ETSEIAT

GrETA
Grau en Enginyeria en Tecnologies Aeroespacials

Numerical study of fluidic oscillators with compressible flow

Author: Pau Guilleumes Mayoral

Director: David del Campo Sud

Co-director: Vanessa del Campo Gatell

Delivery date: 10-01-2017

REPORT

Contents

List of Tables

List of Figures

1 Aim of the study

A comprehensive study of the flow in a fluidic oscillator will be made, using an open source code of computational fluid mechanics (CFD) called OpenFOAM®, giving special attention to the compressible flow behaviour. A constant number of Reynolds will be used in each case, whereas some changes in key points of the geometry will be made, in order to analyse the variations in the different parameters of the simulation results. It is important to remark that this is not an isolated study, but follows an investigation line carried out by the Aerospace Division of Terrassa, of the Physics Department of the Universitat Politècnica de Catalunya, so all the data obtained in this study will be available for further investigation in this field.

2 Scope of the study

The different objectives that must be fulfilled in order to reach the scope of the study are the following:

- Documentation, research and study of the state of the art regarding fluidic oscillators
- Apprenticeship of the basic concepts of computational fluid dynamics.
- Apprenticeship of the OpenFOAM® software

  Lecture of OpenFOAM® guides

  Fulfillment of OpenFOAM® tutorials
- Fluidic oscillator 3D modelling
- Fluidic oscillator meshing using OpenFOAM® tools
- Execution of the simulations
- Extraction, analysis and validation of the simulation results, and comparison to simulation of incompressible flow
- Geometrical parameters variation

  Modelling

  Meshing

  Simulation
- Conclusions
- Contribution to the ETSEIAT investigation line in this field

3 Requirements of the study

The main requirement of this study is to get the most accurate numerical results in the different simulation cases. This means that a very fine mesh has to be used, whose simulations demand a great computational capacity, which will be provided by the cluster available in the ETSEIAT facilities.

It is necessary as well a deep knowledge of fluid mechanics and the CFD software in order to be able to run proper and decent simulations.

The economical requirements are not as important due to the open source nature of the software used. Nevertheless, the electric consume of all the devices such as the cluster or every computer used during the whole process has to be taken into account.

4 Justification

4.1 Identification of the need

Nowadays, in the aeronautics field, one of the main objectives that settle the different lines of investigation and research around the world, is the maximisation of the flight efficiency, rather than the velocity or power, as it had been some years before. A way to accomplish this objective of improvement is using flow control devices in the different aerodynamic surfaces such as the wings or the turbine vanes, in order to control on our will the boundary layer and the laminar to turbulent transition, so that the performance can be improved. This is where the fluidic oscillators become more and more important. Also, its design simplicity and the fact that they have no moving parts, play an important role.

But the application of this device are not limited to the improvement of the flight performance. It can be used in a wide range of fields, such as the motoring world, where it can work as in aeronautics for the boundary layer control, but also for improving the combustion efficiency, among many others.

Gathering together all those reasons, it becomes clear the importance that this field of research has recently gained.

4.2 Usefulness of the project

Most of the recent numerical studies related to fluidic oscillators (including the research done by the Aeronautics department of the UPC ETSEIAT) have been carried out on a bi-dimensional (2D) environment. Moreover, it has been studied under incompressible conditions. This project is aimed to go further and validate, compare and analyse results of a 3D compressible study with the experimental results previously obtained by the Technische Universität Berlin, the study on which this project is based.

Apart from this reason, there is also a purpose of demonstrating the good performance and the enormous potential that an open source cfd tool such as OpenFOAM® has, because most of the previous numerical studies on the subject have been done with ANSYS Fluent®.

5 Background

5.1 Fluidics

Fluidics (fluid logic) is a relatively new technology that studies circuitries or elements without any mechanical moving part. It uses the interaction of different fluid (liquid or gas) streams on its purpose: the fluid flow or pressure is used as a power supply medium, to provide the user with controlling functions, sensing and also computing tasks, as well as many others.

The first patent of a device which working principle was based on fluidics was issued by Nikola Tesla on 1916 for the "Valvular valve" (figure 5.1). Tesla's valve offers greater resistance in one direction than in the other direction (10 to 200 times greater), due to enlargements and paths in different directions that offer virtually no resistance in one direction, but work as a barrier in the opposite direction. Nevertheless, fluidics as it is currently known, was properly discovered by a group of scientists at the U.S. Army Harry Diamond Laboratories (HDL) [21] and it was initially called fluid amplification. Several phenomena was studied such as Prandtl's boundary layer, the Coanda Effect on a jet flow, the transition from laminar to turbulent, as well as many others, and even now there are still numerous unknowns on the fluids behaviour that have not already been explained.

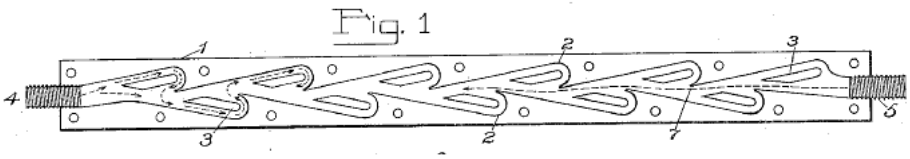

Figure 5.1: Tesla's Valvular Valve patent US-1329559 [3]

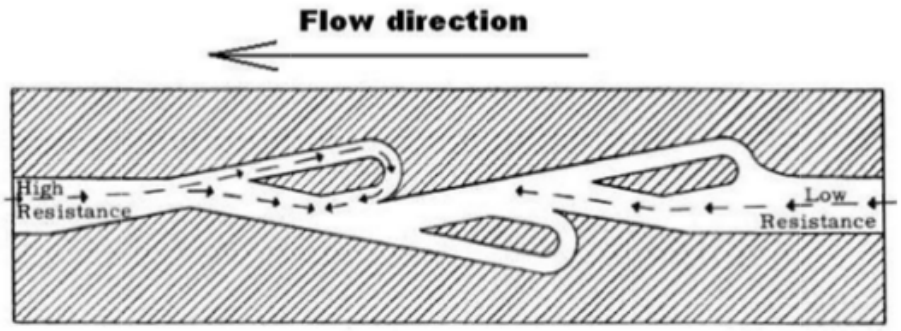

Figure 5.2: Tesla's Valvular valve principle of operation, from US patent US-1329559 [3]

5.1 Fluidics

At the beginning, they used to try to implement fluidics in almost every field, without even considering if it was more advantageous than the existing technologies. This period lasted 10 years and reached the peak in de middle of the 60s, and the majority of devices of this so-called "first generation" were operated in a turbulent flow regime. Nowadays, although it is no longer a novelty, it is treated as it deserves because of its own merits. Since 1970, the main interest has shifted into using laminar flow instead of turbulent, and several truly valid applications have been realized in a wide range of fields including medicine, personal-use items, industrial automation for factories, and last but not least, aerospace industry, and its close relative, the military industry.

Fluidic elements can be categorised, in general, as digital or analogue devices. Digital devices can only give two different outputs, i.e., 1 and 0, whereas analogue devices can generate a varying output signal, dependent of the control and the input signal. Among all the fluidic devices, two main types can be differentiate:

- Sensors
- Actuators

The main difference between an actuator and a sensor is that the actuator is by definition intrusive, whereas the sensor is ideally non-intrusive. Regarding the actuator, an electrical signal as an input produces as an output a flow disturbance. The opposite process occurs in a sensor: the input is a flow disturbance, and the output is an electrical quantity.

Actuators can be used in a fluidic discipline called active flow control (AFC) technology, which is becoming one of the most important fields in aerodynamics. It can be used in several applications, and a huge amount of actuators are currently being studied all around the world under this discipline. Some of the examples of AFC actuators are plasma-fluidic oscillators, powered resonance tubes (PRTs), piezoelectic actuators, synthetic jets, chemical-based actuators, and many others. These actuators provide control over different types of flow providing a steady, oscillating or pulsating acoustic, momentum or mass flow field, depending on the device.

According to Cattafesta and Sheplak (2011) [4], those flow control actuators can be classified as the following figure shows:

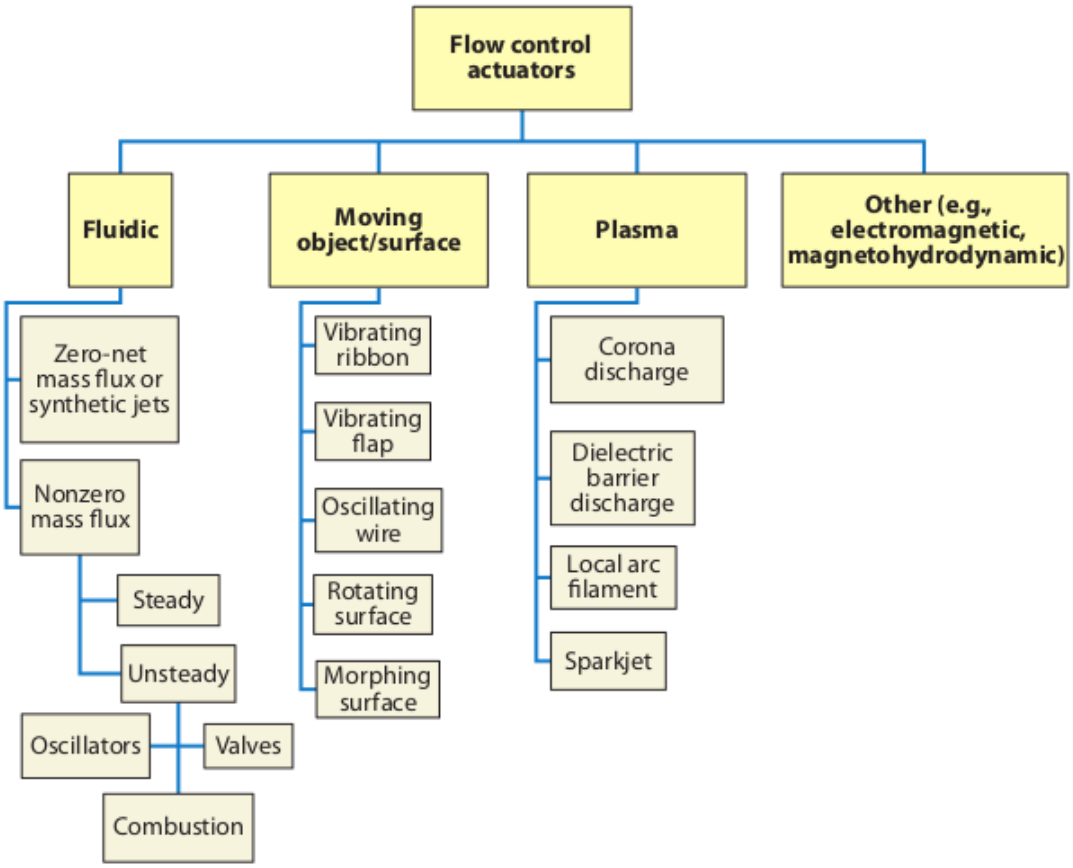

Figure 5.3: Classification of AFC actuators, from Cattafesta, L. N. [4]

The advantages and disadvantages of the fluidic flow control actuators can be found in table 5.1

Table 5.1: Advantages and disadvantages of fluidic flow control actuators

Type	Subtype	Advantages	Disadvantages
Fluidic			
	ZNMF	Requires no external fluid source	Peak velocities typically limited to low to moderate subsonic speeds
		Amenable to various types of drivers and sizes	Resonant devicies
		Suitable for feedback control	
	Valves	Capable of high velocities with either fast time response or high bandwidth but generally not both	May not be amenable to feedback control
			Requires an external flow source
	Oscillators	Capable of producing large disturbances	Standard versions not suitable for feedback control
		Amenable to a range of sizes and hence frequencies	Requires an external flow source
		Potential extensions possible to enable independent control of frequency and velocity	
	Combustion	Capable of producing large perturbations in high-speed flows	Currently limited to relatively low frequencies (a few hundred hertz)
			Requires combustion

5.2 Fluidic oscillators

Fluidic oscillators are a major advance in the active flow control field. They belong to the most common type of ACF actuators, the fluidic actuators, which uses fluid suction or injection to increase momentum in the local flow field. They have been increasingly often used in numerous different applications, such as the boundary layer control on either wing and turbine airfoils 5.4, high-lift devices, and all kind of aerodynamic surfaces, but they have been commonly used for several years on automobiles as a wind-shield washer fluid dispenser [6].

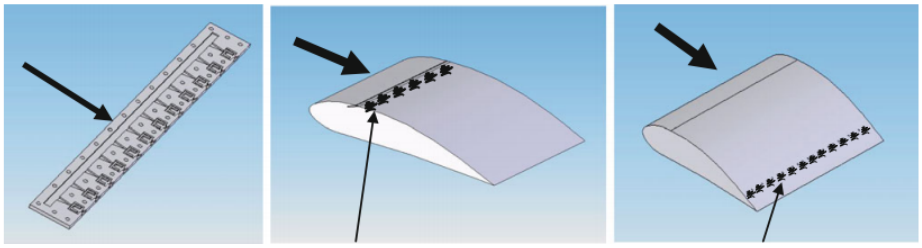

Figure 5.4: Fluidic array (a) integration into the leading edge (b) or the trailing edge (c) of an airfoil, from Raghu, S. [5]

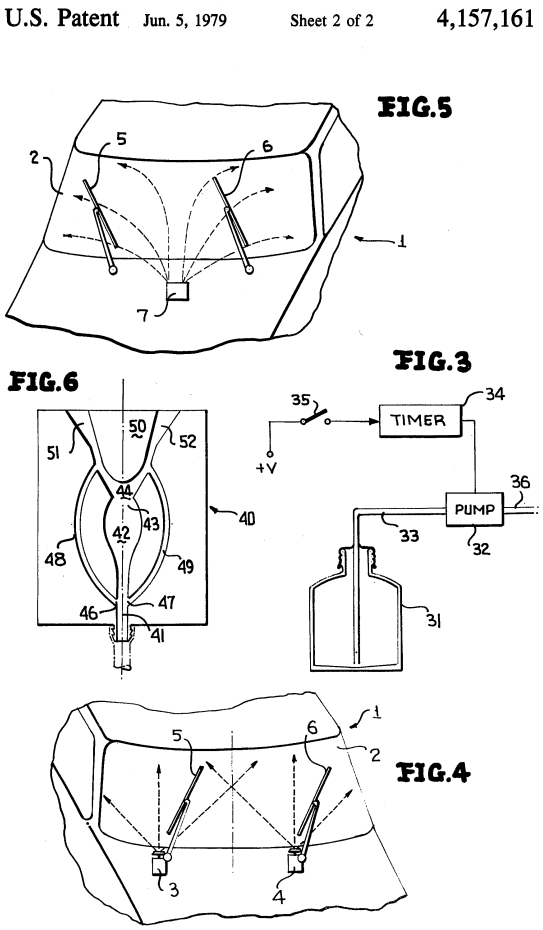

Figure 5.5: Windshield fluid dispenser, US Patent 4157161, from Bauer, P. [6]

Fluidic oscillators generate a periodic oscillating jet, when it is supplied with a fluid at a certain pressure and velocity. That is to say, its working principle is based on the bi-stable states of a single jet or a pair of jets of fluid inside a specially designed flow chamber. These produce sweeping or pulsing jets on the outlet. [5]

The main advantage of a fluidic oscillator against other flow control devices is that it has no moving parts: the oscillating jet is created only by means of fluid-dynamic interactions. That makes it much more plausible to be used into flight vehicles rather than any other AFC system, because it has a reduced risk of failure, as well as the costs and the complexity of the device.

There is three different types of fluidic oscillators:

1. Bi-stable fluidic oscillator
   (a) Double feedback loop
   (b) Single feedback loop
2. Feedback free fluidic oscillator

5.2.1 Bi-stable fluidic oscillators

The bi-stable fluidic oscillators, or wall attachment devices, are based on the fluid jet attachment to an adjacent wall in a divergent nozzle or in an internal chamber, provided by the Coanda Effect (explained in section 7). The attached flow can be released and attached to the opposite wall if a feedback mechanism is used. A pressure pulse perpendicular to the attached jet from the control port induces an unattachment and the following attachment to the opposite adjacent wall due to the recirculation bubble created next to the inner blocks wall, and the process is repeated with a pressure pulse from the other control port, resulting in an oscillating flow. This behaviour was first discovered by [22]. The exiting jet can be divided by a plate in the middle, creating in this way two pulsing jets. If the control port flow comes from the outlet nozzle, it is called feedback channel or loop.

