of the total droplet count can introduce temporary inconsistencies in the ID mapping. Developing a more robust tracking algorithm would improve accuracy and prevent index mismatches during coalescence events.

III. Extended validation and broader applicability.

The framework has been validated primarily under conditions representative of PEMFC water management. However, its formulation is general and can be applied to other multiphase systems involving wetting dynamics, such as condensation, icing, spray cooling, or inkjet deposition. Future studies could also explore the coupling of this boundary with temperature-dependent wetting models or transient pressure variations.

Overall, this work contributes to advancing multiphase CFD by introducing a physically consistent, force-based model for dynamic wetting and droplet interaction. The developed framework provides both methodological flexibility and physical realism, bridging the gap between simplified geometric contact models and fully resolved interface dynamics. Beyond its specific application to PEMFCs, the proposed approach lays the groundwork for simulating a wide range of interfacial processes where the interaction between adhesion, capillarity, and aerodynamic forces governs system behaviour.

.

5. Brief recap of the other performed activities

During the three years of the Ph.D. program, several additional research topics were explored in parallel with the main work presented in this thesis. While not directly connected to the force-based contact angle modelling framework, these activities share the overarching objective of reducing the environmental and energetic impact of human technologies, particularly in the context of sustainable energy systems.

5.1. Ejector-Based Hydrogen Recirculation

To ensure stable operation of a PEMFC and to prevent starvation phenomena, that is, insufficient reactant availability at the catalyst layer, systems are typically operated with a controlled excess of hydrogen. As a consequence, part of the supplied hydrogen does not react electrochemically and exits the cell unconsumed. This unreacted hydrogen represents a loss of efficiency, as the overall system performance must account for the total energy input, not only for the fraction effectively used in the reaction. Therefore, an efficient hydrogen recirculation strategy is essential to optimize system performance.

Three main approaches are generally adopted for hydrogen recirculation:

- Active recirculation via a mechanical pump
  In this configuration, a dedicated pump is used to drive the flow of unused hydrogen back to the anode inlet. This solution is mechanically reliable and ensures full control of the recirculated mass flow rate, but it introduces additional system complexity, control requirements, and maintenance needs.

- Passive recirculation using an ejector.
  The ejector essentially operates as a Venturi device. A primary flow of hydrogen from the supply tank is accelerated through a converging nozzle, generating a local low-pressure region that entrains the secondary flow, i.e. the hydrogen exiting the stack. The two streams mix in a constant-area mixing chamber, exchange momentum, and then the pressure is recovered in a diffuser section before re-entering the cell. Compared to the active pump configuration, the ejector offers the advantages of simplicity, compactness, silent operation, and no moving parts, making it an attractive option for low-maintenance applications. However, its performance strongly depends on geometric design, and its optimal operating range can be narrow if proper precautions are not taken.

- Hybrid solutions
  In hybrid systems, both a pump and an ejector are employed. The pump operates primarily during transients and low-power conditions, whereas the ejector provides

baseline recirculation during steady-state operation. This combined configuration allows for improved system flexibility and can extend the overall operational envelope.

The performance of an ejector is typically quantified through the entrainment ratio, defined as the ratio between secondary and primary mass flow rates. This parameter directly affects the hydrogen excess ratio and therefore the global fuel utilization of the stack. Ejector behaviour is highly sensitive to geometric features such as the primary nozzle throat diameter, the axial position of the nozzle relative to the mixing chamber, the mixing section length-to-diameter ratio, and the diffuser angle. Due to compressible flow effects and potential supersonic regimes, even small geometric variations may produce significant changes in entrainment capability. Moreover, ejector operation is influenced by back-pressure conditions at the outlet, which in real PEMFC systems can vary as a consequence of load changes and/or water accumulation in the anode channels. For this reason, the primary nozzle is often designed to operate under choked conditions, ensuring partial decoupling from downstream pressure fluctuations.

Following an extensive literature review [91], a CFD model was developed to investigate ejector performance under representative operating conditions.

The chosen geometry is the one reported by Nikiforow et al. [92] for a 5 kW PEMFC with an extensive experimental campaign reported in literature. An axisymmetric two-dimensional steady-state formulation was adopted as a compromise between computational cost and predictive capability. The model assumes single-phase compressible hydrogen flow with ideal gas behaviour and adiabatic walls. A density-based solver was selected to properly capture supersonic regions, and turbulence was modelled using a realizable k–ε formulation with enhanced wall treatment. Particular attention was devoted to mesh refinement in the vicinity of the nozzle throat and mixing chamber, where strong velocity gradients and pressure variations occur (see Figure 51). Convergence of the simulation was verified by monitoring residuals and global mass balance; Figure 52 shows the mesh independence analysis, that was performed employing different levels of refinement (105 795, 138 300 and 211 590 elements) to ensure solution robustness of the model.

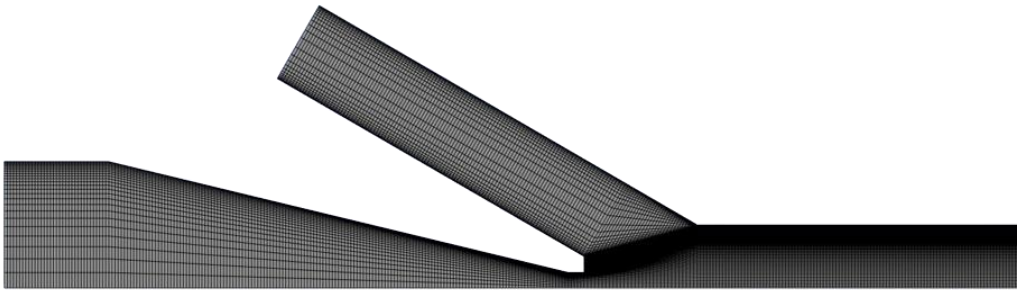


Figure 51. Detail of the mesh in the primary/secondary mixing region

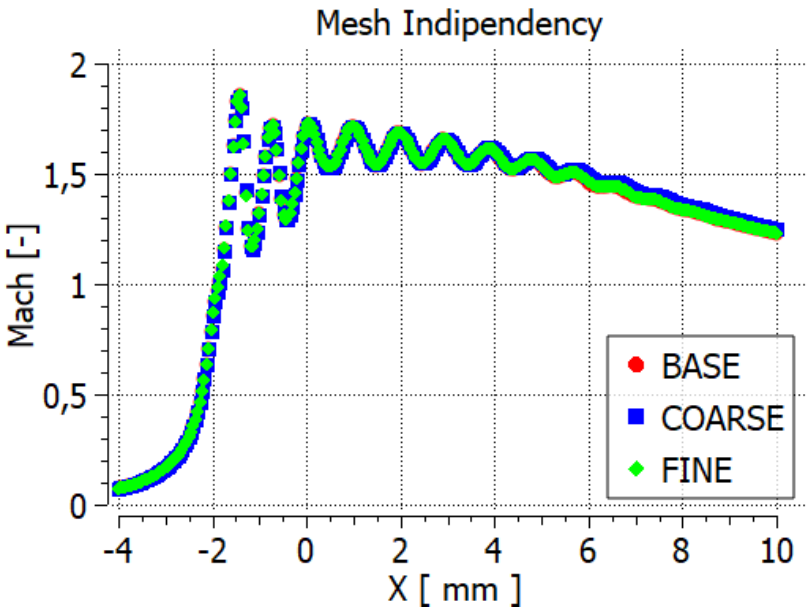


Figure 52. Mach number along the symmetry axis for different meshes

The numerical results reproduced the characteristic features of ejector operation, including the formation of a supersonic jet downstream of the primary nozzle, the development of a low-pressure region responsible for secondary flow entrainment, and the progressive mixing within the constant-area section. Sensitivity analyses with respect to outlet back-pressure highlighted the reduction of the sonic core length and the appearance of recirculation zones at elevated pressures, in agreement with trends reported in the literature. The model was validated against experimental data [92], showing satisfactory agreement in terms of primary mass flow rate (see Figure 53).

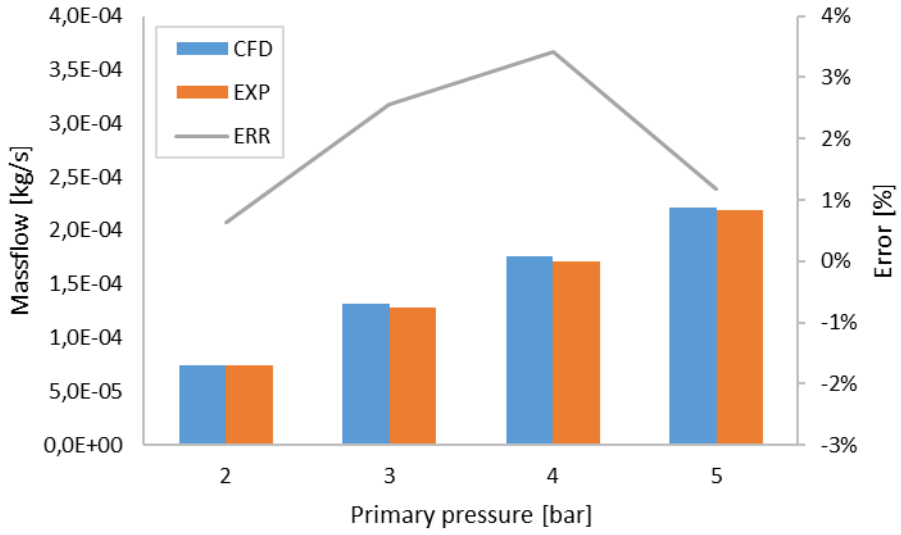


Figure 53. Comparison with experimental data for primary massflow over different pressure levels

Since PEMFC applications range from low-power stationary units to high-power automotive systems, a geometric scaling analysis was subsequently conducted. Ejector scaling does not follow a simple linear law because compressible flow phenomena and choking conditions impose non-dimensional similarity constraints. The primary nozzle throat diameter was determined from isentropic choked-flow relations based on the maximum hydrogen demand at full load. Because the choked mass flow rate scales with throat area, the minimum nozzle diameter increases proportionally to the square root of the required mass flow rate. Key geometric ratios between nozzle, mixing chamber, and diffuser were preserved to maintain similar flow structures [93]. Figure 54 shows the transition from the fluid domain, to the solid object, to the 3D printed version for an ejector able to adequately supply the anodic line of a 80 kW PEMFC.

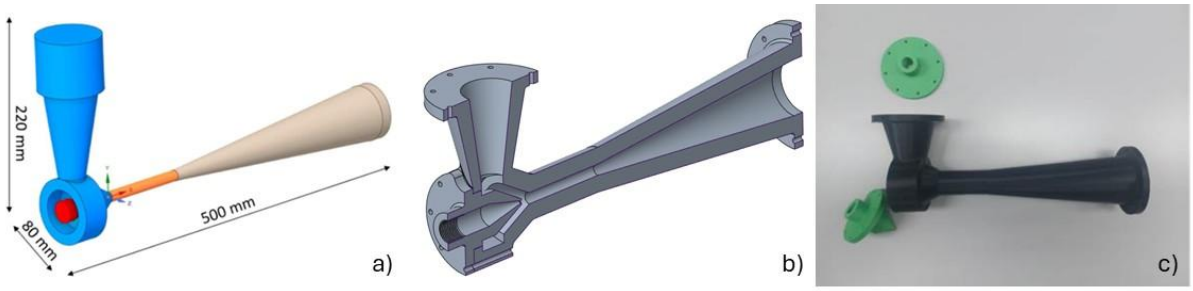


Figure 54. Overall dimensions of the proposed geometry (a), CAD model of the ejector (b), and 3D-printed prototype (c).

5.2. PEMFC 0-D modelling

A second research activity carried out during the Ph.D. concerned the development of a 0-D dynamic model of a complete PEMFC-based powertrain, including all major Balance of Plant (BoP) components. The objective was not to resolve local electrochemical phenomena, but rather to provide a computationally efficient framework capable of reproducing the global behaviour of a full fuel cell system under time-varying load conditions. The model was implemented in MATLAB/Simulink and designed to capture the key electrochemical, thermal, and fluid-dynamic couplings governing stack and system-level performance. Unlike high-fidelity multidimensional simulations, the 0-D approach treats each subsystem through lumped-parameter formulations, enabling fast simulations over long driving cycles or extended durability analyses. This makes the tool particularly suitable for system optimization, control strategy development, and hybrid powertrain studies.

The model includes the following main subsystems:

- **Air supply unit**, comprising a compressor and simplified intercooling representation, accounting for parasitic power consumption and pressure ratio effects.

- **Humidification section**, responsible for conditioning reactant streams in order to maintain adequate membrane hydration.
- **Pre-heating unit**, ensuring suitable inlet temperatures for both anode and cathode flows.
- **Cooling loop**, including a liquid coolant pump and a simplified thermal management model to regulate stack operating temperature.
- **PEMFC stack**, described through a lumped polarization formulation.
- **Battery pack**, coupled in a parallel hybrid configuration.
- **Electric motor and vehicle subsystem**, providing the time-dependent power demand according to standardized driving cycles.

The PEMFC stack voltage is computed as already described in 1.2, through overpotentials; the resulting polarization curve defines the admissible operating points of the stack during dynamic simulations. Once current density is determined, stack power, hydrogen consumption (via Faraday's law), and auxiliary loads are consistently computed. This approach ensures thermodynamic coherence between electrochemical conversion and system-level mass and energy balances.

The total power demand then is derived from the electric motor requirement, which depends on vehicle longitudinal dynamics. The tractive force is computed from inertial, aerodynamic, and rolling contributions, while accounting for transmission efficiency. Standardized driving cycles (e.g., WLTP) are used as input velocity traces.

In the adopted parallel hybrid configuration, total power demand is split between the fuel cell stack and the battery pack according to a rule-based Energy Management System (EMS). The strategy is implemented using Stateflow and distinguishes four operating modes:

- Idle
- Battery-only operation
- Fuel-cell-only operation
- Regenerative braking

The power split is primarily governed by the required motor power and by the battery State of Charge (SOC). Non-symmetrical SOC thresholds are introduced to guarantee stable operation and prevent deep discharge. In high-SOC conditions, the battery supports transient peaks, while the fuel cell operates closer to steady conditions, reducing rapid load variations that are known to accelerate degradation. During low-SOC states, priority is given to recharging. This rule-based logic allows investigation of the trade-off between fuel consumption, stack durability, and battery stress.

For validation purposes, experimental data from Argonne National Laboratory on a commercial Toyota Mirai powertrain were used as reference [94,95]. While the implemented EMS differs from the proprietary strategy of the vehicle, comparisons demonstrated:

- Good agreement in tractive force prediction.
- Comparable stack power trends.
- Realistic hydrogen consumption estimation.
- Preservation of SOC between start and end of the driving cycle.

The main discrepancies arise from unavailable electric motor efficiency maps and from deliberate differences in control philosophy. Further details are reported in publications [96,97].

Acknowledgement

The authors gratefully acknowledge the partial financial support of the European Union, NextGenerationEU - in the framework of the National Sustainable Mobility Center - MOST, CN00000023, Italian Ministry of University and Research Decree n. 1033— 17/06/2022, Spoke 12, CUP B43C22000440001.

List of abbreviations

BoP	Balance of Plant
BP	Bipolar Plate
CAH	Contact Angle Hysteresis
CL	Catalyst Layer
CLSVOF	Coupled Level Set and Volume Of Fluid
CFD	Computational Fluid Mechanic
CSF	Continuum Surface Force (model)
EIS	Electrochemical Impedance Spectroscopy
EMS	Energy Management System
ER	Entrainment Ratio
EU	European Union
GDL	Gas Diffusion Layer
GDP	Gross Domestic Product
GHG	Greenhouse Gas
LBM	Lattice-Boltzmann Method
LS	Level Set
MEA	Membrane Electrode Assembly
PEMFC	Proton Exchange Membrane Fuel Cell
PLIC	Piecewise Linear Interface Calculation
ROI	Region of Interest
SLIC	Simple Line Interface Calculation

SOC	State Of Charge
UDF	User Defined Function
VOF	Volume Of Fluid

List of symbols

Latin symbols

A_{adh}	Adhesion area
a	Tafel coefficient
b	Tafel slope
Ca	Capillary number
C_D	Drag coefficient
D_{adh}	Equivalent adhesion diameter
dA	Interface area (per cell basis)
i_{lim}	Limiting current (conc. losses)
F	Faraday constant
f_H	Hoffman function
F_P	Pressure force (component of the total drag force)
F_μ	Viscous force (component of the total drag force)
f_s	Contact area factor (Cassie-Baxter model)
f_σ	Surface tension force
g	Gravitational acceleration
k_k	Calibration parameter (adhesion model)
n_i	Normal direction
n_w	Normal direction to the wall
R_{ohm}	Ohmic resistance
R	Force ratio
r	Roughness factor (Wenzel model)
V	Cell volume