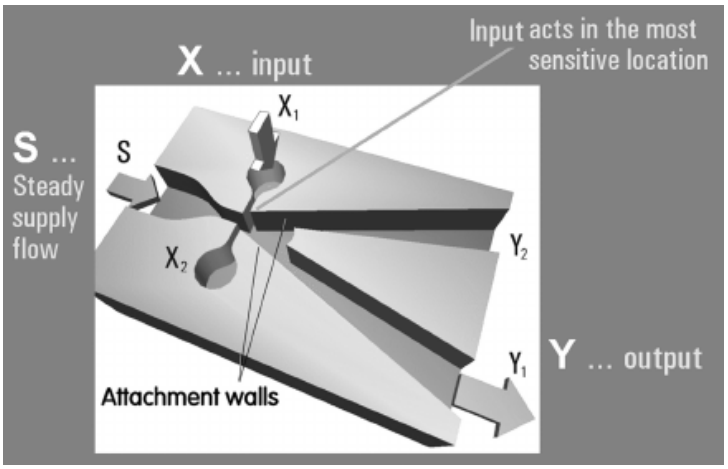

Figure 5.6: Typical bi-stable fluidic oscillator configuration, from Tesař, V. [7]

5.2.1.1 Double feedback loop fluidic oscillators

This is one of the oldest fluidic oscillator designs, originally discovered at the Harry Diamond Laboratories (HDL) [21]. It has been widely used since then in multiple applications. It is also called "seeping jet actuator" because of the characteristic flow at the outlet, or "Warren's circuit", due to the denomination of the feedback channels, called Warren feedback loops [8]. Part of the flow is removed from the output (or at the end of the mixing chamber) and brought back upstream by the feedback channel into the control port nozzle of the same side. Recent investigations are implementing it in arrays for flow control purposes with air, after some tests a V-22 wing showed promising results [23].

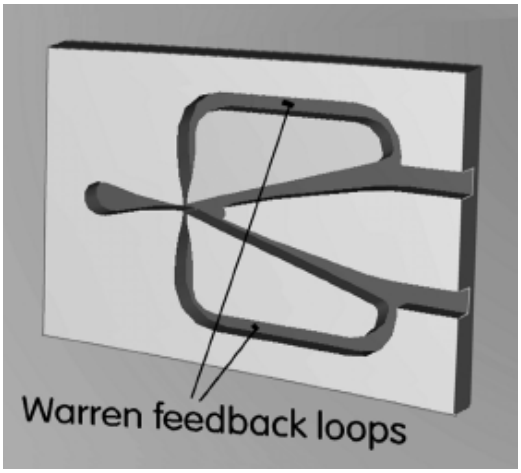

Figure 5.7: Double feedback loop fluidic oscillator configuration, from Warren, W. [8]

5.2.1.2 Single feedback loop fluidic oscillators

It is, by far, less used than the double feedback configuration. The single feedback loop fluidic oscillator uses the pressure difference between the two control ports when the jet is deflected, in order to generate the recirculating bubble that unattaches the main flow from the wall and create the oscillations. It was presented by Spyropoulos [9] (this is where the loops name come from, figure 5.8) but the device was patented by Warren.

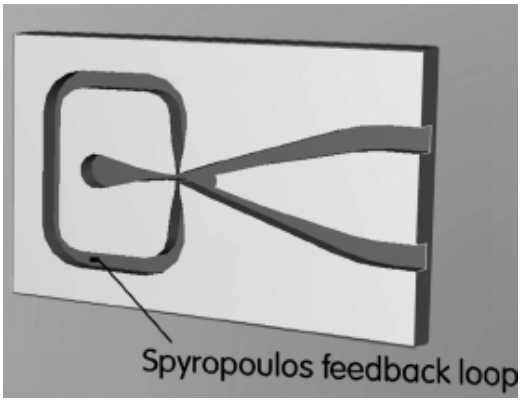

Figure 5.8: Oscillator with single feedback loop, from Spyropoulos, C. E. [9]

5.2.2 Feedback free fluidic oscillators

On the other hand, there is the feedback free fluidic oscillators, or jet interaction devices. These come from recent studies and research, and consist in the interaction of two fluid jets inside the mixing chamber, specially designed to achieve the desired properties of the oscillator (frequency, amplitude, etc.). The power nozzles are connected to at least one source of fluid under pressure. The exit throat can be off axis (off the central axis of the symmetry) by a small fraction to the left or right to move the leftward or rightward yaw angles in the spray.

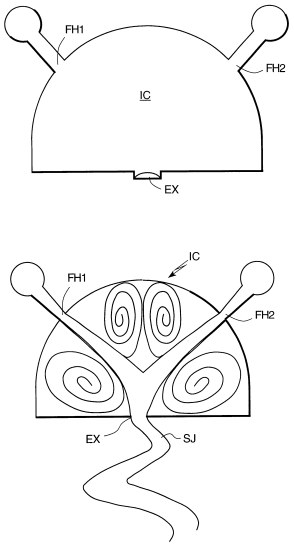

Figure 5.9: Feedback free fluidic oscillator schemes, US Patent 6253782 B1, from Raghu, S. [10]

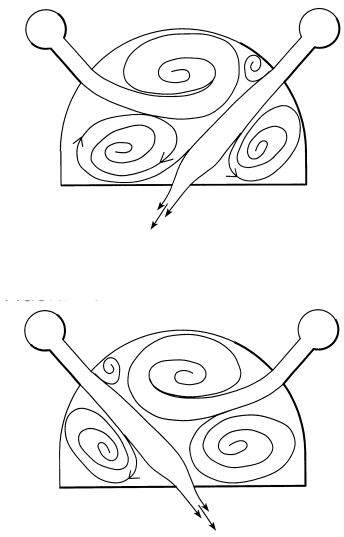

Figure 5.10: Feedback free fluidic oscillator phases 0° and 180°, US Patent 6253782 B1, from Raghu, S. [10]

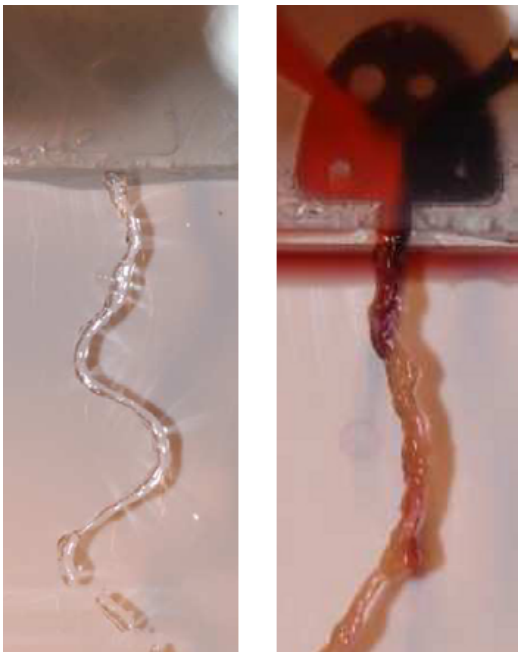

Figure 5.11: Feedback free fluidic oscillator with water supply, from Gregory, S. W. et al. [11]

6 State of the art

Several studies have been done on fluidic actuators, a lot of them performing numerical or CFD models, under laminar conditions as well as considering a turbulent flow.

6.1 TU Berlin investigation

The most recent work which this study is based on was carried on by the Technische Universität Berlin, leaded by Bernhard Bobusch [1]. It is either a numerical and experimental study.

The methodology used in the experimental part was a PIV (particle image velocity), used in fluidics with the purpose of visualisation instantaneous velocities and other physical magnitudes. Even though in his applications in the aeronautic field (for flight vehicles) the inner flow of the device is air, this study was made with water due to its superior optical characteristics, in order to make easier the data acquisition (it makes the measurements with laser easier to make). In figures 6.1 and 6.2 the experimental setup can be appreciated. The flow comes through a pipe pumped by a small pump from the aquarium and enters into the device, made of acrylic glass, placed in the water tank 10cm underneath the fluid surface. A CCD-Camera together with a laser, and a hydrophone are used for the data acquisition.

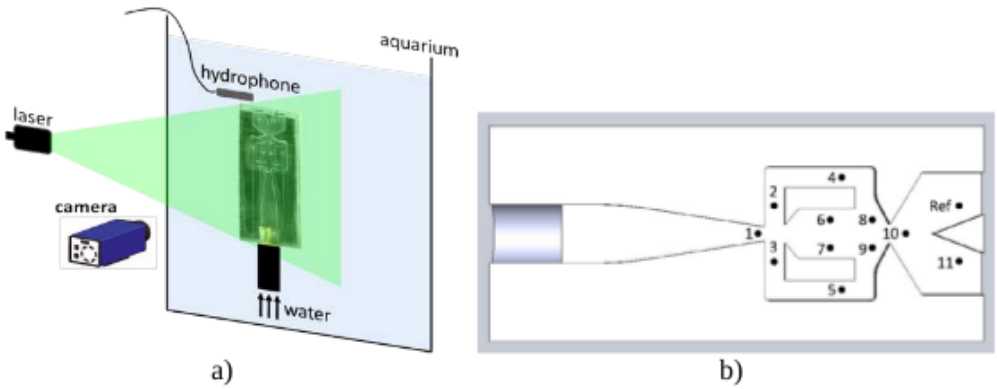

Figure 6.1: a) Experimental sketch of the test rig used. b) Pressure ports distribution. From Bobusch, B. [1]

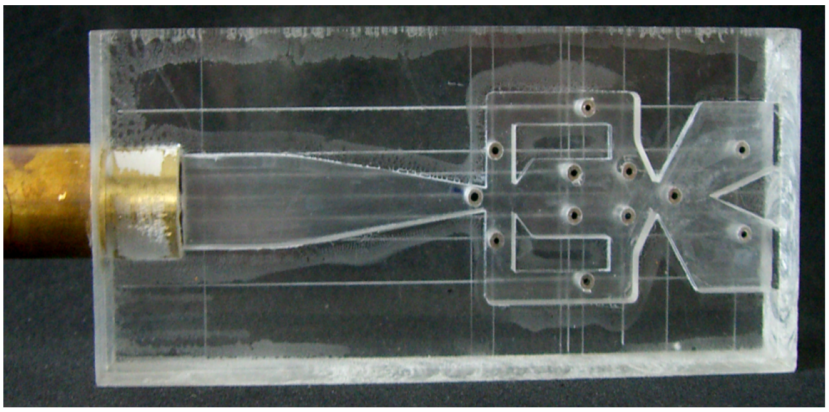

Figure 6.2: Experimental setup photograph from Bobusch, B. [1]

In figure 6.1 b) and 6.2, it can be seen the pressure ports around the mixing chamber, the feedback channels and the outlets where data acquisition is made, and where velocity distribution comparisons are considered.

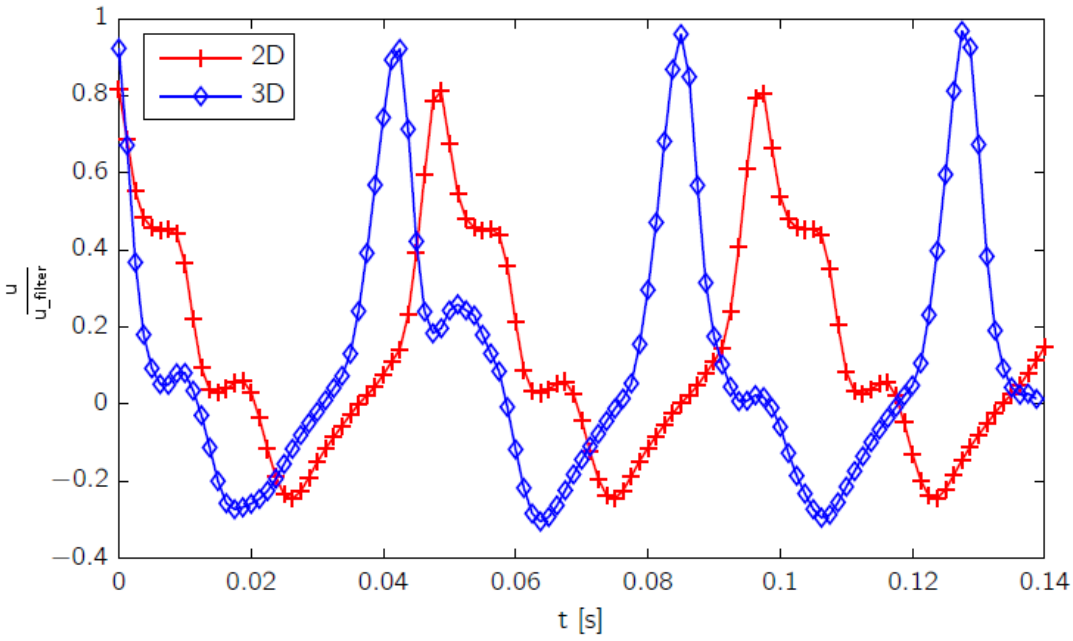

Figure 6.3: 3D and 2D CFD model comparison, adapted from Bobusch, B. [1]

Regarding the numerical approach of Bobusch, B. [1], a bi-dimensional and a three-dimensional mesh were used (to find out the significance of the differences between the two models - figure 6.3), and the simulations were carried out using ANSYS Fluent®. At the beginning, three meshes of different density were designed:

- High density mesh: 442986 nodes
- Medium density mesh: 166782 nodes
- Low density mesh: 41000 nodes

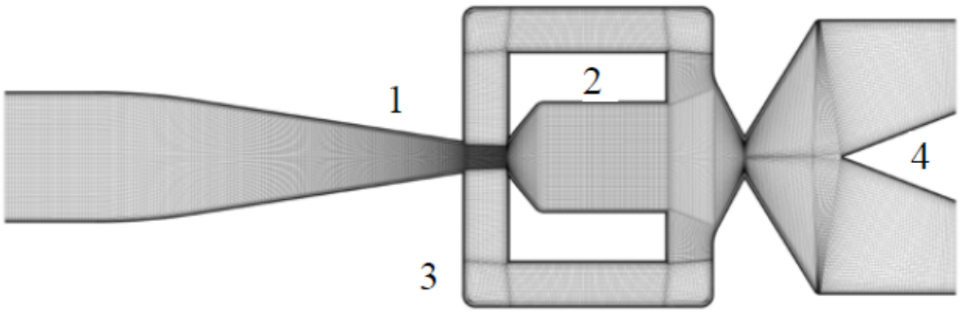

Figure 6.4: Medium density 2D mesh, 166782 nodes, from Bobusch, B. [12]

After validating the results of each one of the meshes, and doing the computational balance, a conclusion was reached that the optimal mesh was the medium density mesh (figure 6.4). In order to perform the simulations, various turbulence models were tested and compared in terms of accuracy:

- Spallart-Almaras model
- Reynolds Stress model (RSM)
- Shear Stress Transport (SST)
- Scale Adaptative Simulation (SAS)
- Detached Eddy Simulation (DES)

The difference between the frequency values obtained with each turbulence model can be appreciated in the following table 6.1. The optimal turbulent models are the Shear Stress model (SST) and the Detached Eddy Simulation (DES). It was finally decided to use the SST model due to the shorter time required for computing.

Turbulent model	**Frequency (Hz)**
Spallart-Almaras model	19.2
Reynolds Stress model (RSM)	20.7
Shear Stress Transport (SST)	24.4
Scale Adaptative Simulation (SAS)	20.7
Detached Eddy Simulation (DES)	24.2
Experimental measurement	**21.8**

Table 6.1: Oscillatior frequency comparison between the turbulent models, from Bobusch, B. [1]

The boundary conditions of the numerical experiment were set to match with the experimental analysis, so that the results of both experimental and numerical approaches can be compared:

Temperatures			
		Temperature °C	
Hydrophone		19.7	
Pressure ports		11.9	
Calibration sensor		15.5	
Second sensor		14.1	
Reynolds, volumetric flow rates and frequency results			
Volumetric flow rate (L/h)	Reynolds	Experimental freq. (Hz)	Numerical freq. (Hz)
80.9	8711	12.9	12.3
103.5	11152	15.5	15.1
126.3	13593	18.7	17.9
148.8	16034	21.8	20.7

Table 6.2: Boundary conditions, experimental and numerical results obtained by Bobusch, B. [1] (SST turbulence model)

It can be noticed that the input flows are all turbulent, because the minimum Re is 8711. Also, the interdependency between the Reynolds number and the outlet frequency is linear, as the following figure shows.

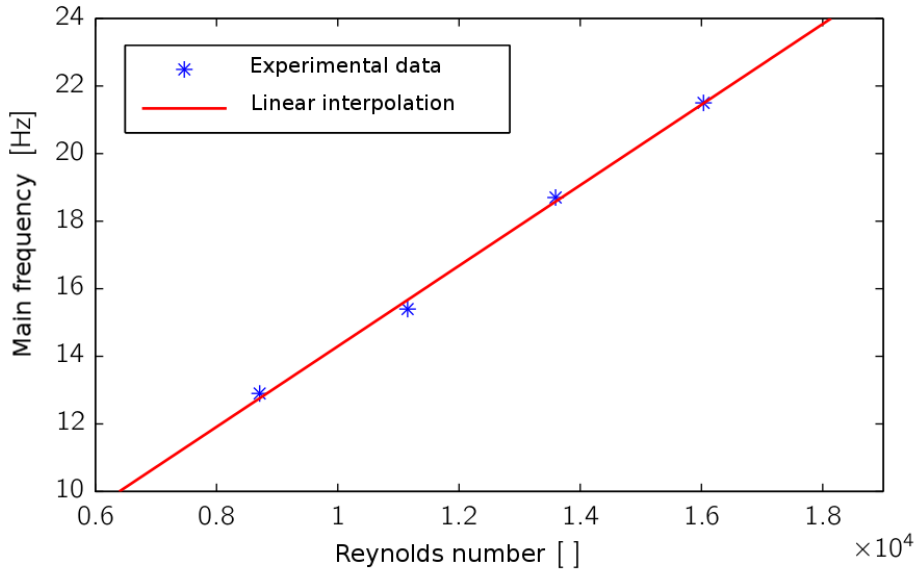

Figure 6.5: Linear dependence between Re and frequency, adapted from Bobusch, B. [12]

- Linear regression of experimental measrements: $f = 0.00122491 * Re + 2.30786$ (regression factor of $R^2 = 0.998$
- Numerical results function: $f = 0.00114707 * Re + 2.30786$ (regression factor of $R^2 = 1$

The linear regression does not apply at low Reynolds numbers, because the oscillator only works in turbulent regime.

6.2 ETSEIAT investigation

In addition, as it has been said before, an extensive research has been also made by the Aerospace Division of Terrassa of the Physics Department of the Universitat Politècnica de Catalunya, and this study intends to continue the investigation done by other students doing either Bachelor, Master and PhD final thesis, as well as the independent investigation of the researchers of this department.

6.2.1 Bi-dimensional and three-dimensional studies

Ruiz, M. [2] work was based on the same oscillator than Bobusch, B., and was focused on validating numerically results by repeating the same procedure, in order to set up a basis for the future research of UPC ETSEUAT Aerospace department. A 2D mesh of 121551 nodes

was built (lower number of nodes than Bobusch, B. mesh), and various turbulence models were used in order to compare them:

- SST
- k-epsilon
- k-omega
- RSM

After concluding that the best turbulence model was again the SST model, the numerical results obtained for different Reynolds numbers are the following:

Reynolds	**Frequency (Hz)**
8509.1	10.2
15376	15.86
30752	29.63
51254	45.073

Table 6.3: Numerical results obtained by Ruiz, M. [2]

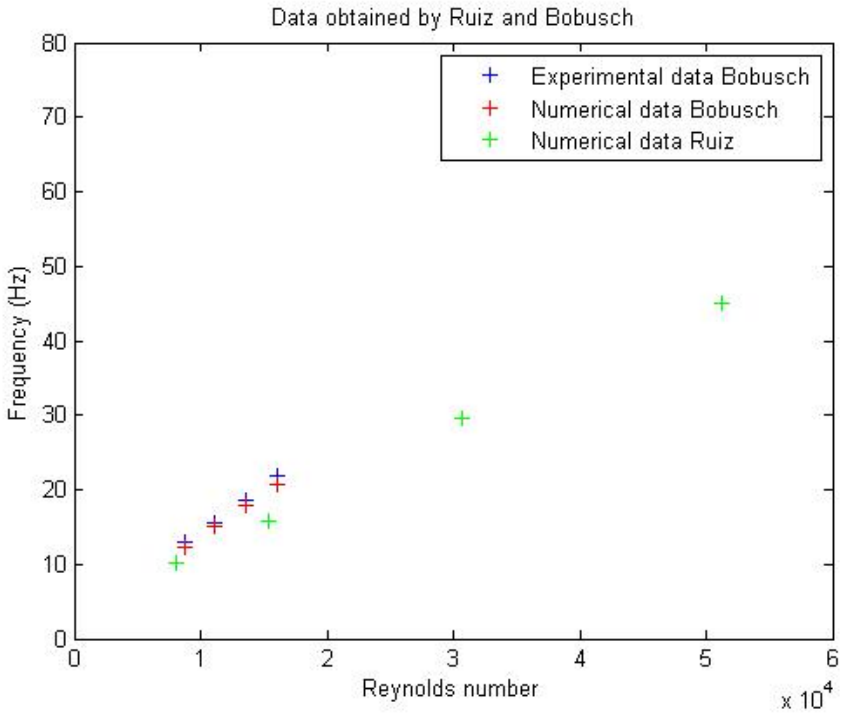

Figure 6.6: Comparison between Bobusch, B. numerical and experimental results and Ruiz, M. numerical results, from Ruiz, M. [2]

The reason why Ruiz, M. results are less accurate than Bobusch's is probably that the mesh was simply better in Bobusch, B. work, but also that it consists on a 2D case. Despite that, the linear regression is still visible, although at high Re loses accuracy. This is also probably due to the mesh, that was not fine enough to present a better result at high Re, where it needs to have higher density.

Calviño, P. carried out the first 3D study on ETSEIAT. The 3D mesh was extruded from the 2D mesh in which the study was based. The final mesh had a distribution of 12 layers on the extruded z-direction, as it can be seen in figure 6.7. In figure 6.8, it is easy to notice that the numerical 3D values of the simulation made by Calviño, P., are much closer to the experimental regression found by Bobusch, B. than the 2D values of Ruiz, M. study.

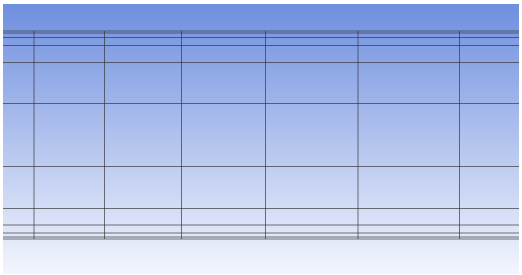

Figure 6.7: Z-direction 12 layers distribution, from Calviño, P. [13]

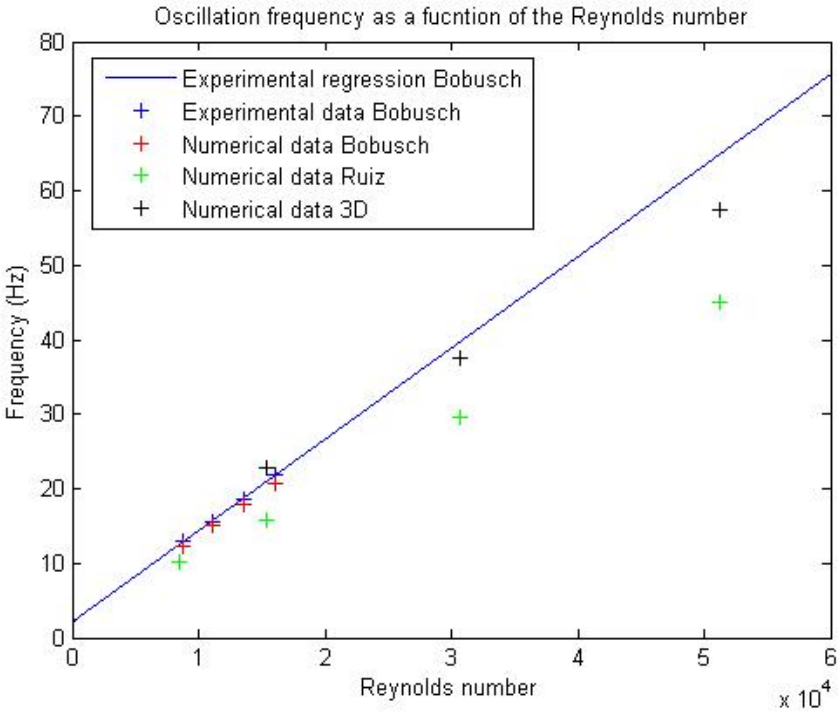

Figure 6.8: Comparison between experimental and numerical (2D and 3D) results from Calviño, P. [13]

6.2.2 Parameters variation studies

Saramiento, M. [14], and Gil, P. [24] together with Candelo, C. A. [15] developed projects where some modifications on the geometry where introduced, in order to compute and analyse the variations on the results.

The geometry modifications implemented by Saramiento, M., and its numerical results, are displayed respectively in figures 6.9, 6.10, 6.11, 6.12, 6.13.

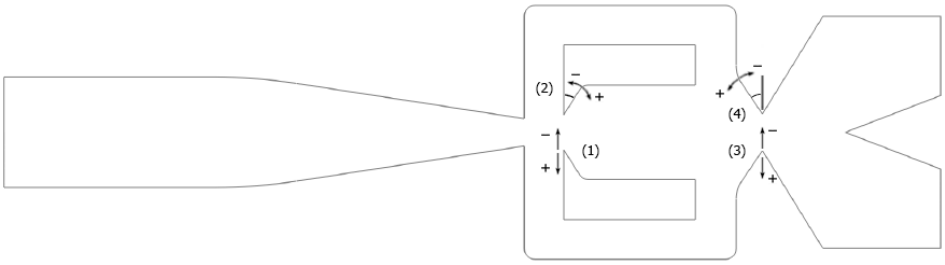

Figure 6.9: Geometry modifications of the fluidic oscillator, from Saramiento, M. [14]

Where:

1. Inlet throat amplitude
2. Inlet throat angle
3. Outlet throat amplitude
4. Outlet throat angle

It can be noticed that the frequency has a parabolic dependence with the inlet throat amplitude, and an inverse linear dependence with the outlet throat angle (probably due to the Coanda Effect on the outlet throat side walls).

Regarding Gil, P. [24] and Candelo, C. A. [15] work, the parameter that was changed was the outlet throat angle (figures 6.14 and 6.15). Although the values differ from the ones obtained by Saramiento, M., the inverse linear regression tendency remains still (figure 6.16).

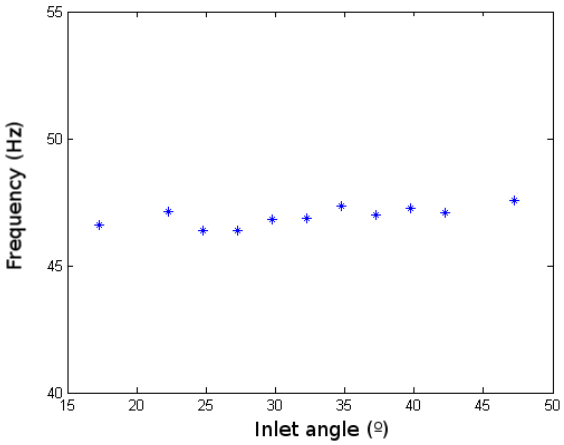

Figure 6.10: Inlet throat angle versus frequency, adapted from Saramiento, M. [14]

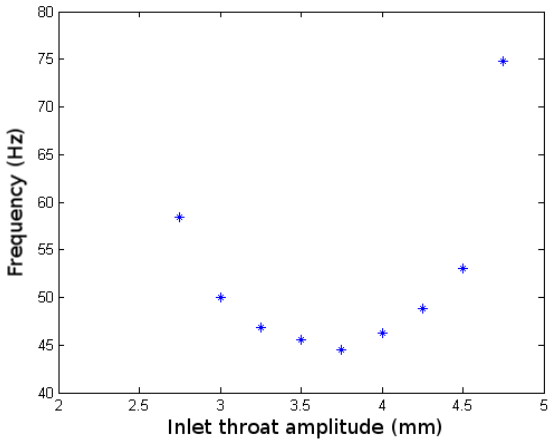

Figure 6.11: Inlet throat amplitude versus frequency, adapted from Saramiento, M. [14]

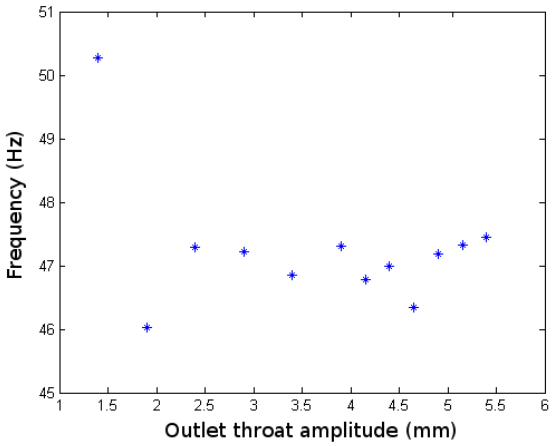

Figure 6.12: Outlet throat amplitude versus frequency, adapted from Saramiento, M. [14]

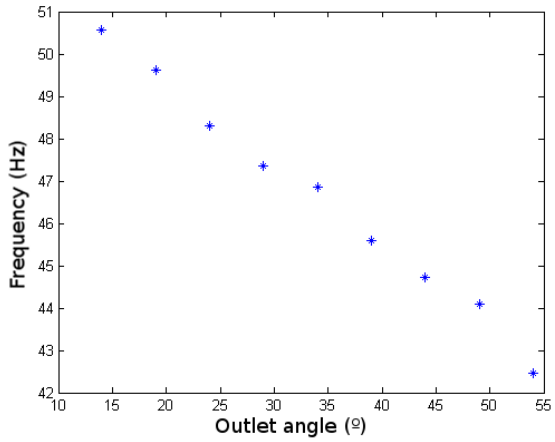

Figure 6.13: Outlet throat angle versus frequency, adapted from Saramiento, M. [14]

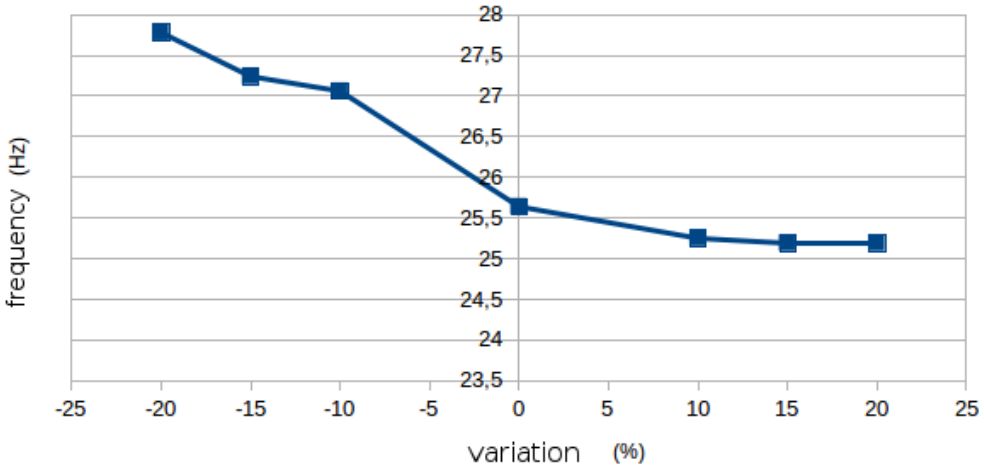

Figure 6.16: Frequency of the oscillator against the angle variations in %, adapted from Candelo, C. A. [15]

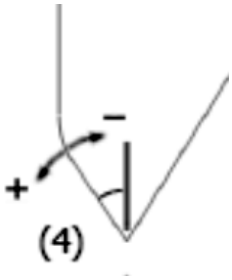

Figure 6.14: Outlet throat angle sketch, from Candelo, C. A. [15]

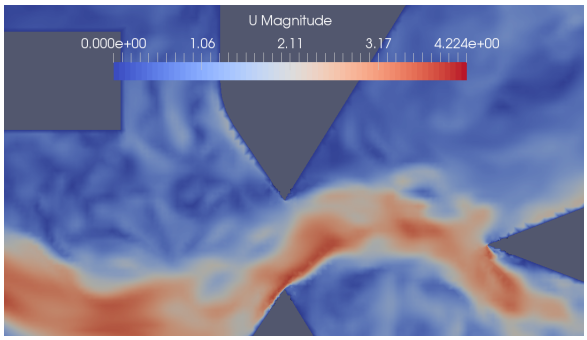

Figure 6.15: Outlet throat angle, simulation view by Candelo, C. A. [15]

6.2.3 Compressible flow studies

In addition, the project made by Candelo, C. A. [15] included a numerical analysis with compressible flow, to evaluate how does the fluidic oscillator work under different conditions. As in the aerospace industry (among other fields), it shall be used with air, it was very important to not simply test it with water, but also with this gas. The study comprehends just one simulation with a reference velocity of the flow of 0.67 m/s, and a detailed study of the effects on the device. The results of the analysis correspond to two different lines in the geometry:

- Axis of symmetry of the fluidic oscillator 6.17
- Upper outlet channel of the fluidic oscillator 6.18