U	Velocity

Greek symbols

α	Volume fraction
ϵ	Length scale (hydrodynamic model)
η	Voltage losses
θ	Contact angle
κ	Interface curvature
μ	Dynamic viscosity
ρ	Density
σ	Surface tension
τ	Viscous stress tensor

Bibliography

1. EU Commission, Press Release, European Green Deal: the Commission proposes transformation of EU economy and society to meet climate ambitions, 14 July 2021, available at https://ec.europa.eu/commission/presscorner/detail/en/IP_21_3541, accessed 07 October 2025.
2. Climate change – driving forces 2023: figures and tables, Eurostat, 2023, available at: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Climate_change_-_driving_forces&stable=1#General_overview, accessed 07 October 2025.
3. Li, Y., & Jenn, A. (2024). Impact of electric vehicle charging demand on power distribution grid congestion. Proceedings of the National Academy of Sciences, 121(18), e2317599121.
4. Palys, M. J., & Daoutidis, P. (2022). Power-to-X: A review and perspective. Computers & Chemical Engineering, 165, 107948.
5. Dell'Aversano, S., Villante, C., Gallucci, K., Vanga, G., & Di Giuliano, A. (2024). E-fuels: a comprehensive review of the most promising technological alternatives towards an energy transition. Energies, 17(16), 3995.
6. Incer-Valverde, J., Patiño-Arévalo, L. J., Tsatsaronis, G., & Morosuk, T. (2022). Hydrogen-driven Power-to-X: State of the art and multicriteria evaluation of a study case. Energy Conversion and Management, 266, 115814.
7. Fan, J. L., Yu, P., Li, K., Xu, M., & Zhang, X. (2022). A levelized cost of hydrogen (LCOH) comparison of coal-to-hydrogen with CCS and water electrolysis powered by renewable energy in China. Energy, 242, 123003.
8. Barbir, F. (2012). PEM fuel cells: theory and practice. Academic press.
9. Chen, L., Xu, K., Yang, Z., Yan, Z., & Dong, Z. (2022). Optimal design and operation of dual-ejector PEMFC hydrogen supply and circulation system. Energies, 15(15), 5427.
10. Gao, Y., & Lin, M. (2024). Research on the performance characteristics of hydrogen circulation pumps for PEMFC vehicles. International Journal of Hydrogen Energy, 50, 1255-1272.
11. The U.S. Department of Energy. Hydrogen and fuel cell technologies office, department of energy. 2019. URL https://www.energy.gov/eere/fuelcells/fact-month-april-2018-fuel-cell-cost-decreased-60-2006. Accessed: 13th February 2025.
12. Chen, K., Laghrouche, S., & Djerdir, A. (2021). Performance analysis of PEM fuel cell in mobile application under real traffic and environmental conditions. Energy Conversion and Management, 227, 113602.
13. Cigolotti, V., Genovese, M., & Fragiacomo, P. (2021). Comprehensive review on fuel cell technology for stationary applications as sustainable and efficient poly-generation energy systems. Energies, 14(16), 4963.
14. Chung, C. G., Kim, L., Sung, Y. W., Lee, J., & Chung, J. S. (2009). Degradation mechanism of electrocatalyst during long-term operation of PEMFC. International Journal of Hydrogen Energy, 34(21), 8974-8981.
15. Zhou, D., Wu, Y., Gao, F., Breaz, E., Ravey, A., & Miraoui, A. (2017). Degradation prediction of PEM fuel cell stack based on multiphysical aging model with particle filter approach. IEEE Transactions on Industry Applications, 53(4), 4041-4052.
16. Ji, M., & Wei, Z. (2009). A review of water management in polymer electrolyte membrane fuel cells. Energies, 2(4), 1057-1106.
17. Jiao, K., Wang, B., Du, Q., Wang, Y., Zhang, G., Yang, Z., ... & Xie, X. (2021). Water and thermal management of proton exchange membrane fuel cells. Elsevier.
18. Xu, S., Liao, P., Yang, D., Li, Z., Li, B., Ming, P., & Zhou, X. (2023). Liquid water transport in gas flow channels of PEMFCs: A review on numerical simulations and visualization experiments. International Journal of Hydrogen Energy, 48(27), 10118-10143.

19. Satjaritanun, P., Weidner, J. W., Hirano, S., Lu, Z., Khunatorn, Y., Ogawa, S., ... & Shimpalee, S. (2017). Micro-scale analysis of liquid water breakthrough inside gas diffusion layer for PEMFC using X-ray computed tomography and lattice Boltzmann method. Journal of The Electrochemical Society, 164(11), E3359.
20. Sergi, J. M., & Kandlikar, S. G. (2011). Quantification and characterization of water coverage in PEMFC gas channels using simultaneous anode and cathode visualization and image processing. International Journal of Hydrogen Energy, 36(19), 12381-12392.
21. Ghasabehi, M., Jabbary, A., & Shams, M. (2022). Cathode side transport phenomena investigation and Multi-Objective optimization of a tapered parallel flow field PEMFC. Energy conversion and management, 265, 115761.
22. Ma, Y., Wang, X., Yuan, H., Chang, G., Zhu, J., Dai, H., & Wei, X. (2025). Review of electrochemical impedance spectroscopy in fault diagnosis for proton exchange membrane fuel cells. Renewable and Sustainable Energy Reviews, 211, 115226.
23. Pivac, I., Bezmalinović, D., & Barbir, F. (2018). Catalyst degradation diagnostics of proton exchange membrane fuel cells using electrochemical impedance spectroscopy. International Journal of Hydrogen Energy, 43(29), 13512-13520.
24. Josyula, T., Kumar Malla, L., Thomas, T. M., Kalichetty, S. S., Sinha Mahapatra, P., & Pattamatta, A. (2024). Fundamentals and applications of surface wetting. Langmuir, 40(16), 8293-8326.
25. Soltman, D., Smith, B., Morris, S. J. S., & Subramanian, V. (2013). Inkjet printing of precisely defined features using contact-angle hysteresis. Journal of colloid and interface science, 400, 135-139.
26. Farris, S., Introzzi, L., Biagioni, P., Holz, T., Schiraldi, A., & Piergiovanni, L. (2011). Wetting of biopolymer coatings: Contact angle kinetics and image analysis investigation. Langmuir, 27(12), 7563-7574.
27. Hodgson, G., Passmore, M., Skarysz, M., Garmory, A., & Paolillo, F. (2021). Contact angle measurements for automotive exterior water management. Experiments in Fluids, 62(5), 119.
28. Zhang, X., & Qin, Y. (2019). Contact angle hysteresis of a water droplet on a hydrophobic fuel cell surface. Journal of colloid and interface science, 545, 231-241.
29. Ko, Y. C., Ratner, B. D., & Hoffman, A. S. (1981). Characterization of hydrophilic—hydrophobic polymeric surfaces by contact angle measurements. Journal of Colloid and Interface Science, 82(1), 25-37.
30. Wenzel, R. N. (1936). Resistance of solid surfaces to wetting by water. Industrial & engineering chemistry, 28(8), 988-994.
31. Cassie, A. B. D., & Baxter, S. (1944). Wettability of porous surfaces. Transactions of the Faraday society, 40, 546-551.
32. Hebbar, R. S., Isloor, A. M., & Ismail, A. F. (2017). Contact angle measurements. In Membrane characterization (pp. 219-255). Elsevier.
33. Blake, T. D. (2006). The physics of moving wetting lines. Journal of colloid and interface science, 299(1), 1-13.
34. Voinov, O. V. (1976). Hydrodynamics of wetting. Fluid dynamics, 11(5), 714-721.
35. Dussan, V. E. B. (1976). The moving contact line: the slip boundary condition.
36. Cox, R. G. (1986). The dynamics of the spreading of liquids on a solid surface. Part 1. Viscous flow. Journal of fluid mechanics, 168, 169-194.
37. Sprittles, J. E., & Shikhmurzaev, Y. D. (2013). Finite element simulation of dynamic wetting flows as an interface formation process. Journal of Computational Physics, 233, 34-65.
38. Glasstone, S., Laidler, K. J., & Eyring, H. (1941). The theory of rate processes: the kinetics of chemical reactions, viscosity, diffusion and electrochemical phenomena.
39. Shikhmurzaev, Y. D. (2007). Capillary flows with forming interfaces. Chapman and Hall/CRC.
40. Huh, C., & Mason, S. G. (1977). Effects of surface roughness on wetting (theoretical). Journal of colloid and interface science, 60(1), 11-38.
41. Eral, H. B., 't Mannetje, D. J. C. M., & Oh, J. M. (2013). Contact angle hysteresis: a review of fundamentals and applications. Colloid and polymer science, 291(2), 247-260.
42. Butt, H. J., & Kappl, M. (2018). Surface and interfacial forces. John Wiley & Sons.

43. Tadmor, R. (2004). Line energy and the relation between advancing, receding, and young contact angles. Langmuir, 20(18), 7659-7664.
44. Kistler, S. F. (1993). Hydrodynamics of wetting. Wettability, 6, 311-430.
45. Bazylak, A. (2009). Liquid water visualization in PEM fuel cells: A review. International journal of hydrogen energy, 34(9), 3845-3857.
46. Ji, M., & Wei, Z. (2009). A review of water management in polymer electrolyte membrane fuel cells. Energies, 2(4), 1057-1106.
47. Liu, X., Peng, F., Lou, G., & Wen, Z. (2015). Liquid water transport characteristics of porous diffusion media in polymer electrolyte membrane fuel cells: A review. Journal of Power Sources, 299, 85-96.
48. Sinha, P. K., & Wang, C. Y. (2007). Pore-network modeling of liquid water transport in gas diffusion layer of a polymer electrolyte fuel cell. Electrochimica Acta, 52(28), 7936-7945.
49. Shin, S., & Juric, D. (2002). Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. Journal of Computational Physics, 180(2), 427-470.
50. Scardovelli, R., & Zaleski, S. (1999). Direct numerical simulation of free-surface and interfacial flow. Annual review of fluid mechanics, 31(1), 567-603.
51. Baraldi, A., Dodd, M. S., & Ferrante, A. (2014). A mass-conserving volume-of-fluid method: volume tracking and droplet surface-tension in incompressible isotropic turbulence. Computers & Fluids, 96, 322-337.
52. Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S., & Zanetti, G. (1994). Modelling merging and fragmentation in multiphase flows with SURFER. Journal of computational physics, 113(1), 134-147.
53. Brackbill, J. U., Kothe, D. B., & Zemach, C. (1992). A continuum method for modeling surface tension. Journal of computational physics, 100(2), 335-354.
54. Afkhami, S., & Bussmann, M. (2008). Height functions for applying contact angles to 2D VOF simulations. International journal for numerical methods in fluids, 57(4), 453-472.
55. Afkhami, S., Zaleski, S., & Bussmann, M. (2009). A mesh-dependent model for applying dynamic contact angles to VOF simulations. Journal of computational physics, 228(15), 5370-5389.
56. Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of computational physics, 79(1), 12-49.
57. Sussman, M., & Puckett, E. G. (2000). A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. Journal of computational physics, 162(2), 301-337.
58. Albadawi, A., Donoghue, D. B., Robinson, A. J., Murray, D. B., & Delauré, Y. M. C. (2013). On the analysis of bubble growth and detachment at low capillary and bond numbers using volume of fluid and level set methods. Chemical Engineering Science, 90, 77-91.
59. Li, Q., Luo, K. H., Kang, Q. J., He, Y. L., Chen, Q., & Liu, Q. (2016). Lattice Boltzmann methods for multiphase flow and phase-change heat transfer. Progress in Energy and Combustion Science, 52, 62-105.
60. Jasak, H., Jemcov, A., & Tukovic, Z. (2007, September). OpenFOAM: A C++ library for complex physics simulations. In International workshop on coupled methods in numerical dynamics (Vol. 1000, pp. 1-20).
61. Park, S., Lee, J. W., & Popov, B. N. (2012). A review of gas diffusion layer in PEM fuel cells: Materials and designs. International Journal of Hydrogen Energy, 37(7), 5850-5865.
62. Nabovati, A., Hinebaugh, J., Bazylak, A., & Amon, C. H. (2014). Effect of porosity heterogeneity on the permeability and tortuosity of gas diffusion layers in polymer electrolyte membrane fuel cells. Journal of Power Sources, 248, 83-90.
63. Yang, D., Garg, H., & Andersson, M. (2023). Numerical simulation of two-phase flow in gas diffusion layer and gas channel of proton exchange membrane fuel cells. International Journal of Hydrogen Energy, 48(41), 15677-15694.
64. Nitta, I., Himanen, O., & Mikkola, M. (2008). Thermal conductivity and contact resistance of compressed gas diffusion layer of PEM fuel cell. Fuel Cells, 8(2), 111-119.
65. Eller, J., Roth, J., Gaudenzi, R., Irvine, S., Marone, F., Stampanoni, M., ... & Büchi, F. N. (2013). Water distribution in GDL near optimal humidification. ECS transactions, 50(2), 477.

66. Park, S., & Popov, B. N. (2011). Effect of a GDL based on carbon paper or carbon cloth on PEM fuel cell performance. Fuel, 90(1), 436-440.
67. Flückiger, R., Marone, F., Stampanoni, M., Wokaun, A., & Büchi, F. N. (2011). Investigation of liquid water in gas diffusion layers of polymer electrolyte fuel cells using X-ray tomographic microscopy. Electrochimica Acta, 56(5), 2254-2262.
68. Zenyuk, I. V., Parkinson, D. Y., Connolly, L. G., & Weber, A. Z. (2016). Gas-diffusion-layer structural properties under compression via X-ray tomography. Journal of Power Sources, 328, 364-376.
69. Yang, D., Garg, H., Beale, S. B., & Andersson, M. (2023). Numerical Reconstruction of Proton Exchange Membrane Fuel Cell Gas Diffusion Layers. ECS transactions, 112(4), 49.
70. Gao, Y., Jin, T., & Wu, X. (2020). Stochastic 3D carbon cloth GDL reconstruction and transport prediction. Energies, 13(3), 572.
71. Calvimontes, A., Badrul Hasan, M. M., & Dutschk, V. (2010). Effects of topographic structure on wettability of woven fabrics. Woven Fabric Engineering. Rijeka: Sciyo, 71-92.
72. Calvimontes, A., Dutschk, V., & Stamm, M. (2010). Advances in topographic characterization of textile materials. Textile Research Journal, 80(11), 1004-1015.
73. Hou, Y., Deng, H., Zamel, N., Du, Q., & Jiao, K. (2020). 3D lattice Boltzmann modeling of droplet motion in PEM fuel cell channel with realistic GDL microstructure and fluid properties. International Journal of Hydrogen Energy, 45(22), 12476-12488.
74. Liu, S., Zhang, L., Wang, Z., & Li, R. (2021). Influence of the surface microstructure of the fuel cell gas diffusion layer on the removal of liquid water. International Journal of Hydrogen Energy, 46(62), 31764-31777.
75. ANSYS Inc. Software. ANSYS-Fluent 12.0/12.1 Documentation Web Site, 2020.
76. Gariboldi, C., & Takahashi, T. (2022). Asymptotic analysis of an optimal control problem for a viscous incompressible fluid with Navier slip boundary conditions. Asymptotic Analysis, 126(3-4), 379-399.
77. Legendre, D., & Maglio, M. (2015). Comparison between numerical models for the simulation of moving contact lines. Computers & Fluids, 113, 2-13.
78. Holmgren, H., & Kreiss, G. (2015). Towards accurate modeling of moving contact lines. arXiv preprint arXiv:1510.06639.
79. Kistler, S. F. (1993). Hydrodynamics of wetting. Wettability, 6, 311-430.
80. Hoffman, R. L. (1975). A study of the advancing interface. I. Interface shape in liquid—gas systems. Journal of colloid and interface science, 50(2), 228-241.
81. Tanner, L. H. (1979). The spreading of silicone oil drops on horizontal surfaces. Journal of Physics D: Applied Physics, 12(9), 1473.
82. Nichita, B. A., Zun, I., & Thome, J. R. (2010). A VOF method coupled with a dynamic contact angle model for simulation of two-phase flows with partial wetting. In 7th International Conference on Multiphase Flow, ICMF.
83. Heinrich, M., & Schwarze, R. (2020). 3D-coupling of Volume-of-Fluid and Lagrangian particle tracking for spray atomization simulation in OpenFOAM. SoftwareX, 11, 100483.
84. ElSherbini, A.I., &Jacobi, A.M. (2006).Retention forces and contact angles for critical liquid drops on non - horizontal surfaces.Journal of colloid and interface science, 299(2), 841 - 849.
85. Cho, S. C., Wang, Y., & Chen, K. S. (2012). Droplet dynamics in a polymer electrolyte fuel cell gas flow channel: Forces, deformation, and detachment. I: Theoretical and numerical analyses. Journal of power sources, 206, 119-128.
86. Li, M., Li, Y., Qin, Y., Yin, Y., Zhang, J., & Che, Z. (2022). Water droplet detachment characteristics on surfaces of gas diffusion layers in PEMFCs. International Journal of Hydrogen Energy, 47(18), 10341-10351.
87. Schlichting, H., & Gersten, K. (2016). Boundary-layer theory. springer.
88. Podgorski, T. (2000). Ruissellement en conditions de mouillage partiel (Doctoral dissertation, Université Pierre et Marie Curie-Paris VI).
89. Le Grand, N., Daerr, A., & Limat, L. (2005). Shape and motion of drops sliding down an inclined plane. Journal of Fluid Mechanics, 541, 293-315.