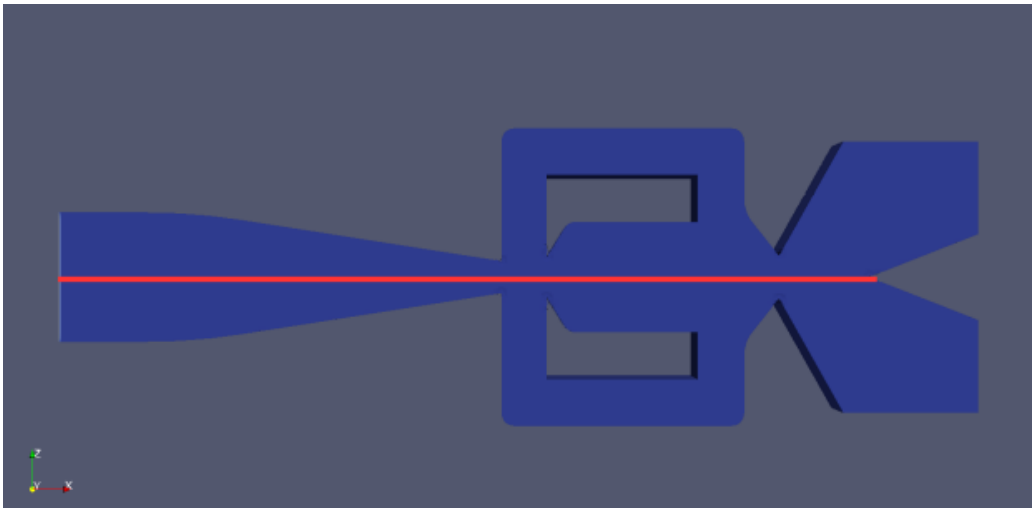

Figure 6.17: Analysis of the axis of symmetry under compressible conditions from Candelo, C. A. [15]

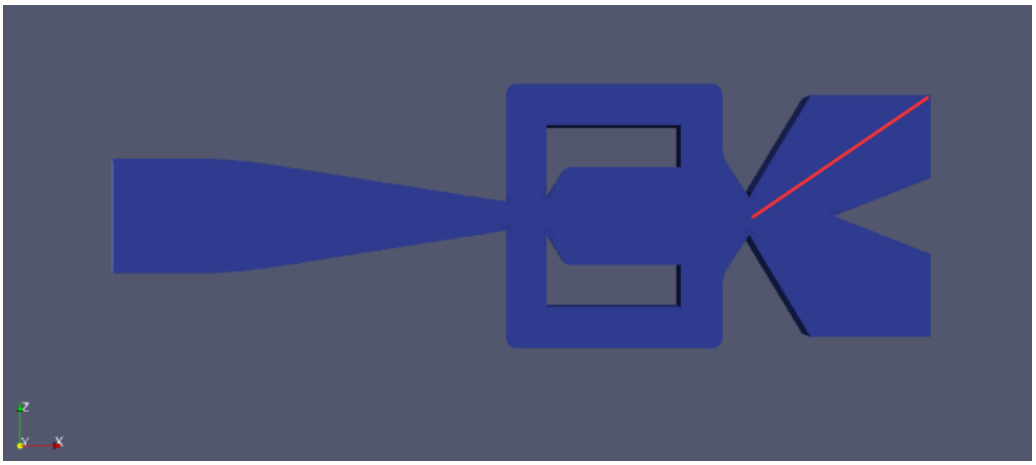

Figure 6.18: Analysis of the upper outlet channel under compressible conditions from Candelo, C. A. [15]

On the one side, according to the the axial symmetry analysis, it was concluded that the pressure distribution is completely different in both cases, compressible and incompressible. When feeded with air (compressible), the fluidic oscillator presents two clear high pressure zones, the inlet throat and the walls of the outlet throat where the fluid collides at the end of the mixing chamber. This high pressure areas cannot be distinguished in the incompressible case.

On the other side, referring to the upper outlet channel analysis, both cases presented again a completely different pressure distribution. The incompressible case experimented a sudden drop of the pressure, but it recovered as fast as it had dropped, to reach the final value that remains almost constant until the end of the outlet (figure 6.19). But in the compressible case, the pressure dropped at light strokes, maintaining at certain points a constant value (figure 6.20).

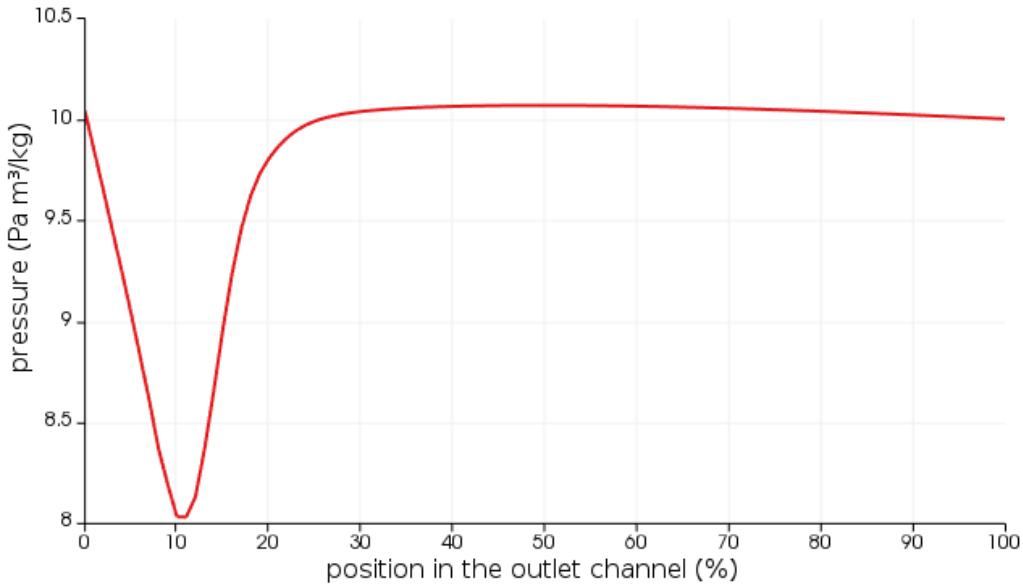

Figure 6.19: Pressure distribution in the outlet channel, incompressible flow, adapted from Candelo, C. A. [15]

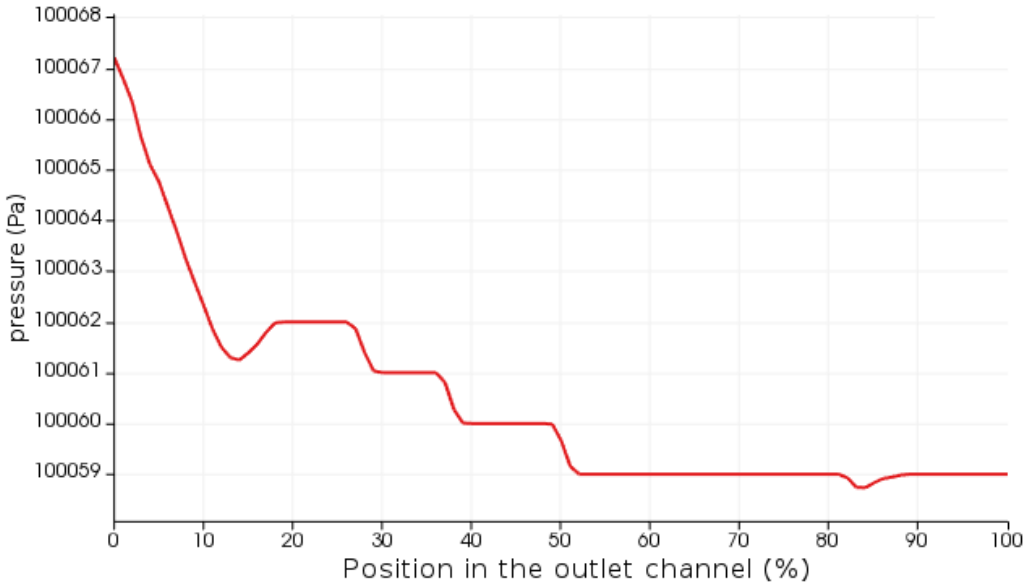

Figure 6.20: Pressure distribution in the outlet channel, compressible flow, adapted from Candelo, C. A. [15]

7 Theoretical framework

7.1 Fluid dynamics fundamentals

In the fluid dynamics field, the fluid is considered as continuum, and it has properties such as density, pressure or temperature. It means that the behaviour of individual molecules is neglected, so what matters is an average over the fluid elements. The fundamental equations that govern the behaviour of a fluid flow are made basically of differential equations which represent the interdependence between the different variables, and how they evolve in time and space. Also, they can be complemented by algebraic relations, like when working with compressible flow with the equation of state.

The governing equations for compressible flow are:

- The continuity equation 7.2
- The momentum equations (Navier-Stokes) 7.3 7.4 7.5
- The energy equation 7.8

The terms of these equations are u, ν, w, the velocity components in the x, y, z directions, the density ρ, the temperature T, the static pressure p, the dynamic viscosity μ and C_p the specific heat at constant pressure. These equations are all conservation laws. The conservation law for a quantity U states:

> The variation of the total amount of a quantity U inside a given domain is equal to the balance between the amount of that quantity entering and leaving the considered domain, plus the contributions from eventual sources generating that quantity [16].

The equation of the integral form of the previous statement can be written as:

$$\frac{\partial}{\partial t}\int_{\Omega} U \mathrm{d}\Omega + \oint_{S} \vec{F} \cdot \mathrm{d}\vec{S} = \int_{\Omega} Q \mathrm{d}\Omega \tag{7.1}$$

The continuity equation (also known as Mass Conservation Equation, or 1st conservation law) defends the empirical fact that matter is conserved in a fluid system, it cannot be created nor destroyed. That is to say:

Per unit volume, the sum of all masses flowing in and out per unit time must be equal to the change of mass due to change in density per unit time [17]

It can be applied at each point in a fluid and to all fluids, both compressible and incompressible flows, and Newtonian and non-Newtonian fluids:

$$-\frac{\partial \rho}{\partial t} = \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho \nu)}{\partial y} + \frac{\partial(\rho w)}{\partial z} \tag{7.2}$$

The momentum equation 7.3 7.4 7.5 is defined as the product of the density and the velocity, so it is a vector with three components in space that make a total of three equations. They are also called Navier-Stokes equations, and together with the continuity equations are the basis for the fluid dynamics analysis. They follow Newton's second law of motion, that can be equally applied to solids, as well as to liquids and gases, with the only difference that fluids distort without limit. Navier-Stokes equations state that:

The inertial forces acting on a fluid element are balanced by the surface and body forces [17]

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \nu\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) = -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + F_x \tag{7.3}$$

$$\rho\left(\frac{\partial \nu}{\partial t} + u\frac{\partial \nu}{\partial x} + \nu\frac{\partial \nu}{\partial y} + w\frac{\partial \nu}{\partial z}\right) = -\frac{\partial p}{\partial y} + \mu\left(\frac{\partial^2 \nu}{\partial x^2} + \frac{\partial^2 \nu}{\partial y^2} + \frac{\partial^2 \nu}{\partial z^2}\right) + F_y \tag{7.4}$$

$$\rho\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + \nu\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) = -\frac{\partial p}{\partial z} + \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) + F_z \tag{7.5}$$

It is necessary to define the sources that influence the variation of momentum, in order to understand the physical significance of each term of the equations. The sources of the variation of momentum are the forces acting on the physical system that can be external ($\vec{f_e}$) or internal ($\vec{f_i}$). The internal forces act as sources in the surface, as all the other internal forces are cancelled due to the action = reaction rule.

The terms placed in the left side of the equation correspond to the inertial terms, arising from the changes in the momentum. They give a measure of the velocity change of one fluid element through space ($\partial/\partial t$ is the local derivative and gives the variation at a fixed point,

and the remaining three terms grouped together are called convective terms or convective differential). On the other side of the equation, the countering terms are the pressure gradient ($\partial p/\partial x$), the viscous shear stress (always retarding the flow) and if there are any, body forces such as the gravitational or magnetic force, among many others. The Navier-Stokes equations can be applied too in a turbulent flow analysis, if the appropriate modifications are introduced. It has to be noticed that the convective term is non-linear. This term has great responsibility for the appearance of turbulence.

The vectorial form of the momentum conservation equations in an inertial frame of reference, with the forces acting on the body decomposed, is the following:

$$\frac{\partial}{\partial t}(\rho\vec{v}) + \nabla \cdot (\rho\vec{v}\vec{v}) = -\nabla p + \nabla \cdot (\overline{\overline{\tau}}) + \rho\vec{g} + \vec{F} \tag{7.6}$$

where ρg is the gravitational body force, $\vec{F}$ represents the external body forces and $\overline{\overline{\tau}}$ is the stress tensor, defined as follows with I as a unit tensor:

$$\overline{\overline{\tau}} = \mu\left[(\nabla\vec{v} + \nabla\vec{v}^T) - \frac{2}{3}\nabla \cdot \vec{v}I\right] \tag{7.7}$$

Finally, the Energy Conservation Equation 7.8, which follows the 1st law of thermodynamics, state that:

> The total energy within the domain remains constant, so the rate of change equals the sum of the heat addition and the work done.

The volume sources (as they have been defined in the general transport law), correspond to the sum of the work of the volume forces $\vec{f_e}$ and the heat sources other than conduction (chemically released heat, radiation, etc.). The surface sources, similar to the momentum equation, are the result of the work done on the fluid by the internal shear stresses acting on the surface of the volume [16].

$$\begin{aligned}\rho c_p\left(\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + \nu\frac{\partial T}{\partial y} + w\frac{\partial T}{\partial z}\right) = \Phi + \frac{\partial}{\partial x}\left[k\frac{\partial T}{\partial x}\right] + \frac{\partial}{\partial y}\left[k\frac{\partial T}{\partial y}\right] + \frac{\partial}{\partial z}\left[k\frac{\partial T}{\partial z}\right] \\ + \left(u\frac{\partial p}{\partial x} + \nu\frac{\partial p}{\partial y} + w\frac{\partial p}{\partial z}\right)\end{aligned} \tag{7.8}$$

The coefficient k is the thermal conductivity coefficient that comes from the Fourier's law of heat conduction:

$$\vec{F}_D = -k\vec{\nabla}T \tag{7.9}$$

where T is the absolute temperature and $\vec{F}_D$ the diffusive flux of energy. There is also the relation

$$k = \rho C_p \kappa = \mu C_p / Pr \tag{7.10}$$

$$Pr = \nu/\kappa = \mu C_p/k \tag{7.11}$$

where Pr is the Prandtl number and κ is the thermal diffusivity, empirically defined together with the dynamic viscosity μ.

Φ is the energy dissipation function that follows the equation:

$$\Phi = 2\mu \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial \nu}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2 + 0.5\left(\frac{\partial u}{\partial y} + \frac{\partial \nu}{\partial x}\right)^2 + 0.5\left(\frac{\partial \nu}{\partial z} + \frac{\partial w}{\partial y}\right)^2 + 0.5\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)^2 \right] - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial \nu}{\partial y} + \frac{\partial w}{\partial z}\right)^2 \tag{7.12}$$

If the viscosity and the specific heat are considered as constant, the resulting system of equations contain 6 unknowns, and has 5 equations, so a further equation is needed. The usual way to close the system is using a constitutive relation for the pressure, i.e. the ideal gas equation where R is the gas constant:

$$p = \rho R T \tag{7.13}$$

If some other complex phenomena is added to the problem to solve such as combustion, multiphase flows, chemical reactions, multi-species flows with condensation, evaporation, bursting or agglomeration of gas bubbles or liquid drops, or others, some further eqautions are needed to model the physical laws that describe all of those phenomena. For example, the chemical species transport equation that has tu be used if there is combustion is the following:

$$\frac{\partial}{\partial t}\left(\rho Y_k\right) + \frac{\partial}{\partial x_j}\left(\rho Y_k u_j\right) = \frac{\partial}{\partial x_j}\left(\rho D_k \frac{\partial Y_k}{\partial x_j}\right) + S_k \tag{7.14}$$

7.2 Turbulence

The different fluid flows that can be easily found in the nature, but also in the laboratory or in the industry, they are usually turbulent. During the 1880's, Reynolds, O. discovered while carrying out some visualisation studies of a flow in a pipe, that the laminar flow degenerated into a chaotic motion after reaching certain velocity. This event was linked to a non-dimensional quantity called Reynolds number (Re) after its discoverer.

$$Re = \frac{\rho U D}{\mu} \tag{7.15}$$

It represents the ratio between inertia forces and viscous forces. For a low Re, the flow runs with order in parallel streamlines. If it is increased, the flow becomes turbulent from certain Reynolds number, not necessarily very high, and the laminar structure is lost. This Re number of transition is called **critical Reynolds number ($\mathbf{Re_{Cr}}$)**, and it depends on each case conditions, geometry, boundaries, etc. Some examples of approximate critical Reynolds number values are the following:

- Flow around a sphere: $\mathrm{Re_{Cr}} = 300000$
- Flow inside a tube: $\mathrm{Re_{Cr}} = 2300$
- Flow over a flat plate: $\mathrm{Re_{Cr}} = 300000$
- Flow over an airfoil: $\mathrm{Re_{Cr}} = 500000$

A turbulent flow is characterized by its disorder, impressibility and statistical fluctuations of all the flow variables (velocity, pressure, density, temperature, etc.) around mean values. Laminar flows can be accurately computed only with the Navier-Stokes equations system and a constitutive law defining the fluid nature, without any additional information, but unfortunately, this is not the case of turbulent flow. The equations that govern a turbulent flow are non-lineal, so it can just be solved analytically in a few cases. Nowadays, with the increasing computer power in speed and memory, it is becoming easier to solve them numerically with CFD techniques. This method is called **Direct Numerical Simulation (DNS)**. But the accurate numerical description of the fluctuations of a turbulent flow is still out of reach, specially at high Reynolds numbers, because the size of the smallest turbulent eddies is inversely proportional

to $Re^{3/4}$, as the Kolmogorov scale related to the dissipation of turbulence states. For getting a resolution of n points per unit length on the smallest eddy, the mesh points required are $n^3 \times Re^{9/4}$.