90. Maglio, M., & Legendre, D. (2014). Numerical simulation of sliding drops on an inclined solid surface. In Computational and Experimental Fluid Mechanics with Applications to Physics, Engineering and the Environment (pp. 47-69). Cham: Springer International Publishing.
91. Antetomaso, C. (2024). State of the art of modelling and design approaches for ejectors in proton exchange membrane fuel cell. Modelling and Simulation in Engineering, 2024(1), 7931501.
92. Nikiforow, K., Koski, P., Karimäki, H., Ihonen, J., & Alopaeus, V. (2016). Designing a hydrogen gas ejector for 5 kW stationary PEMFC system–CFD-modeling and experimental validation. International Journal of Hydrogen Energy, 41(33), 14952-14970.
93. Antetomaso, C., Irimescu, A., Merola, S. S., Vaglieco, B. M., Di Micco, S., & Jannelli, E. (2024). Ejector design for PEM fuel cells and assessment of its scalability. International Journal of Hydrogen Energy, 95, 1235-1241.
94. Usmanov, U., Ruzimov, S., Tonoli, A., & Mukhitdinov, A. (2023). Modeling, simulation and control strategy optimization of fuel cell hybrid electric vehicle. Vehicles, 5(2), 464-481.
95. Lohse-Busch, H., Stutenberg, K., Duoba, M., & Iliev, S. (2017). Technology assessment of a fuel cell vehicle: 2017 Toyota Mirai (No. ANL/ESD--18/12). Argonne National Laboratory (ANL), Argonne, IL (United States).
96. Martoccia, L., Antetomaso, C., Merola, S., Marra, C., Breda, S., & D'Adamo, A. (2025). Effects of Cell Design Improvement on an Automotive PEMFC System (No. 2025-01-8549). SAE Technical Paper.
97. Parmiggiani, D., Antetomaso, C., Martoccia, L., Merola, S., & d'Adamo, A. (2024, November). A MATLAB/Simulink model of a parallel hybrid PEMFC/battery powertrain for passenger cars. In Journal of Physics: Conference Series (Vol. 2893, No. 1, p. 012070). IOP Publishing.

Appendix A - Force-Based Dynamic Contact Angle Pseudocode

This appendix presents the pseudocode representation of the custom numerical routines developed and implemented within the OpenFOAM® framework to simulate droplet dynamics. Each algorithm corresponds to a specific module of the solver and is responsible for a distinct physical or numerical task, such as reading material properties, identifying and labeling liquid droplets, evaluating their kinematic and dynamic characteristics, and managing contact angle models. The pseudocode has been reformatted for clarity and consistency, omitting low-level implementation details while preserving the logical sequence of operations and physical meaning. Each section reports the algorithm's in the following manner:

Input:

[Input variable of the section]

Procedure:

[Alghoritm]

Output:

[Output results of the section]

The presented routines cover the following main functionalities:

- Reading and validation of transport and physical properties
- Detection and labeling of individual droplets through a flood-fill algorithm
- Evaluation of droplet center of mass, velocity, and total forces
- Time-resolved output of droplet properties and forces
- Computation of dynamic contact angles, including hysteresis and correlation-based models

Together, these algorithms form the computational backbone of the custom boundary, enabling accurate tracking and analysis of droplet motion, deformation, and interaction with solid boundaries.

A.1 - readProperties.H

Input:

- Mesh information from simulation domain
- File 'transportProperties' (containing fluid parameters)
- File 'g' (containing gravity vector)

Procedure:

Read the mesh structure from the boundary patch.

Read file 'transportProperties' from the constant directory.
Read file 'g' from the constant directory (gravity field).

Set default value def ← -999

Read the following parameters from 'transportProperties':

mu_water	← dynamic viscosity of water
sigma	← surface tension
rho_air	← density of air
rho_water	← density of water

If any of the above parameters equals def then

Display error message:

"Cannot read material properties (water mu/rho, air rho, and sigma)!

Please check your transportProperties file."

Terminate execution.

Output:

- Successfully loaded physical properties:

{mu_water, sigma, rho_air, rho_water, g}

- If reading fails: simulation stops with fatal error

A.2 - findDrop.H

Input:

- Volume fraction field α_water in the computational mesh
- Volume fraction threshold α_threshold = 0.01
- Mesh connectivity information (cell neighbors)

Procedure:

```
Create a scalar field vofID initialized to 0 for all cells.
Assign each cell a unique global ID (for parallel runs).

For each cell in the mesh:
    If α_water(cell) > α_threshold and vofID(cell) = 0 then
        Assign a new global droplet ID to the cell.
        Initialize a list 'cells' with all its neighboring
         cells.

        While 'cells' is not empty:
            Create an empty list 'neighbors'.
            For each cell_i in 'cells':
                If α_water(cell_i) > α_threshold and
                 vofID(cell_i) = 0 then
                    Set vofID(cell_i) = droplet ID
                    Append all neighbors of cell_i to
                     'neighbors'
            Replace 'cells' with 'neighbors'

Update boundary conditions for vofID across all processors.

If running in parallel:
    Repeat until no changes occur:
```

```
            For each processor boundary:
                For each face shared by two processors:
                    Compare IDs of neighboring cells (ID_own,
                     ID_nei)
                    If both are water cells and ID_own > ID_nei:
                        Propagate the smaller ID (ID_nei) through
                         the connected region.
            Synchronize vofID and correction flags across all
             processors.

    Write the final vofID field to file.

    Determine the list of unique droplet IDs (Labels):
        If parallel run:
            Each processor collects its local IDs (positive and
             unique).
            Master processor gathers all IDs, removes duplicates,
            counts the total number of droplets (dropNumber),
            and broadcasts Labels to all processors.
        Else:
            Identify unique IDs directly from vofID.

    On master processor:
        If the number of droplets changes from the previous
         timestep:
            Overwrite "dropNumber.txt" with the current droplet
             count.
        For each droplet i:
            Write its label ID to "droplet_i_label.txt".
```

Output:

- Field vofID labeling each liquid droplet with a unique ID

- Total number of droplets (dropNumber)
- Text files:
    - dropNumber.txt – number of identified droplets
    - droplet_i_label.txt – label ID for each droplet

A.3 - dropG.H

Input:

- List of identified droplet labels (Labels)
- Volume fraction field α_water
- Density of water ρ_water
- Mesh cell volumes V[cell] and cell centers C[cell]
- Volume fraction threshold α_threshold