But it is even more complicated than that, since the Navier-Stokes equations needs to be integrated in time for transient flows, with a time step determined by the smallest turbulent time scales, which keep a proportionality to $Re^{3/4}$. To sum up, the total calculations to be made are proportional to Re^3. This is the case for homogeneous turbulence, but if wall flows and other cases which have inhomogeneous characteristics have to be computed, the computational cost is even higher, because the mesh has to adapt its resolution near the wall.

For this reason it is essential to use some approximations that enable to run the simulations and get accurate results in acceptable CPU times. From higher to lower level of approximation, there are the following models [16]

1. Large Eddy Simulation (LES)
2. Reynolds Averaged Navier-Stokes (RANS)
3. Thin Shear Layer Approximation
4. Parabolic Approximations
5. Boundary Layer Approximation
6. Distributed Loss Model
7. Viscous-inviscid interaction models
8. Time-dependant Euler equations
9. Potential flow model

The Hierarchy of the different models and their degree of approximation is summarized in figure 7.1.

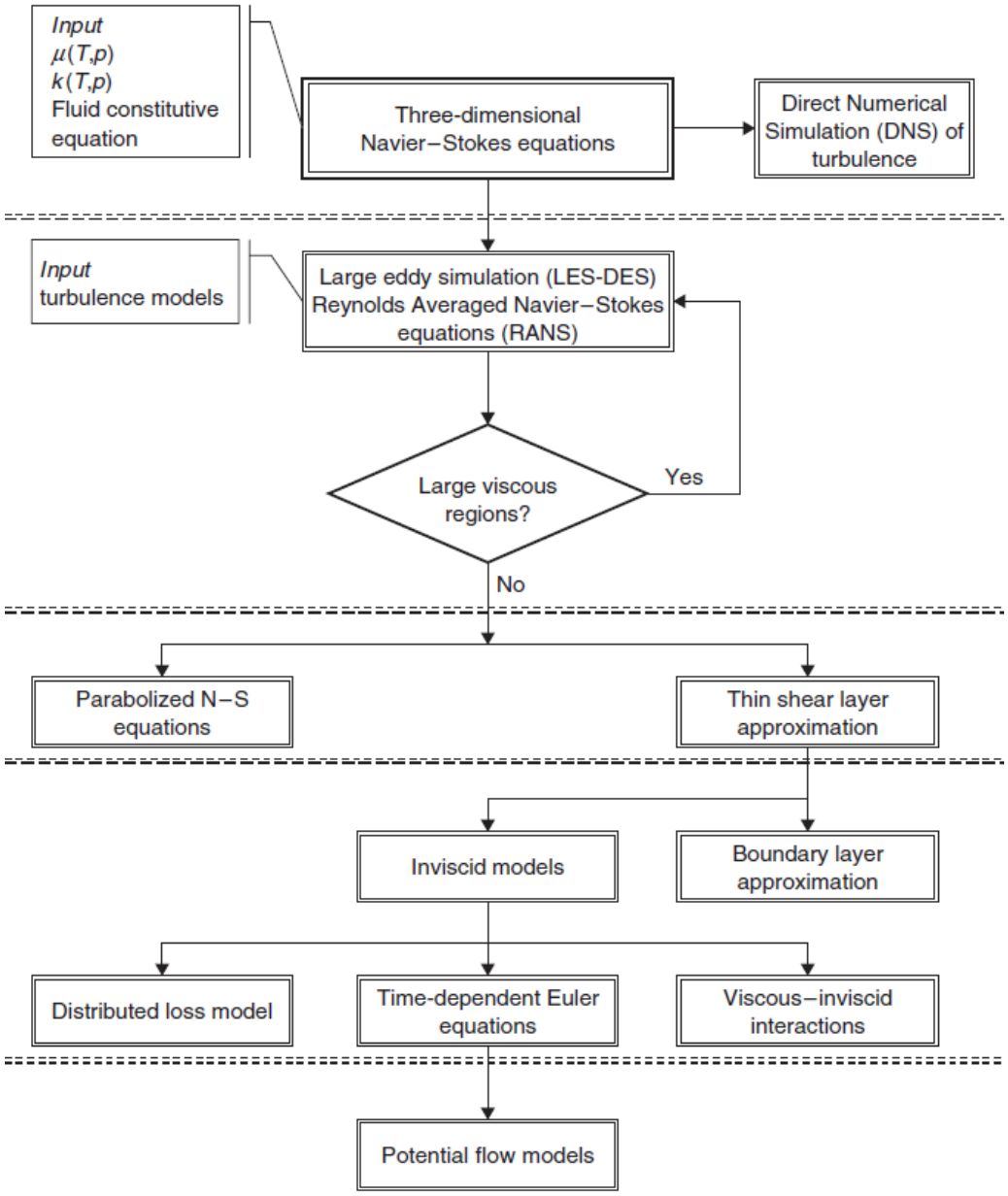

Figure 7.1: Hierarchy of the different dynamical levels of approximation from Hirsch, C. [16]

7.2.1 Boussinesq hypothesis

Most of the turbulence models are based upon the Boussinesq hypothesis:

- The turbulence decays unless there is shear in isothermal incompressible flows (this phenomena was experimentally observed)
- Turbulence increases as the mean rate of deformation increases

- The Reynolds stresses can be linked to the mean rate of deformation, or what is the same, the momentum transfer caused by turbulent eddies can be modelled with an eddy viscosity.

The last assumption states that the Reynolds stress tensor, τ_{ij}, is proportional to the trace-less mean strain rate tensor, S^*_{ij} [25]:

$$\tau_{ij} = 2\,\mu_t\,S^*_{ij} - \frac{2}{3}\rho k\delta_{ij} \tag{7.16}$$

where μ_t is the scalar property called eddy viscosity. The strain rate tensor is a physical quantity that describes the rate of change of the deformation of a material in the neighbourhood of a certain point.It is a purely kinematic concept. For any fluid, any gradual change in its deformation (i.e. a non-zero strain rate tensor) gives rise to viscous forces in its interior, due to friction between adjacent fluid elements. k is the turbulence kinetic energy, and δ_{ij} the Kronecker delta (not to be confused with Dirac delta function). Its value is 0 if $i \neq j$ and 1 if $i = j$.

7.2.2 LES

The Large Eddy Simulation is the highest approximation for numerical calculations of turbulent flow, so it requires high CPU resources. Its approach is similar to DNS. It directly computes the turbulent fluctuations in space and time above a certain value of length scale.The smaller scale eddies, called subgrid scales are modelled using semi-empirical laws. Those are called SubGrid-Scale Models (SGS), and some of the currently used are the following:

- Smagorinsky model
- Algebraic Dynamic model
- Dynamic Global-Coefficient model
- Localized Dynamic Model
- WALE (Wall-adapting Local Eddy-viscosity) model
- RNG-LES model
- Structural modelling

The computational effort requirement for LES is much lower than DNS but still too high for large Re number simulations. It is proportional to $Re^{9/4}$.

7.2.3 RANS

When it is not possible to solve directly the equations due to the high Re, turbulence models are used in order to predict the behaviour of the small scale fluctuations, so that just the bigger scale fluctuations has to be solved with the conventional formulation. This process is called Reynolds-averaged Navier-Stokes. It consists on the following fundamental idea:

magnitude = average magnitude + magnitude fluctuations

Or what is the same:

$$\phi = \phi + \phi' \tag{7.17}$$

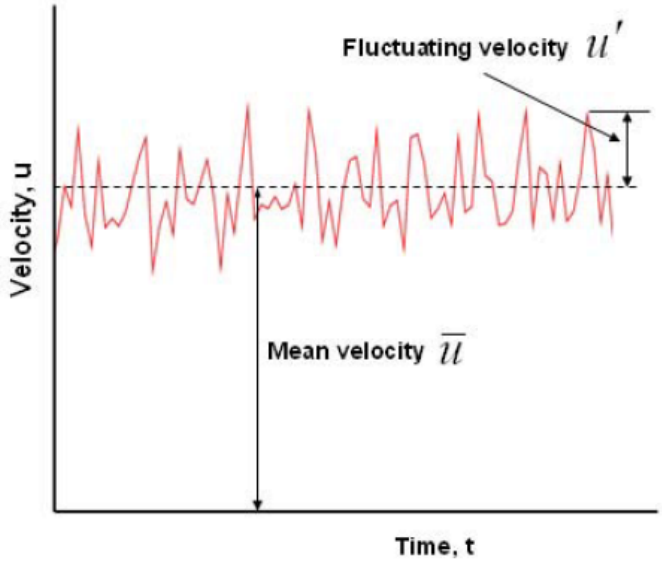

Figure 7.2: Instantaneous and average velocity in turbulent flow from Sayma, A. [17]

Rewriting the Navier-Stokes equations with those new terms, the final result of the RANS approach is:

$$\frac{\partial}{\partial t}\left(\rho\overline{u}_i\right) + \frac{\partial}{\partial x_j}\left(\rho\overline{u}_i\overline{u}_j\right) = -\frac{\partial\overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial\overline{u}_l}{\partial x_l}\right)\right] + \frac{\partial}{\partial x_j}\left(-\rho\overline{u_i'u_j'}\right) \tag{7.18}$$

This reordered equation uses the concept of eddy viscosity introduced in 1887 by Boussinesq, J. V. $-\rho\overline{u_i'u_j'} = \tau_{ij}^t$ is called Reynolds stress tensor. It represents the stresses caused by the momentum transported by the turbulent flow, and it is responsible of the energy

dissipation due to fluctuations. Its nature and the relations between the Reynolds stresses and the mean flow are still unknown, and that is why turbulence models needs to be applied when working with RANS. The main idea of the turbulence models is to be able to solve the Reynolds stresses tensor using approximations, in order to solve the whole system of equations. Some of the turbulence methods that are currently used are the following:

- $k - \varepsilon$
- $k - \omega$
- $k - \omega$ Shear Stress Transport (SST)
- Spallart-Allmaras
- Reynolds Stress Transport (RST)
- Scale Adaptative Simulation (SAS)

Prandtl introduced in 1930 the concept of turbulent viscosity, μ_t as an analogy of Newton's viscosity law:

$$\tau_{ij}^t \approx \mu_t \frac{\partial \overline{u}_i}{\partial x_j} \tag{7.19}$$

Again, the term μ_t has to be defined. There exist several turbulent models to calculate its vale. Prandtl proposed a simple method known as Prandtl's mixing-length method:

$$\mu_t \approx \rho l^2 \left| \frac{\partial \overline{u}_i}{\partial x_j} \right| \tag{7.20}$$

where l is the mixing-length. This simple model is the basis for the **Law of the wall**, an accurate model for flow fields attached to a wall with small pressure gradients.

The range of application of those three different approaches described so far (DNS, LES, RANS) is summarized in figure 7.3. The energy spectrum is dependant of the wave number k which is defined as:

$$k = 2\pi/\lambda \tag{7.21}$$

where λ is the wavelength.

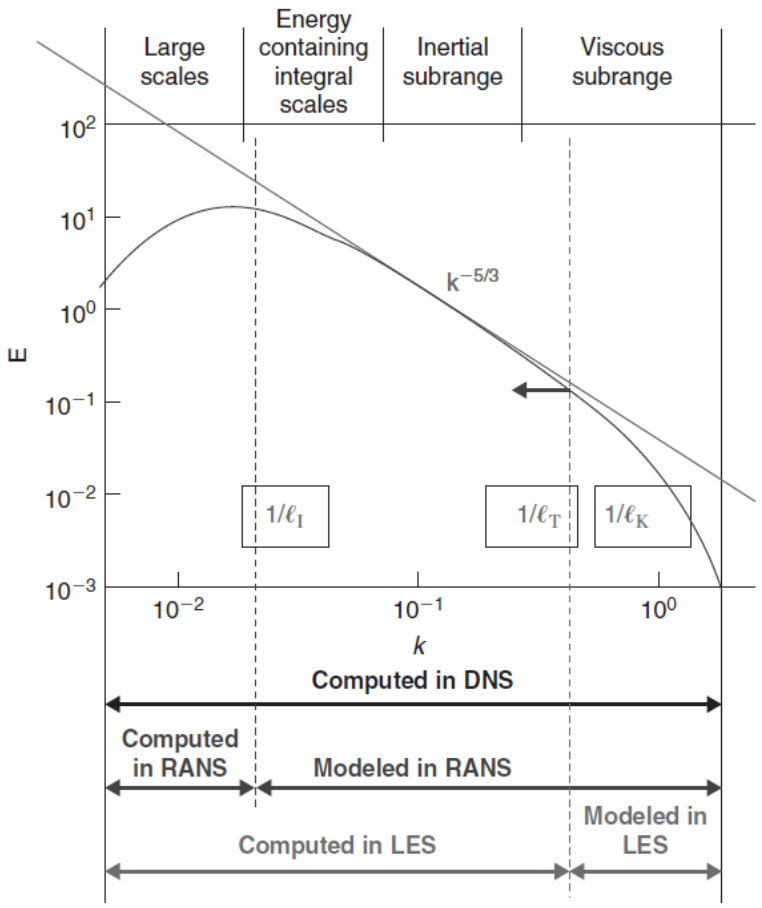

Figure 7.3: Range of application of DNS, LES, RANS models and energy spectrum of turbulence in function of k, from Hirsch, C. [16]

There are some recent studies that use hybrid combinations of LES and RANS models in order to compute wall bounded flows and attached boudnary layers. Those can be found in Haase et al. [26]

7.2.4 Thin Shear Layer approximation (TSL)

Considering an arbitrary coordinates system with ξ^1 and ξ^2 along the surface and ξ^3 pointing toward the normal, the TSL approximation of the NavierStokes equations consists in neglecting the derivatives of ξ^1 and ξ^2 that occur in the turbulent and viscous shear stress terms [16]. This approximation is very similar to the Boundary Layer approximation, even though the latter is not as accurate as the TSL. The reason is becausein both approximions the viscous terms are neglected, but in the TSL, the momentum equations in the directions normal tho the shear layer are retained and the viscous-inviscid transition is integrally calculated, instead of applying

the constant pressure rule in the normal direction of the boundary layer, as it happens in the Boundary Layer approximation.

7.2.5 Parabolic approximations

The Parabolic approximations of the Navier-Stokes equations, (PNS), can only be applied to the steady state formulation of the Navier-Stokes equations. It is used in situations where the main flow direction is predominant over the cross-flow components, that have a much lower order of magnitude. An example of application of this method would be a channel flow. Considering x the stream-wise direction, its derivatives are neglected in the shear stress terms, when comparing them to the other directions derivatives. A similar approximation is used to simplify the energy diffusion terms, as long as the x-direction, as it has been said, corresponds to the forward flow direction.

7.2.6 Boundary Layer approximation

Prandtl discovered that the viscous region extension (δ) remains constant at high Re numbers ($\delta/L \approx \sqrt{\nu/UL}$ for a body of length L). Thus, in practical terms, on an aircraft wing with a chord of 1m, for $Re = 10^6$, the boundary layer thickness will be just a few mm. That is to say, the velocity will change from zero on the wall to approximately 800 km/h in a few millimetres. This means that the gradients in the normal direction are much larger than in the stream-wise direction [16]. In those cases, as there is no separation, the pressure field can be decoupled from the viscous velocity field calculations. Assuming that the vertical velocity component can be neglected against stream-wise velocity components it can be also assumed:

$$\frac{\partial p}{\partial z} \equiv \frac{\partial p}{\partial n} \cong 0 \tag{7.22}$$

so the inner pressure of the viscous boundary layer, and the outer pressure take equal values, obtained from an inviscid computation. It represents a significant simplification of the Navier-Stokes equations.

7.2.7 Distributed Loss model

The Distributed Loss model is basically used in the fields of turbo-machinery, river hydraulics and oceanography, in internal channel flows [16]. It is assumed that the effect of the shear stresses on the motion is equivalent to a distributed friction force. Some of the three-dimensional

flow details are lost in the simulation when this model is applied, in particular, those that are deeply influenced by viscosity. The governing equations of this model can be taken as inviscid, but with an entropy source. Thus, the approximation is inviscid but non-isentropic.

7.2.8 Viscous-inviscid interaction models

Back to the case of the Boundary Layer approximation, if the influence of the boundary layers on the inviscid flow can not be neglected, the interaction must be taken into account in an iterative process, by recalculating the inviscid pressure field with the limits of the inviscid region on the edge of the boundary layer that has been obtained on the previous iteration. This concept is applied for thick boundary layers up to small separated regions.