Procedure:

```
    Initialize arrays:
        dropMass[dropNumber]    ← 0.0
        GCenter[dropNumber]     ← (0, 0, 0)
        GCenterOld[dropNumber]  ← (0, 0, 0)

    For each droplet index dropIdx in [1 .. dropNumber]:
        Initialize:
            waterCell      ← 0
            dropMass_s     ← 0.0
            cellCenterSum  ← (0, 0, 0)

        For each cell in the mesh:
            If α_water(cell) > α_threshold
               and vofID(cell) = Labels[dropIdx] then
                waterCell       ← waterCell + 1
                cellCenterSum   ← cellCenterSum + C[cell]
                dropMass_s      ← dropMass_s + ρ_water * V[cell] *
α_water(cell)

        If running in parallel:
```

```
        Sum waterCell, cellCenterSum, and dropMass_s across all
         processors.


    If waterCell > 0 then

        Compute droplet center of mass:

            GCenter_s ← cellCenterSum / waterCell


        Store results:

            dropMass[dropIdx] ← dropMass_s

            GCenter[dropIdx] ← GCenter_s
```

Output:

- Array dropMass[dropNumber]: mass of each droplet
- Array GCenter[dropNumber]: position of each droplet's center of mass
- Results synchronized across all processors (if parallel run)

A.4 - dropVel.H

Input:

- Velocity field U
- Volume fraction field α_water
- Volume fraction threshold α_threshold
- List of droplet labels (Labels)
- Field vofID identifying droplet cells

Procedure:

```
Initialize array velocityMean[dropNumber] ← (0, 0, 0)

For each droplet index dropIdx in [1 .. dropNumber]:
    Initialize:
        velocitySum    ← (0, 0, 0)
        cellCount      ← 0

    For each cell in the mesh:
        If α_water(cell) > α_threshold
           and vofID(cell) = Labels[dropIdx] then
            cellCount       ← cellCount + 1
            velocitySum     ← velocitySum + U[cell]

    If running in parallel:
        Sum velocitySum and cellCount across all processors.

    If cellCount > 0 then
        Compute mean velocity:
            velocityMean[dropIdx] ← velocitySum / cellCount
    Else:
```

```
            velocityMean[dropIdx] ← (0, 0, 0)
```

Output:

- Array velocityMean[dropNumber]: mean velocity vector of each droplet
- Results synchronized across all processors (if parallel run)

A.5 - printGposition&velocity.H

Input:

- Current simulation time currentTime
- Output flag mesh.time().outputTime()
- Arrays:
  - GCenter[dropNumber] → droplet center of mass positions
  - velocityMean[dropNumber] → droplet mean velocities
- droplet count dropNumber

(Executed only by master processor)

Procedure:

```
If running on master processor then
    If currentTime = 0 or output time,
       and currentTime ≠ lastWrittenTime then
        Update lastWrittenTime ← currentTime

        For each droplet index dropIdx in [1 .. dropNumber]:
            Define fileName ← "CenterOfMass_dropIdx.txt"
            Open fileName in append mode.

            If file is empty then
                Write header:
                    "Time [s]    Center Of Mass Position [m]
                     Center Of Mass Velocity [m/s]"

            Append current data line:
                currentTime, GCenter[dropIdx],
                velocityMean[dropIdx]
```

Close file

Output:

- One text file per droplet:

  "CenterOfMass_i.txt"

- Each file contains time history of:
  - droplet center of mass position
  - droplet mean velocity
- Written only at simulation output timesteps

A.6 - forcesOnDroplet.H

Input:

- Volume fraction field α_water
- Volume fraction threshold α_threshold
- Velocity field U and its gradient ∇U
- Interface normal field n_Inter = ∇α / |∇α|
- Droplet labels (Labels) and IDs (vofID)
- Physical parameters: σ (surface tension), μ_water, ρ_air, ρ_water, g (gravity)
- Contact angle parameters: θA, θR
- Mesh geometry: cell centers C[cell], volumes V[cell], face-to-cell connectivity

Procedure:

```
Initialize arrays for each droplet:
    viscousForce       ← (0, 0, 0)
    pressureForce      ← (0, 0, 0)
    totalDragForce     ← (0, 0, 0)
    adhesionForce      ← 0
    g_force_X          ← 0
    ratio              ← 0
    dA_TOT             ← 0

For each droplet index dropIdx in [1 .. dropNumber]:
    For each cell in the mesh:
        If α_water(cell) > α_threshold
           and α_water(cell) < 0.99
           and vofID(cell) = Labels[dropIdx] then

            Compute local interface normal vector n_Inter(cell)
```

Determine local interface area dA via geometric reconstruction:

- For 2D: plane $y = A \cdot x + D$
- For 3D: plane $z = A \cdot x + B \cdot y + D$
- Compute D using iterative procedure to match $\alpha_{water}(cell)$
- Find intersection between plane and cube edges
- Compute dA = intersection area
- Log warning file if intersection geometry is inconsistent

Compute local reference point for dynamic forces:

$p_{dyn\_point} \leftarrow C[cell] - k \cdot n_{Inter}(cell) \cdot l$ $(k = 3)$

Find closest cell to p_dyn_point to evaluate local properties:

$p_{dyn\_point\_cell} \leftarrow$ nearest cell center

$U_{local} \leftarrow$ velocity of closest cell

$\nabla U_{local} \leftarrow$ velocity gradient of closest cell

$viscousStress \leftarrow \mu_{water} \cdot \nabla U_{local}$

Compute local forces:

$p_{dyn} \leftarrow 0.5 \cdot \rho_{air} \cdot |U_{local}|^2$

$localPressureForce \leftarrow n_{Inter}(cell) \cdot p_{dyn} \cdot dA$

$localViscousForce \leftarrow (viscousStress \cdot n_{Inter}(cell)) \cdot dA$

Accumulate:

$pressureForce[dropIdx] \leftarrow pressureForce[dropIdx] + localPressureForce$

$viscousForce[dropIdx] \leftarrow viscousForce[dropIdx] + localViscousForce$

```
                    dA_TOT[dropIdx] ← dA_TOT[dropIdx] + dA

        Compute total drag force:

            totalDragForce[dropIdx] ← pressureForce[dropIdx] +
             viscousForce[dropIdx]


        Compute droplet contact area on wall (adhesion region):

            dA_adh ← Σ (l_adh²) for cells in contact with wall

        If parallel run:

            Sum all force and area contributions across processors.


        Compute adhesion force:

            D_adh ← sqrt(4 · dA_adh / π)

            adhesionForce[dropIdx] ← κ · 24σD_adh · (cos θR – cos
             θA) / π³


        Compute gravitational force component in X-direction:

            g_force_X[dropIdx] ← dropMass[dropIdx] · g_x


        Compute ratio between drag and adhesion forces:

            If adhesionForce[dropIdx] ≠ 0 then

                ratio[dropIdx] ← |(totalDragForce_x + g_force_X) /
adhesionForce|

            Else

                ratio[dropIdx] ← 999

                // Droplet not in wall contact
```

Output:

- viscousForce[dropIdx]: total viscous drag on each droplet
- pressureForce[dropIdx]: total pressure drag on each droplet
- totalDragForce[dropIdx] = viscousForce + pressureForce

- adhesionForce[dropIdx]: estimated wall adhesion force
- g_force_X[dropIdx]: gravitational force in X direction
- ratio[dropIdx]: (drag + gravity) / adhesion → indicator of droplet motion
- Optional warning files for geometric reconstruction inconsistencies

A.7 - printForces.H

Input:

- Current simulation time currentTime
- Output flag mesh.time().outputTime()
- Arrays (for each droplet):
  - pressureForce[dropIdx] → pressure contribution [N]
  - viscousForce[dropIdx] → viscous contribution [N]
  - g_force_X[dropIdx] → gravitational force component [N]
  - totalDragForce[dropIdx] → total drag in X-direction [N]
  - adhesionForce[dropIdx] → adhesion force [N]
  - ratio[dropIdx] → (drag + gravity) / adhesion ratio
- droplet count dropNumber

(Executed only by master processor)

Procedure:

```
If running on master processor then
    If currentTime = 0 or output time,
       and currentTime ≠ lastWrittenTime_2 then
        Update lastWrittenTime_2 ← currentTime

        For each droplet index dropIdx in [1 .. dropNumber]:
            Define fileName ← "ForcesOnDroplet_dropIdx.txt"
            Open fileName in append mode.

            If file is empty then
                Write header:
                    "Time [s]    Pressure Force [N]    Viscous
                      Force [N]
```

```
                    G Force X [N]    Total Drag X-dir [N]
                    Adhesion X-dir [N]    Ratio [-]"

                Append current data line:
                    currentTime, pressureForce[dropIdx],
                    viscousForce[dropIdx], g_force_X[dropIdx],
                    totalDragForce_x, adhesionForce, ratio

                Close file
```

Output:

- One text file per droplet:
  "ForcesOnDroplet_i.txt"
- Each file contains time history of:
  - Pressure, viscous, and gravitational forces
  - Total drag and adhesion in X-direction
  - Force ratio (drag/adhesion indicator)
- Written only at output timesteps

A.8 - lateralScaleFactor.H

Input:

- Field θ_dyn_OLD: contact angle values from previous timestep
- Initial static contact angle θ_0
- Interface normal field n_Inter
- Mean droplet velocities velocityMean[dropIdx]
- Face-to-cell connectivity (faceCells)
- Droplet labels (Labels)
- Field vofID identifying droplet cells
- Current simulation time mesh.time().value()

Procedure:

```
Initialize:
    factor[faces] ← 0       // Scaling factor for contact angle
    uwall[faces] ← 0        // Velocity of contact line (advancing/receding)

For each droplet index dropIdx in [1 .. dropNumber]:
    If current simulation time ≠ 0 then
        Set max_theta ← 0
        Set min_theta ← 180

        For each face on the patch:
            cellI ← faceCells[faceI]
            If vofID(cellI) = Labels[dropIdx] then
                Update:
                    max_theta ← max(max_theta, θ_dyn_OLD[faceI])
                    min_theta ← min(min_theta, θ_dyn_OLD[faceI])
```

```
If parallel run:
    Synchronize max_theta and min_theta across
    processors

Compute projections of interface normals:
    n_x_max ← sin(max_theta × deg2rad)
    n_x_min ← sin(min_theta × deg2rad)

For each face on the patch:
    cellI ← faceCells[faceI]
    If vofID(cellI) = Labels[dropIdx] then
        If n_Inter[cellI][x] ≤ 0 then
            // Advancing front (interface entering
             droplet)
            factor[faceI] ← |n_Inter[cellI][x] /
             n_x_max|
            uwall[faceI] ← –velocityMean[dropIdx][x]
        Else
            // Receding front
            factor[faceI] ← |n_Inter[cellI][x] /
             n_x_min|
            uwall[faceI] ← velocityMean[dropIdx][x]

        // Limit scaling factor
        If factor[faceI] > 1 then
            factor[faceI] ← 1

Reinitialize θ_dyn_OLD ← θ₀ for next timestep
```

Output:

- Field factor[faces]: lateral scaling factor for dynamic contact angle

- Field uwall[faces]: wall velocity of advancing/receding contact line
- Updated θ_dyn_OLD for next iteration

A.9 - contactAngleHysteresis.H

Input:

- Initial equilibrium contact angle θ_0
- Advancing and receding angles θA, θR
- Ratio[dropIdx] = (drag + gravity) / adhesion
- Arrays:
  - factor[faces] → lateral scaling factor
  - uwall[faces] → wall velocity (advancing/receding)
  - vofID[cell] → droplet cell identification
  - Labels[dropIdx] → droplet IDs
  - totalDragForce[dropIdx], adhesionForce[dropIdx], dropMass[dropIdx]
- θ_dyn_OLD: contact angle field from previous timestep

Procedure:

Initialize:

θ_dyn ← θ_0 for all patch faces

dropAlreadySlidingCheck[dropIdx] ← false // Flag for droplet motion state

For each droplet index dropIdx in [1 .. dropNumber]:

Define:

fileName_check ← "Adhesion_Check_dropIdx.txt"

fileName_dec ← "dropDeceleration_dropIdx.txt"

If master processor:

Delete fileName_check and fileName_dec (to be rewritten if droplet adheres)

If ratio[dropIdx] ≤ 1 then

```
// Droplet in adhesion regime
If master processor:
    Create empty file fileName_check  // Flag file for
    solver

    Compute droplet deceleration due to adhesion:
        dropDeceleration ← |(totalDragForce_x –
        adhesionForce)| / dropMass

    Open fileName_dec and write:
        If dropAlreadySlidingCheck[dropIdx] = false:
            Write 0
        Else:
            Write dropDeceleration

For each face in patch:
    cellI ← faceCells[faceI]
    If vofID(cellI) = Labels[dropIdx] then
        uu ← uwall[faceI]  // Contact line velocity

        If uu ≥ 0 then
            // Receding front
            θ_dyn[faceI] ← θ₀ – ratio[dropIdx] × (θ₀ –
            θR) × factor[faceI]
        Else
            // Advancing front
            θ_dyn[faceI] ← θ₀ + ratio[dropIdx] × (θA –
            θ₀) × factor[faceI]

        θ_dyn_OLD[faceI] ← θ_dyn[faceI]   // Update for
        next timestep
```

Output:

- θ_dyn[faces]: updated dynamic contact angle field
- Text files (master only):
  "Adhesion_Check_i.txt" – flag for adhesion condition
  "dropDeceleration_i.txt" – deceleration value passed to solver
- Updated θ_dyn_OLD for next timestep

A.10 - contactAngleCorrelations.H

Input:

- Physical parameters:
  - μ_water → dynamic viscosity of water
  - σ → surface tension
  - θ_0 → equilibrium contact angle
  - θA → advancing contact angle
  - θR → receding contact angle
- Fields:
  - uwall[faces] → contact line velocity
  - factor[faces] → lateral scaling factor
  - vofID[cell] → droplet ID field
  - Labels[dropIdx] → droplet labels
  - ratio[dropIdx] → (drag + gravity) / adhesion ratio
- θ_dyn_OLD: dynamic contact angle from previous timestep

Procedure:

```
Compute Capillary number for each face:
    Ca[face] ← (μ_water × uwall[face]) / σ

For each droplet index dropIdx in [1 .. dropNumber]:
    If ratio[dropIdx] > 1 then   // Droplet is sliding
        For each face on the patch:
            cellI ← faceCells[faceI]
            If vofID(cellI) = Labels[dropIdx] then
                u ← uwall[faceI]

                If u ≥ 0 then
```

```
                    // Receding front: Hoffman–Voinov–Tanner
                     relation
                    θ_rad ← θR × deg2rad
                    θ_temp ← ((θ_rad³ − 72 × Ca[faceI])^(1/3)) ×
                     rad2deg
                    Δθ ← θ₀ − θ_temp
                    θ_dyn[faceI] ← θ₀ − Δθ × factor[faceI]

                Else
                    // Advancing front: Kistler correlation
                    θ_rad ← θA × deg2rad
                    f_hoff_inv ← θ_rad³ / 72
                    x_hoff ← Ca[faceI] + f_hoff_inv
                    θ_temp ← acos(1 − 2 × tanh(5.16 × ((x_hoff /
                     (1 + 1.31 × x_hoff^0.99))^0.706))) ×
                     rad2deg
                    Δθ ← θ_temp − θ₀
                    θ_dyn[faceI] ← θ₀ + Δθ × factor[faceI]

                Update previous timestep field:
                    θ_dyn_OLD[faceI] ← θ_dyn[faceI]

            Mark droplet as sliding:
                dropAlreadySlidingCheck[dropIdx] ← true
```

Return:

θ_dyn → updated (dynamic) contact angle field

Appendix B - Auxiliary Geometric Functions

The present appendix collects and documents a set of auxiliary geometrical routines developed to support the evaluation of liquid-gas interfacial properties within the customized boundary condition described in 3.3.1.4.1. Although these functions are not directly part of the OpenFOAM dynamic contact-angle framework, they play a fundamental role in computing the instantaneous capillary and hydrodynamic forces acting on each droplet identified by the solver. Specifically, these functions are used to reconstruct the interface geometry at the cell level by approximating the local interface surface with a plane. This geometric reconstruction enables the model to evaluate:

- the orientation of the liquid-gas interface
- the local interfacial area
- the distribution of surface forces (pressure and viscous components).

The algorithmic workflow is structured into three main steps:

1. Determination of the interface plane (findInterfacePlane()):

   For each interfacial cell, the routine identifies the position of the plane that best reproduces the local volume fraction value, using a binary-search approach.

2. Evaluation of intersection points (findIntersection()):

   Once the plane is known, its intersection with the cubic control volume is determined by analysing all the cell edges, producing a list of vertices that define the interfacial polygon.

3. Computation of interfacial area (intersectionArea()):

   The final step calculates the polygonal area corresponding to the plane–cell intersection, either by direct geometric formulation (2D case) or by triangulation (3D case).