7.2.9 Time-dependant Euler equations

If the shear stresses and the heat conduction terms are neglected in the original Navier-Stokes equations, what is left is the most general flow configuration for non-viscous and non-heat conducting fluids. This set of equations is called Euler equations. This approximation is only valid for high Re, outside viscous regions. In an absolute frame of reference, in conservation form, the equations are:

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot \vec{F} = Q \tag{7.23}$$

where U are the conservative variables, $\vec{F}$ is the flux vector with the Cartesian components f, g, h and the source term Q can be neglected:

$$U = \begin{vmatrix} \rho \\ \rho\vec{\nu} \\ \rho E \end{vmatrix} = \begin{vmatrix} \rho \\ \rho u \\ \rho \nu \\ \rho w \\ \rho E \end{vmatrix} \quad f = \begin{vmatrix} \rho u \\ \rho u^2 + p \\ \rho u \nu \\ \rho u w \\ \rho u H \end{vmatrix} \quad g = \begin{vmatrix} \rho \nu \\ \rho \nu u \\ \rho \nu^2 + p \\ \rho \nu w \\ \rho \nu H \end{vmatrix} \quad h = \begin{vmatrix} \rho w \\ \rho w u \\ \rho w \nu \\ \rho w^2 + p \\ \rho w H \end{vmatrix} \tag{7.24}$$

7.2.10 Potential flow model

The potential flow is assumed inviscid and irrotational. It is the most simplified mathematical description of a flow system. The vorticity is set to zero, so that the velocity field can be described as an scalar function $\phi(t, x, y, z)$:

$$\vec{\xi} = \vec{\nabla} \times \vec{\nu} = 0 \longrightarrow \vec{\nu} = \vec{\nabla}\phi \tag{7.25}$$

Using the potential flow approximation, the energy equation is no longer independent, but is determined by the initial boundary conditions and the function ϕ. The basic potential equation in conservation form is:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{\nabla}\phi) = 0 \tag{7.26}$$

7.3 The Coanda Effect

The Coanda effect is the phenomena in which a flow is attached to a wall or walls placed around it and remains attached even the surface curves away from the initial jet direction. This effect was first reported by Thomas Young in 1800, and many investigations have been made since then. Wall attachment has been successfully and effectively used in fluid amplifiers, combustion, sizing of particles, air conditioning and so on [19].

The phenomena as we know it nowadays, in which all contemporary studies on the field are based was later discovered and patented by Henri Marie Coanda, a Romanian aeronautical engineer. He designed and built, in 1910, the first jet-propelled aircraft, with a rotary compressor powered by a 4-cylinder piston engine. After performing the first flight which unfortunately ended in an accident, he noticed that the flames and exhausted gases from the engine "hugged" the fuselage (so the aircraft side rapidly caught fire). After 20 years of studies, the phenomenon was recognized as a new aerodynamic effect: the Coanda effect.

Nowadays, the Coanda Effect is used in many devices, but one of the most significant examples is the NOTAR® (No tail rotor) helicopter, that uses a fan with adjustable blades in the tail-boom to get a high volume of low-pressure air. This air creates a boundary layer coat using the Coanda effect when exiting through two designed slots in the tail, and provides anti-torque control for the helicopter.

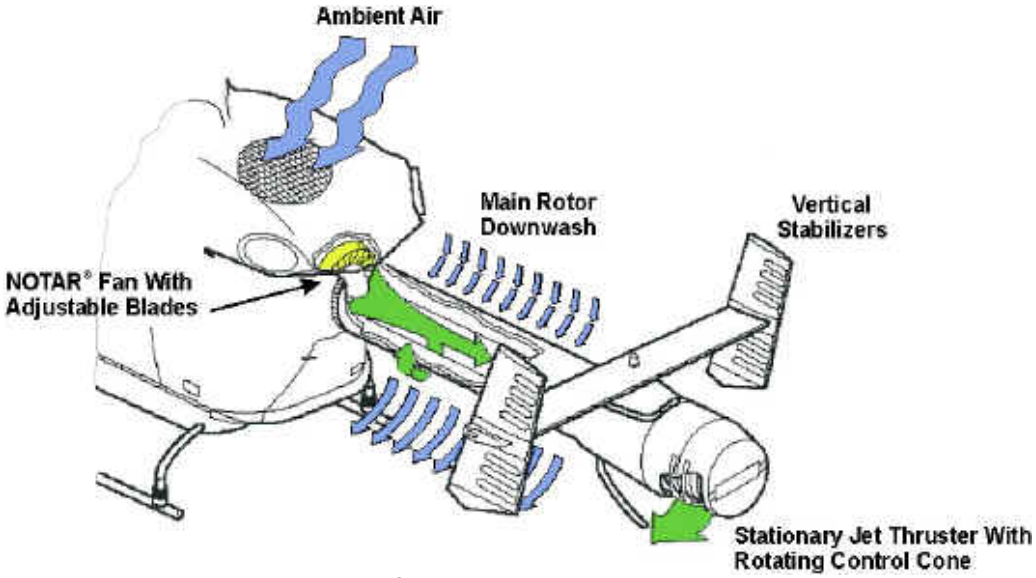

Figure 7.4: NOTAR® system sketch by MD Helicopters [18]

7.3.1 Mechanism of Coanda Effect

The mechanism of the Coanda effect can be qualitatively described with figure 7.5. In figure 7.5a), a bubbling jet that is generated vertically upwards through a single nozzle, rises through the surrounding liquid and spreads horizontally. As there is no wall or another jet next to it, the pressure around it is uniform at any vertical position. In this case, the bubbling jet ascends vertically upwards. If there is a vertical wall next to the jet (figure 7.5b), the pressure of the fluid between the wall and the jet decreases as the outer edge of the jet approaches the wall. The consequence of this pressure difference pulls the jet towards the wall so that it becomes attached. If instead of a wall, there is another vertical jet next to the first one, a very similar phenomenon occurs, as shown in figure 7.5c).

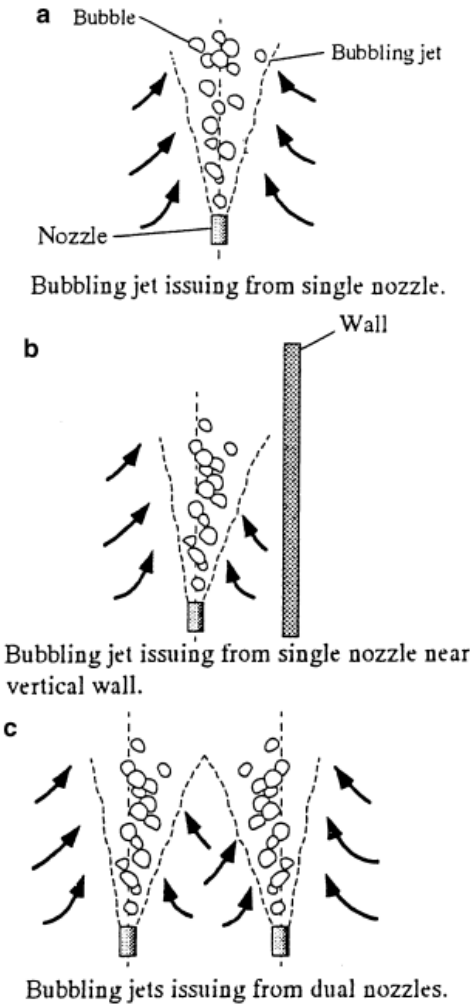

Figure 7.5: Explanation of the Coanda Effect from Iguchi, M. et al. [19]

7.4 CFD

Computational fluid dynamics (CFD) provide cost-effective means of real flows simulation, by using the numerical solution of the governing equations (also known as Navier-Stokes equations when talking about Newtonian fluid dynamics). When working in a turbulent problem, they are mostly applied in reduced forms, which are still in development, to save computational effort. Since the 1950's, the computing power and memory has increased significantly, allowing CFD techniques to arise as an great alternative way of testing fluid flow systems, validating the results with experimental and theoretical data. It can also allow to analyse systems whose conditions are not possible or extremely difficult to test empirically, and are not amenable to find any analytic solution. Nowadays, there are several CFD commercial packages in the market such as ANSYS Fluent®, OpenFOAM® , CalculiX®, AVL FIRE®, and many others,

some of them being of public domain.

7.4.1 CFD design cycle

The design cycle of a CFD analysis follows some clear steps that have to be identified:

1. Problem statement: the problem to be solved needs to bee stated, defined and understood by whoever is willing to run the analysis. Without a proper comprehension of the problem, the analysis can not be done with guarantees of success.
2. Mathematical model definition and validation: as it has been said before, there are plenty of mathematical approaches with different methods, each one of them being appropriate for certain type of calculations, and with a different level of approximation. If the problem has been understood, the optimal mathematical model can be implemented.
3. Mesh generation: a virtual 2D or 3D model of the case (depending on the approach) has to be introduced, in order to generate the mesh that will define the elements in which the mathematical model will be applied
4. Space discretization: the space discretization will be further explained in section 7.4.2.
5. Time discretization: the time discretization will be further explained in section 7.4.3.
6. Iterative solver: the iteration stability and accuracy parameters have to be established.
7. CFD software: all the previous data, mathematical models, and defined parameters, have to be implemented in the CFD software to be used. It may have a user friendly interface, or maybe the code would need to be modified directly. The most appropriate time integration and resolution methods (including convergence acceleration techniques) have to be selected.
8. Simulation run: the hardware runs the simulation under the CFD implemented parameters. It can be ran in a single computer CPU, in parallel between the different cores of a single computer or in parallel between the different computers of a cluster.
9. Graphic post-processing: Once the results have been acquitted, they have to be inspected and its format modified in order to emphasize the important parts of the analysis, and make them understandable and interpretable. A visualization software is used for this purpose.
10. Verification of the results: they have to be analysed and compared with other CFD analysis, empirical and analytical results .

It is important not to mistake **Verification** and **Validation**:

- Verification: look for errors in the model implementation. The question that should be asked while verifying is: **Are we solving the equations right?**
- Validation: check if the model is adequate.The question that should be asked while validating is: **Are we solving the right equations?**

7.4.2 Space discretization

Once the mathematical model is selected, the discretization process has to be done. This process is about translating the geometrical and mathematical models into numbers, so that the computer can understand it in order to run the simulation. The first parameter to be discretized is the space. This includes the empty spaces where the flow runs, but also the geometries and solid bodies present in the field, even those that enclose the domain. The numerical methods convert the continuous differential equations to a system that enables to find the solution at discrete points in space, called grid points. Or, what is the same, the continuous information contained in the differential equation is replaced with discrete values,

To begin with, the solid surfaces in the domain are available from a CAD system in a suitable digital form (such as the **.stl** format). Around them, a set of points is distributed in the flow domain and on the solid surfaces creating the grid, also called mesh. Instead of the continuity of the real space, there is a finite number of isolated points. The numerical approximation depends on the mesh size. The closer the points are between each other, a better discretization of the continuum flow field is reached. This is a major step in the whole process, since for complex geometries the solution will be also dependant on the form of the mesh. This step is even more significant in three-dimensional CFD simulations.

There is two different types of grids or meshes: structured and unstructured. In the structured meshes, formed by n families of lines (for a n-dimension space), the points are placed at the intersection of a line of each family. The unstructured grids have an arbitrary mesh point distribution, linked through polynomials (in 2D) or polyhedrals (in 3D). The accuracy of the obtained numerical results is critically dependent on mesh quality [16].

Once the mesh is available, the second part of the discretization process can be started. It consists on the discretization of the mathematical model equations. All mathematical

operators have to be transformed into arithmetic operations on the mesh point values [16]. The result is a set of algebraic relations between adjacent points in the mesh, called **numerical scheme**.

It is very important the choice between a steady or unsteady model for the flow equations when defining the algorithm, but in this case, there is no alternative to the use of a time-dependant mathematical model, since we are dealing with a problem with a transient flow behaviour.

There are different methods to perform the conversion from derivatives to arithmetic operations.The methods that are more widely used are:

- Finite Difference Method
- Finite Element Method
- Finite Volume Method.

7.4.2.1 The Finite Difference Method

The Finite Difference Method is the simplest method to apply. It requires high degree of mesh regularity, so it has to be set in a very structured way. It consists on approximating the derivatives of the differential equations using truncated Taylor series. According its theoretical definition, the derivative at point x is defined by:

$$u_x = \frac{\partial u}{\partial x} = \lim_{\Delta x \to 0} \frac{u\left(x + \Delta x\right) - u(x)}{\Delta x} \tag{7.27}$$

Removing the limit, the finite difference is obtained. If Δx is small but finite, the finite difference is an approximation, which will be improved by reducing Δx, but as long as it is finite, there will be an error called truncation error. The order of accuracy of the difference approximation will be related to the power of Δx with which this error trends to zero.

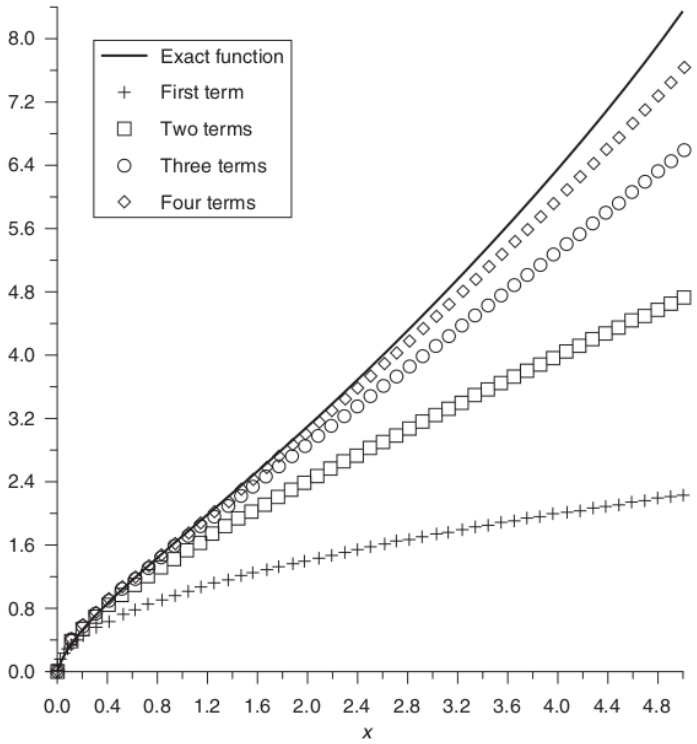

Figure 7.6: Taylor expansion of $(e^{\sqrt{x}} - 1)$ up to order 5 from Hirsch, C. [16]

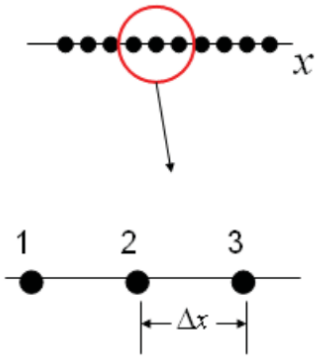

Figure 7.7: Finite difference discretization method from Sayma, A.[17]

For grid point 2 in the middle between points 1 and 3 (figure 7.7), the Taylor series expansion gives the value of the variable at point 1 as a function of the variable at point 2 and its derivatives 7.28, or the analogous expression of variable at point 3, depending of point 2 variables and its derivatives 7.29:

$$\theta_1 = \theta_2 - \Delta x \left(\frac{\partial \theta}{\partial x}\right)_2 + \frac{1}{2}(\Delta x)^2 \left(\frac{\partial^2 \theta}{\partial x^2}\right) - ..... + \frac{1}{n}(\Delta x)^n \left(\frac{\partial^n \theta}{\partial x^n}\right) - ..... \tag{7.28}$$

$$\theta_3 = \theta_2 + \Delta x \left(\frac{\partial \theta}{\partial x}\right)_2 + \frac{1}{2}(\Delta x)^2 \left(\frac{\partial^2 \theta}{\partial x^2}\right) - ..... + \frac{1}{n}(\Delta x)^n \left(\frac{\partial^n \theta}{\partial x^n}\right) - ..... \tag{7.29}$$

If the Taylor series is truncated after the second term and rearranged, the expression 7.30 for the first derivative at point 2 is obtained, and it is called first order backward difference approximation of the first derivative. The truncated terms are associated to the truncation error.

$$\left(\frac{\partial \theta}{\partial x}\right)_2 = \frac{\theta_2 - \theta_1}{2\Delta x} \tag{7.30}$$

Subtracting equation 7.28 from 7.29, a second order central difference approximation for the first derivative is obtained:

$$\left(\frac{\partial \theta}{\partial x}\right)_2 = \frac{\theta_3 - \theta_1}{2\Delta x} \tag{7.31}$$

Finally, adding 7.30 and 7.29, the second derivative at point 2 is obtained:

$$\left(\frac{\partial^2 \theta}{\partial x^2}\right)_2 = \frac{\theta_1 - 2\theta_2 + \theta 3}{(\Delta x)^2} \tag{7.32}$$

If the expression for the approximate derivatives are replaced into a differential equation, the **finite difference equation** is obtained. Hence, at point 2, the differential equation is transformed into an algebraic equation. The process must be repeated for all the points. At the end points, the known values of the boundary conditions are set, so the system can be solved.

7.4.2.2 The Finite Element Method

The Finite Element Method was developed between 1940 and 1960, in order to be applied mostly in structural dynamic problems, even though it was later extended to the fluid flow field. It is based on the "Method of Weighted Residuals", a powerful partial differential equations solver. The main advantage of this method over the Finite Difference Method is that it can be easily applied for complex geometries, using irregular grids of various shapes. Moreover, its functions give the variation of the differential equations not only at grid points, but also between them. Its mathematical formulation is the following:

Assuming the general differential equation 7.33, only for presentation purposes (the method can be applied for any other differential equation)

$$Q(T) = 0 \tag{7.33}$$

By seeking an approximate solution T' using some trial function, if we substitute this into the differential equation, it will not satisfy the equation, thus a residual appears on the right hand side:

$$Q(T') = R \tag{7.34}$$

Then, a weighting function W is introduced, so that the integral over the weighted residual vanishes over the whole domain:

$$\int_{\Omega} WQ(T')\mathrm{d}\Omega = 0 \tag{7.35}$$

It is assumed that the trial functions are polynomial with a number of unknown coefficients, so that the solution of the differential equation can be obtained after computing those coefficients by solving the system of the integrated trial functions. There are several methods which purpose is to define the trial and weighting functions. Some of them are:

- Sub-Domain Method
- Collocation Method
- Least-Square Method
- Galerkin Method

This last technique called Galerkin Method is the most widely used, and it will be further explained. First of all, local trial functions have to be assumed over the domain, discretized by subdividing it into non-overlapping cells called elements or sub-regions. They can present a variety of shapes such as triangles or quadrilaterals in two dimensions, and hexahedral, pentahedral or prisms in three dimensions. The trial functions are interpolation functions which assume the shape of the variation of the variable to compute between the grid points comprising the cell [17]. The simplest trial functions are the linear shape functions, which assume that the variable has a linear variation between grid points. The solution can be expressed as:

$$T'(x) = T_i N_i(x) \tag{7.36}$$

where T_i is the sought solution and N_i is the interpolation function, both evaluated at node i. Then, the weighting functions must be chosen. They have a value of W^i at node i and zero elsewhere for each node in the domain. The discrete weighted residual form expression is the following:

$$\int_\Omega W^i Q\left(T_i N_i(x)\right) \mathrm{d}\Omega = 0 \tag{7.37}$$

If all the domain cells are integrated, a system of algebraic equations is created:

$$K \cdot T_i = r \tag{7.38}$$

It can be solved for the coefficients T_i that represent the field function at the nodal points. K represent the Jacobean or Mass matrix, and the right hand side r usually contains boundary conditions and source terms. Another way to represent the final system of equations in matrix notation is:

$$[K]\{U\} = \{f\} \tag{7.39}$$

$[K]$ is obtained by assembling the mass matrix of each individual element $[K]_e$, and the same happens with $\{f\}$. A typical form of those parameters may be:

$$[K]_e = A \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{7.40}$$

considering A the geometric and/or physical parameters of the element, and

$$\{f\}_e = \begin{bmatrix} q_i \\ q_j \end{bmatrix} \tag{7.41}$$

7.4.2.3 The Finite Volume Method

The Finite Volume Method was developed in the 1970's, and can be considered as a variation of the Weighted Residual Method used in the Finite Element Method of discretization, where the weighting function is 1. It is currently the most used method in CFD, and can be applied in both structured and unstructured grids. It consists on discretizing directly in the physical space the integral formulation of the conservation laws, instead of the differential form, as

the FDM does. Once the grid is generated, the FVM associates a local finite volume called control volume to each mesh point, and the integral conservation laws are applied to those local volumes, obtaining cell-averaged values. This is the main difference between this method and the FEM method, where the discretized space is considered as a set of points, instead of a set of small cells, associated to a mesh point.

Moreover, the FVM has the great advantage that the conservative discretization is automatically satisfied, through the direct discretization of the integral form of the conservation laws (equation 7.1) [16]. The main property that this formulation has is that the time variation of U inside the volume only depends on the surface values of the fluxes. If an arbitrary subdivision of the volume Ω is made as shown in figure 7.8, the global conservation law can be recovered by adding up the local conservation laws from each subvolume Ω_1, Ω_2, Ω_3:

$$\frac{\partial}{\partial t}\int_{\Omega_1} U \mathrm{d}\Omega + \oint_{ABCA} \vec{F}\cdot \mathrm{d}\vec{S} = \int_{\Omega_1} Q \mathrm{d}\Omega \tag{7.42}$$

$$\frac{\partial}{\partial t}\int_{\Omega_2} U \mathrm{d}\Omega + \oint_{DEBD} \vec{F}\cdot \mathrm{d}\vec{S} = \int_{\Omega_2} Q \mathrm{d}\Omega \tag{7.43}$$

$$\frac{\partial}{\partial t}\int_{\Omega_3} U \mathrm{d}\Omega + \oint_{AEDA} \vec{F}\cdot \mathrm{d}\vec{S} = \int_{\Omega_3} Q \mathrm{d}\Omega \tag{7.44}$$

It has to be taken into account that the contributions of internal lines ABD and DE always appear twice, but they cancel each other:

$$\oint_{ED} \vec{F}\cdot \mathrm{d}\vec{S} = -\oint_{DE} \vec{F}\cdot \mathrm{d}\vec{S} \tag{7.45}$$

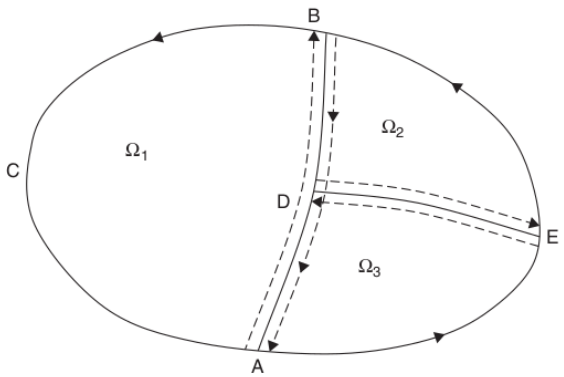

Figure 7.8: Subdivisions Ω_1, Ω_2, Ω_3, of the volume Ω from Hirsch, C.[16]

If this property is satisfied, the scheme is said to be conservative. Otherwise, if after summation of the discretized equations, the resulting equation contains contributions of the flux from inside the global cell, the discretization is said to be non-conservative. Those contributions appear as numerical internal volume sources.

The Finite Volume Method, due to its generality, it has some different approaches regarding the relation between the control volumes and the grid:

- Cell-centered approach: the unknowns are at the center of the mesh cells. The finite volumes and surfaces are defined by the grid lines.
- Cell-vertex approach: the unknowns are defined at the corners of the mesh. The variable are attached to the mesh points, i.e. to the cell vertices.

Whichever the approach of the Finite Volume Method discretization is, the conservation law (equation 7.1) has to be replaced by its discrete form:

$$\frac{\partial}{\partial t} + \sum_{faces} \vec{F} \cdot \Delta\vec{S} = Q_J \Omega_J \tag{7.46}$$

7.4.3 Time discretization

The temporal discretization is done through integration over time on the general discretized equation, so this step comes right after the spatial discretization. It is required in all of the transport equations:

$$\frac{\partial U_{(t)}}{\partial t} = f(t, U_{(t)}) \tag{7.47}$$

If the variable U depends on time ($U_{(t)}$), the case consists on an unsteady state problem, where the value of the variable U in the initial time T_0 is always going to be known:

$$U_{(t_0)} = U^0 \tag{7.48}$$

The integral form of the time discretization is the following:

$$\int_{t_n}^{t_{n+1}} \frac{\partial U_{(t)}}{\partial t} \mathrm{d}t = U^{n+1} - U^n \tag{7.49}$$

where $t_{n+1} = t_n + \Delta t$. Δt corresponds to the **timestep**, an important parameter that always has to be taken into account when setting the numerical scheme specifications. Integrating equation 7.49, the following expression is obtained:

$$\frac{U^{n+1} - U^n}{\Delta t} = \beta \cdot f(t_{n+1}, U^{n+1}) + (1 - \beta) \cdot f(t_n, U^n) \tag{7.50}$$

β correspond to a weighting factor, which value is determined by the approach of the scheme. Some of the most usual approximations are:

- Explicit discretization scheme
- Implicit discretizationscheme
- Hybrid discretization schemes

The balance between implicit and explicit methods has to be made taking into account the product

(cost per time step) * (number of time steps)

Implicit methods require lower number of time steps, but are more costly for each time step. It has to be evaluated too the stability of the time integration.

7.4.3.1 Explicit discretization scheme

This scheme, also called Euler explicit, is a first order method where the variable of the new time step only depends on the old time step variable, so $\beta = 0$.

$$U^{n+1} = U^n + f(t_n, U^n)\Delta t \tag{7.51}$$

It is the most simple method. Its main disadvantage is that it is not so stable. At best, they are conditionally stable, and have a limitation on the maximum allowable $(\Delta t)_{stab}$. It is usually quite small, because its maximum value is limited by the Courant, Friedrichs, Lewy condition (CFL-condition). This law expresses that "the distance covered during the time interval Δt, by the disturbances propagating with speed a, should be lower than the minimum distance between two mesh points." [16]

Despite the low CPU cost per iteration (no matrices have to be involved in the process), the low stability limit usully requires a large number of time steps.

7.4.3.2 Implicit discretization scheme

In this first order scheme, also known as Backward Euler, the dependant variable of the new time step is unknown and must be found with an iterative process. It takes a lot more time for calculation than the explicit scheme, but it is unconditionally numerically stable. The weighting factor value of this method is $\beta = 1$.

$$U^{n+1} = U^n + f(t_{n+1}, U^{n+1})\Delta t \tag{7.52}$$

It can handle larger time steps than the explicit method even though it will affect the accuracy of the results as well as negatively defining time resolution.

7.4.3.3 Hybrid discretization schemes

A middle point between the explicit and the implicit schemes can be reached by applying hybrid methods, where the weighting factor β take intermediate values: $\beta \in [0, 1]$, so either more reliability on the stability is ensured, as well as reducing computational effort.

$$U^{n+1} = U^n + \beta f(t_{n+1}, U^{n+1})\Delta t + [1 - \beta] f(t_n, U^n)\Delta t \tag{7.53}$$

The Crank-Nicolson discretization scheme (or trapezium method) is a second order scheme that considers the value of $\beta = 0.5$

Other time integration methods of different orders can be reached by varying the values of the parameters θ, ξ and ϕ in a generalized form of equation 7.50 containing three time levels:

$$(1 + \xi)U^{n+1} - (1 + 2\xi)U^n + \xi U^{n-1} = \Delta t \left[\theta H^{n+1} + (1 - \theta + \phi)H^n - \phi H^{n-1}\right] \tag{7.54}$$

where $H(U) = \frac{dU}{dt}$, and the parameters functions are:

- θ controls the implicitness of the method, equivalent to the previous β. $\theta = 0$ generates explicit methods.
- ξ controls the order of the finite difference formula for dU/dt
- ϕ controls the number of time levels of the space-discretized terms.

7.4.4 Iterative Methods

The resultant algebraic system of any discretization process can be solved using two large families of methods: the direct and the iterative methods. While direct methods are based on a finite number of arithmetic operations that lead to the exact solution of the system, iterative methods are based on a succession of approximate solutions, which theoretically lead to the exact solution after an infinite number of steps. Instead, a small number of operations on the matrix elements of the algebraic system is performed, approaching the exact solution within a preset level of accuracy[16]. That is to say, convergence is achieved once the difference of the solution between two consecutive iterations reaches a preset tolerance and does not increase with more iterations.

Although iterative methods are generally thought of as being designed to nonlinear systems, they can be also used for linear systems or nonlinear linearised systems.

A large number of methods with various rates of convergence and levels of complexity can be found in the literature, as it has been extensively treated so far and it is still a subject of research (in order to improve the algorithms).

7.4.4.1 Basic iterative methods

The basic iterative methods were the first methods developed for iterative solvers, and they are closely related to simple time marching, working as the approach to steady-state solutions using finite time steps.

The Jacobi method is the simplest one. Considering the matrix equation

$$[A]\{U\} = B \tag{7.55}$$

it first assume that an initial estimated solution U_i^n is known (the superscript n indicates the time level). Then, the iterative procedure is set up as follows:

$$U_i^{(n+1)} = \frac{B_i - \left(\sum_{\substack{j \\ j \neq i}}^{n} A_{ij} U_j^n \right)}{A_{ii}} \tag{7.56}$$

The two main features of the Jacobi methods are the following: there is no need to store the entire matrix as the right hand side vector $A_{ij}U_j^n$ is assembled directly during the calculations, and two solution vectors need to be stored, because the solution at the current iteration level depends on the solution in the previous iteration.

Although it is a simple method, it is usually expensive in computational terms, due to the large number of iterations required to reach convergence.

The Gauss-Seidel method is a direct improvement of the Jacobi method. The only difference between both methods is that in the Gauss-Seidel method, the values of the solution vector U_i^{n+1} are used on the right hand side of equation 7.56 as soon as they become available, so the solution becomes:

$$U_i^{(n+1)} = \frac{B_i - \left(\sum_{j=1}^{i-1} A_{ij} U_j^n \right) - \left(\sum_{j=i+1}^{n} A_{ij} U_j^n \right)}{A_{ii}} \tag{7.57}$$

A scheme of both Jacobi and Gauss-Seidel methods are presented in the following figure:

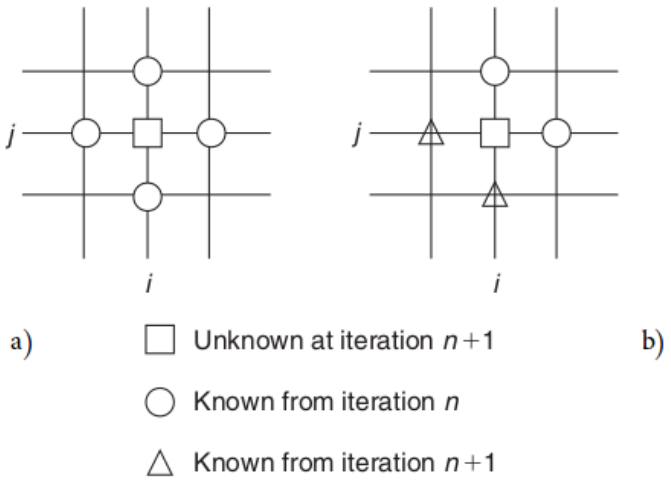

Figure 7.9: a) Jacobi method scheme, b) Gauss-Seidel method scheme. From Hirsch, C. [16]

With the Gauss-Seidel method, a higher convergence rate can be obtained since the influence of a perturbation on U^n is rapidly transmitted. On the other hand, with the Jacobi method, a perturbation of $U^{n+1}_{i-1,j}$ will be felt on $U_{i,j}$ after the whole mesh is swept. Moreover, the new value can be stored in the same location than the old value by overwriting it, as it will not be needed anymore. Therefore only one solution vector has to be stored.

Both Jacobi and Gauss-Seidel methods, even though the second one is much faster than the first one, will converge if the matrix A is diagonally dominant.

7.4.4.2 Overrelaxation methods

Overrelaxation methods are used to increase the convergence rate of an iterative methods. They consist on the propagation of the corrections ($\Delta U^n = U^{n+1} - U^n$) faster through the grid. If $\overline{U^{n+1}}$ is the value obtained from the basic iterative scheme, the value introduced at the next iterative level U^{n+1} can be:

$$U^{n+1} = \omega\overline{U^{n+1}} + (1-\omega)U^n \tag{7.58}$$

or, alternatively,

$$\Delta U^n = U^{n+1} - U^n \Delta U^n = \omega\overline{\Delta U^n} \tag{7.59}$$

with

$$\Delta U^n = \omega\overline{\Delta U^n} \tag{7.60}$$

where ω is the **overrelaxation coefficient**. A considerable gain can be achieved if the overrelaxation coefficient is appropriately optimized for maximum convergence rate.

7.4.4.3 Multigrid methods

Multigrid acceleration methods are the most efficient and general iterative techniques known so far. They can be applied to linear and non-linear systems. They work with a sequence of grids, where each grid has approximately twice the number of grid points from the previous grid, so that the final grid is the finest, and the solution is sought there. For example, considering a number of grids from 1 to M where M is the finest, the solution of the system is:

$$A^M U^M = b^M \tag{7.61}$$

The approximation to the solution U^M required by the iterative method is provided by the solution of the previous coarser grid U^{M-1}, and so on. In other words, the solution of any grid U^M is considered a good approximation of the solution on the next finer grid U^{M+1}, so that it will reduce the convergence time and computational effort.