All functions have been implemented in a modular, stand-alone fashion, allowing independent testing and future reuse in other solvers or boundary conditions. Their inclusion as a separate appendix is motivated by the need to maintain readability and clarity in the main body of the thesis, while still providing a complete methodological record of the algorithms employed.

B.1 - findInterfacePlane()

This routine reconstructs the water-air interface within each computational cell by iteratively determining the plane offset Dthat yields the target liquid fraction. The algorithm performs a binary search over the interval $[D_{min}, D_{max}]$, refining the position of the plane until the fraction of points below the plane matches the prescribed phase volume fraction within a specified tolerance. The same logic is applied to both 2D and 3D cases, depending on the number of plane coefficients.

Input:

- Cell edge length l
- Plane coefficients:
  - A → slope in x-direction
  - B → slope in y-direction (for 3D case only)
- Search range for plane offset: [D_min, D_max]
- Maximum number of iterations: maxIterations
- Convergence tolerance: tolerance
- Target volume (or area) fraction: targetAlpha
- PointCloud: set of discretized points within the cell

Procedure:

```
    Initialize:
        D ← (D_max + D_min) / 2          // Initial guess for plane
offset


    For each iteration in [1 .. maxIterations]:
        pointsBelow ← 0
        pointsTotal ← 0


        For each point ∈ PointCloud:
            Read coordinates (x, y, z) or (x, y)
            Compute plane position:
```

```
            2D case: y_plane = A·x + D
            3D case: z_plane = A·x + B·y + D
        If (point lies below plane):
            pointsBelow ← pointsBelow + 1
        pointsTotal ← pointsTotal + 1

    Compute current volume (or area) fraction:
        α_temp ← pointsBelow / pointsTotal
    Compute error:
        err ← α_temp – targetAlpha

    If |err| ≤ tolerance:
        Break iteration loop   // Converged

    If err > 0:
        D_max ← D      // Overshoot: plane too low
    Else:
        D_min ← D      // Undershoot: plane too high

    Update:
        D ← (D_max + D_min) / 2
```

Return:

- D // Offset coefficient giving correct plane position

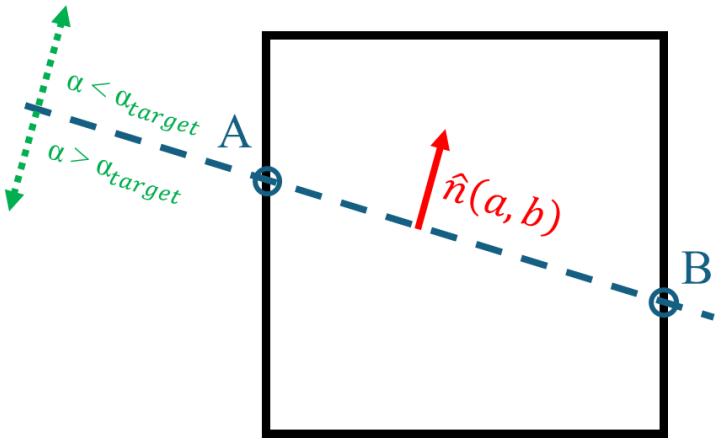


Figure A1. The binomial method starts from an initial guess and iteratively search for the plane that best approximates the interface

B.2 - findIntersection()

This routine determines the intersection points between the reconstructed interface plane and the cell edges, both in two and three dimensions. The intersection is computed by solving the plane-edge equation parametrically for each edge, ensuring that only points within the cell bounds are retained. Duplicate intersections, which may occur when the plane passes exactly through a vertex, are filtered out to maintain geometric consistency.

Input:

- Plane coefficients:
  - 2D case: A, D → plane equation $y = A \cdot x + D$
  - 3D case: A, B, D → plane equation $z = A \cdot x + B \cdot y + D$
- EdgeList: list of edges representing the cell geometry
  (each edge defined by two vertices p_1 and p_2)

Procedure (common logic for both 2D and 3D cases):

Initialize:

intersections ← empty list

For each edge in EdgeList:

Extract endpoints p_1 and p_2

Define parametric point along the edge:

$p(t) = p_1 + t \cdot (p_2 - p_1)$, where $0 \leq t \leq 1$

Substitute p(t) into plane equation and solve for t:

2D: $y_1 + t \cdot (y_2 - y_1) = A \cdot (x_1 + t \cdot (x_2 - x_1)) + D$

3D: $z_1 + t \cdot (z_2 - z_1) = A \cdot (x_1 + t \cdot (x_2 - x_1)) + B \cdot (y_1 + t \cdot (y_2 - y_1)) + D$

Compute:

t_num = $A \cdot x_1 + B \cdot y_1 + D - z_1$ (or y_1 for 2D)

t_den = $(z_2 - z_1) - A \cdot (x_2 - x_1) - B \cdot (y_2 - y_1)$

If t_den = 0 → no intersection, continue

t = t_num / t_den

If $0 \leq t \leq 1$ then

Compute intersection point:

p_int = $p_1 + t \cdot (p_2 - p_1)$

Check for duplicates:

If p_int is not within tolerance of existing points

→ append p_int to intersections list

Return:

- Intersections // List of intersection coordinates between plane and cell edges

B.3 - intersectionArea()

This algorithm evaluates the interfacial area between the liquid and gas phases within each computational cell. In two dimensions, the area is computed directly from the intersection segment and the cell height. In three dimensions, the intersection polygon is decomposed into triangles whose areas are summed after sorting the vertices in counterclockwise order. This geometric procedure ensures a consistent and accurate reconstruction of the water–air interface for force and curvature calculations.

Input:

- intersections: list of intersection points between plane and cell edges
- l: cell edge length (for 2D case only)

```
Procedure (2D case):
    If dimension = 2:
        Compute base length:
            base = √[(x₁ − x₂)² + (y₁ − y₂)²]
        Compute height = l
        Interface area:
            dA = base × height
        Return dA

Procedure (3D case):
    If dimension = 3:
        If number of intersection points = 3:
            // The plane cuts a triangular area
            dA = area of single triangle

        Else:
            // Determine reference vertex (lowest x-coordinate)
```

```
refPoint ← point with minimum x among intersections


// Sort intersection vertices counterclockwise around refPoint
Sort(intersections) by polarAngle(p, refPoint)


// Decompose polygon into triangles depending on number of vertices
If 4 vertices → quadrilateral:
    dA = area(A,B,D) + area(B,C,D)
If 5 vertices → pentagon:
    dA = area(A,B,E) + area(B,C,D) + area(B,D,E)
If 6 vertices → hexagon:
    dA = area(A,B,C) + area(A,C,D) + area(A,D,F) + area(D,E,F)


Each triangle area computed as:
    AB = B − A
    AC = C − A
    cross = AB × AC
    dA_triangle = 0.5 × |cross|
```

Return

- total dA

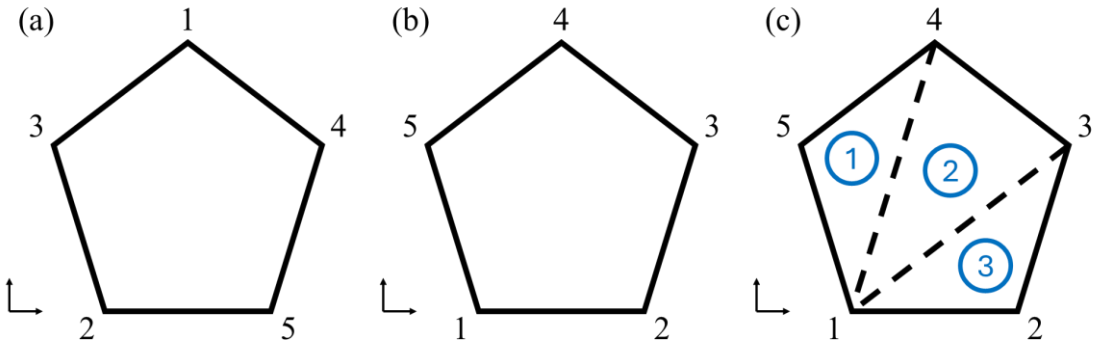


Figure A2. General algorithm to evaluate interface area; the initial point list (a) is ordered in anti-clock manner (b) and then the total area is evaluated as a sum of triangles (c)