7.4.5 OpenFOAM®

OpenFOAM® (Field Operation And Manipulation) [27] is an open source software for computational fluid dynamics (CFD). It was created in 1989 by Henry Weller under the name "FOAM", although it was released open source much later (2004) with the current name, under the conditions of GNU General Public Licence, as it is right now after many years. It consists on a C++ toolbox for the development of customized numerical solvers and pre-/post-processing utilities.

Regarding the pre-processing utilities, OpenFOAM® offers a variety of features related to the meshing process such as `snappyHexMesh`, for the mesh generation for complex geometries, or `blockMesh` for simple geometries, apart from various mesh conversion and manipulation tools.

OpenFOAM® has an extensive list of solvers in order to simulate all kinds of flows, each one of them designed for a specific type of problem. Moreover, there are several models related to various physical properties of the flow. In the first place, there are the thermophysical models, that are concerned with energy, heat and physical properties, so they must be used when the thermal energy, the compressibility or the mass transfer is an important factor. There are also different transport models, comprising the Newtonian, the Bird-Carreau or the Cross Power Law models among many others. And finally, two different methods can be used for the turbulence modelling (if the flow has not been defined as "laminar"):

- Reynolds-Averaged Simulation (RAS)
- Large-Eddy Simulation (LES) and Detached-Eddy Simulation (DES, DDES, etc)

Both of them can be applied in either incompressible and compressible flows, and include a variety of different models. The RAS models for compressible flows are the following:

- LRR: Launder, Reece and Rodi Reynolds-stress turbulence model for incompressible and compressible flows.
- LaunderSharmaKE: Launder and Sharma low-Reynolds k-epsilon turbulence model for incompressible and compressible and combusting flows including rapid distortion theory (RDT) based compression term.
- RNGkEpsilon: Renormalization group k-epsilon turbulence model for incompressible and compressible flows.
- SSG: Speziale, Sarkar and Gatski Reynolds-stress turbulence model for incompressible and compressible flows.
- SpalartAllmaras: Spalart-Allmaras one-eqn mixing-length model for incompressible and compressible external flows.
- buoyantKEpsilon: Additional buoyancy generation/dissipation term applied to the k and epsilon equations of the standard k-epsilon model.
- kEpsilon: Standard k-epsilon turbulence model for incompressible and compressible flows including rapid distortion theory (RDT) based compression term.
- kOmega: Standard high Reynolds-number k-omega turbulence model for incompressible and compressible flows.
- kOmegaSSTSAS: Scale-adaptive URAS model based on the k-omega-SST RAS model.
- realizableKE: Realizable k-epsilon turbulence model for incompressible and compressible flows.
- v2f: Lien and Kalitzin's v2-f turbulence model for incompressible and compressible flows, with a limit imposed on the turbulent viscosity given by Davidson et al.

The available LES and DES models for compressible flows are the following:

- DeardorffDiffStress: Differential SGS Stress Equation Model for incompressible and compressible flows
- Smagorinsky: The Smagorinsky SGS model.
- SpalartAllmarasDDES: SpalartAllmaras DDES turbulence model for incompressible and compressible flows

- SpalartAllmarasDES: SpalartAllmarasDES DES turbulence model for incompressible and compressible flows
- SpalartAllmarasIDDES: SpalartAllmaras IDDES turbulence model for incompressible and compressible flows
- WALE: The Wall-adapting local eddy-viscosity (WALE) SGS model.
- dynamicKEqn: Dynamic one equation eddy-viscosity model
- dynamicLagrangian: Dynamic SGS model with Lagrangian averaging
- kEqn: One equation eddy-viscosity model
- kOmegaSSTDES: Implementation of the k-omega-SST-DES turbulence model for incompressible and compressible flows.

Moreover, OpenFOAM® also offers a wide range of wall function models can be applied as boundary conditions on individual patches, so different wall functions can be applied to different wall regions.

The turbulence methods used in the simulations of the fluidic oscillator will be later discussed on the Simulation parameters section 10.1.

OpenFOAM® is also supplied with different post-processing utilities. On the one hand, it has a GUI (graphical user interface) called paraView used by the `paraFoam` utility. It also has a CLI (command line interface) which enables the user to have a conventional data processing activity after the simulation has run (post-processing), or a run-time processing, where the data processing is performed during the running of a simulation. Moreover, it has a set of functions for data sampling and monitoring, and it is able to convert the data into other formats for third-party post-processing tools.

8 Problem approach

This study will focus its attention on the **double feedback loop bi-stable fluidic oscillators**.

8.1 Oscillator characteristics

Regarding their geometry, they can present a wide range of variable dimensions, depending on various factors such has the flow regime (laminar or turbulent), the purpose of the actuator and the use that is given to it, but also its integration with the full system. Its different parts are:

a) **Inlet nozzle:** It consists on the longest channel of the device, due to the necessity of the incoming flow to be as longitudinal as possible, to not to alter the mixing chamber flow. Moreover, it determines the mixing chamber inlet flow conditions.

b) **Mixing chamber:** Where the crucial phenomena that determines the outlet flow and the oscillator characteristics take place. The working principle of the fluidic oscillator is based on the formation of zones of different pressure on both sides of the jet inside the mixing chamber that control the direction of the flow.

c) **Inner blocks:** The end of the inner blocks is a sharp edge, which angle and amplitude of the throat between both of them have a deep impact on the turbulence inside the mixing chamber.

d) **Feedback channels:** Connect the inlet and outlet of the mixing chamber. When the flow collides with the outlet throat walls of the mixing chamber, part of the flow is redirected upstream through the feedback channels, and influences the incoming flow from the inlet channel.

e) **Outlet:** The flow that passes through the outlet throat reaches the outlet, where a splitter plate divides it in two channels. If it was in laminar regime, the flow would be divided and each half would go throughout both channels, but as the oscillator works under turbulent conditions, an oscillating flow between both channels is created. After the bifurcation, the channels get thinner and the flow speed is increased to adapt it to the outlet conditions.

f) **Recirculation bubble:** Created with the interaction of the main flow and perpendicular flow coming from the control port (in the case of study, the feedback channel. As

more fluid is injected from the feedback channel, the bubble size increases and extends downstream while unattaching the flow from the wall of the mixing chamber.

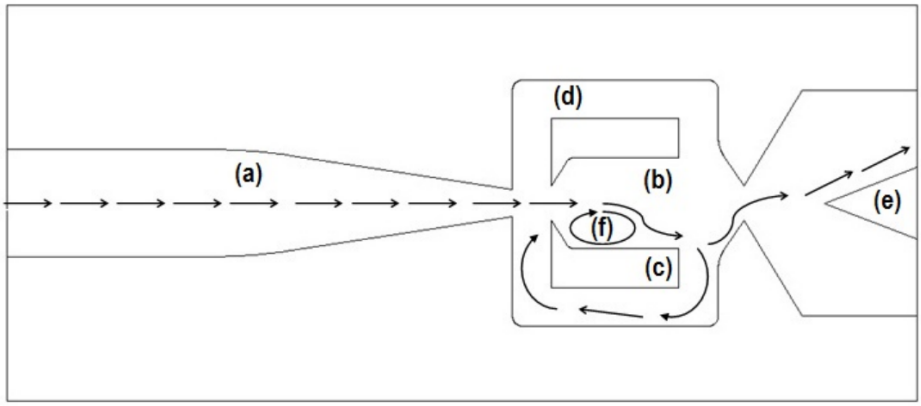

Figure 8.1: Bi-stable fluidic oscillator flow regions scheme from Ruiz, M. [2]

8.1.1 Stages of the oscillator

At the beginning, the flow is randomly attached to the upper or the lower wall of the mixing chamber, next to the inner block. It has been considered, for this particular explanation, that the process begins in the lower wall:

In the first stage, $\phi = 0°$ (figure 8.2), the jet is attached to the lower wall of the mixing chamber. It partially collides with the outlet throat lower wedge before exiting through the upper outlet channel, generating some vorticity in the outlet, and a recirculating flow in the lower feedback channel, that starts to pump the recirculating bubble between the main flow and the lower wall.

In the stage where $\phi = 60°$ (figure 8.3), the jet recirculating bubble begins to grow, fed by the perpendicular flow coming from the lower feedback channel. It unattaches the main flow from the inner block, and pushes it to the upper side of the oscillator, while colliding entirely against the outlet throat, that redirects it to the upper outlet channel.

Then, in $\phi = 120°$ (figure 8.4), the lower recirculating bubble, still being fed by the perpendicular flow, has grown enough to attach the main flow to the upper side. The jet passes through the throat with almost any losses, and exits through the upper outlet channel, although it is about to be split by the plate.

Finally, the phase $\phi = 180°$ (figure 8.5) is the symmetric stage of $\phi = 0°$. The jet is attached to the upper wall of the mixing chamber because the lower recirculating bubble has reached its maximum. The upper recirculation flow begins to form, creating the consequent recirculating bubble. The main flow starts to collide with the upper part of the outlet throat and exits the device through the lower outlet channel, even though there is some vorticity in front of the splitter plate in the outlet.

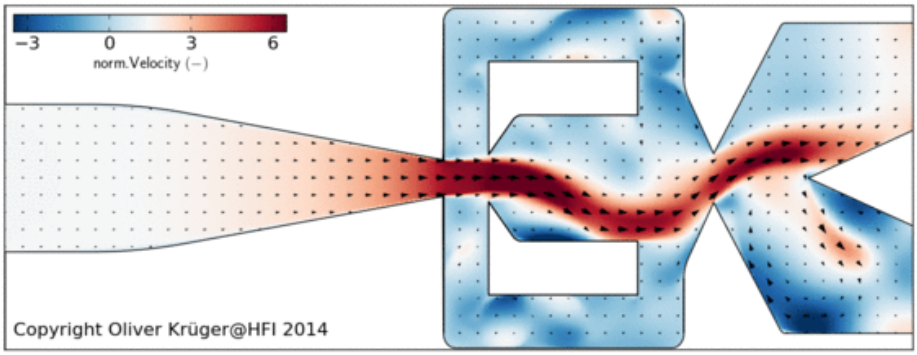

Figure 8.2: CFD simulation results of the flow in a fluidic oscillator, phase $\phi = 0°$ from Krüger, O.[20]

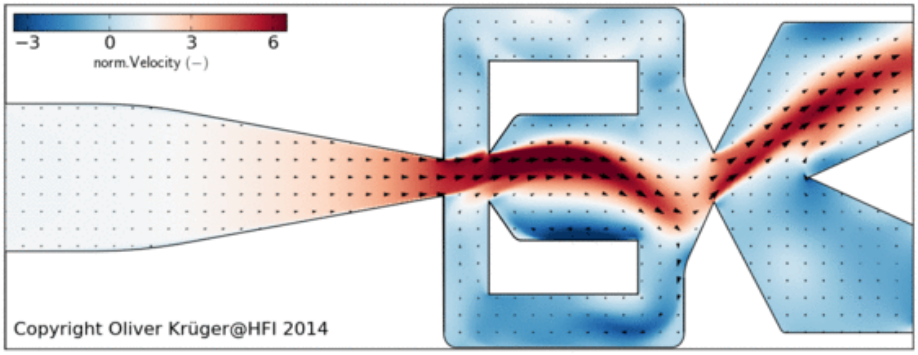

Figure 8.3: CFD simulation results of the flow in a fluidic oscillator, phase $\phi = 90°$ from Krüger, O. [20]

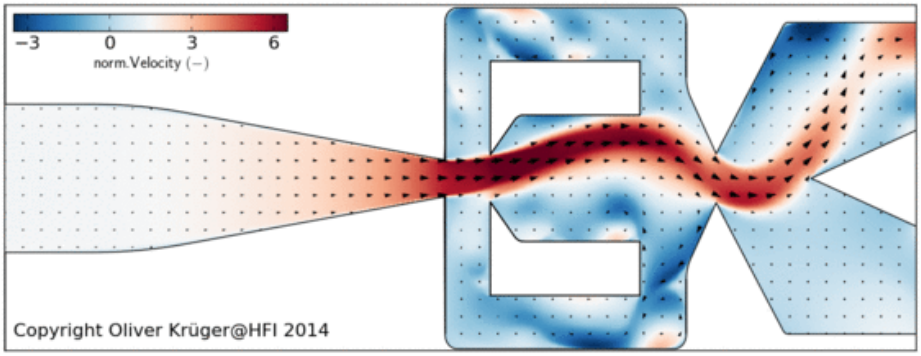

Figure 8.4: CFD simulation results of the flow in a fluidic oscillator, phase $\phi = 180°$ from Krüger, O. [20]

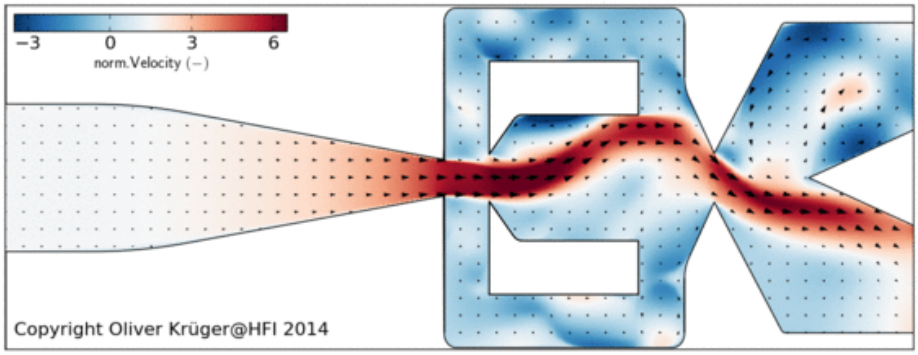

Figure 8.5: CFD simulation results of the flow in a fluidic oscillator, phase $\phi = 270°$ from Krüger, O. [20]

As any oscillatory fluid-flow phenomena with fixed-geometry boundaries, fluidic oscillators are governed by the Strouhal number, defined as:

$$St = \frac{fb}{w} \tag{8.1}$$

where:

- f (Hz): frequency of the oscillations or vortex shedding
- b (m): width of the inlet nozzle throat
- w (m/s): bulk velocity in the exit of the inlet nozzle

For large Strouhal numbers (order of 1) the viscosity dominates the fluid flow resulting in a collective oscillating movement of the fluid, whereas for low Strouhal numbers ($St << 1$) the flow can be considered as quasi-steady and high speeds dominate the flow oscillations. At intermediate Strouhal numbers, the oscillations are characterized by quick build up and subsequent shedding of vortices.

The results will be presented as the variation of the Strouhal number with the Reynolds number.

8.2 Turbulence model

The turbulent model used by the solver in this study correspond to one of those that have been more accurate in previous studies related to numerical analysis of fluidic oscillators [1] [28] [2] (among others). In this case, the fluidic oscillator will be analysed according the $k - \omega$ SST approach.

The $k-\omega$ Shear Stress Transport turbulence model is a two-equations eddy-viscosity model developed by Menter in 1993 [29], which combines the $k-\omega$ and the $k-\varepsilon$ models. It presents a gradual transition from the standard $k-\omega$ model in the inner region of the boundary layer, to a version of the $k-\varepsilon$ model applied to high Reynolds number in the outer part of the boundary layer. Whereas the $k-\omega$ model links turbulence kinetic energy k with the specific dissipation ω, the $k-\varepsilon$ model links turbulence kinetic energy k with the dissipation velocity ε. Both of them provide the solver with two extra equations that give the needed parameters to compute the dynamic viscosity, necessary for the Navier-Stokes equations. It contains a modified formulation for the turbulent viscosity, focused on the transport effects of the turbulent shear stress:

$$\nu_T = \frac{a_1 k}{max(a_1\omega, SF_2)} \tag{8.2}$$

instead of:

$$\nu_T = k/\omega \tag{8.3}$$

as in the Wilcox's standard $k-\omega$ model.

Further explanation of the turbulent model formulation can be found in [29].

8.3 Experiment conditions

A standard ambient temperature and pressure (SATP) environment is set for the simulations, with a temperature of 25 °C or 298.15 K and an absolute pressure value of 100kPa (1 bar).

The inlet velocity parameter is crucial in this study. As the compressible effects on the flow are intended to be studied, the velocity value has to be high enough, but in the other hand, if it is too high, the flow becomes trans-sonic, so the results and the implemented parameters would no longer be valid. The base inlet velocity is set to $M = 0.5$. Following the next equation, the resultant value in m/s is 173.034 m/s.

$$U = M\Delta c = M\sqrt{\frac{\gamma RT}{M}} \tag{8.4}$$

where $\gamma = 1.4$ for air, $R = 8.314510 \frac{J}{mol \Delta K}$ (universal gas constant) and the molecular mass of the gas $M = 0.028964 kg/mol$.

Further simulations with a variation on the inlet velocity value will be carried on, either to seek a numerically stable solution of the computations in the event of divergence, and to expand the knowledge of the turbulence phenomena inside the fluidic oscillator. The range of tested inlet velocities will be always high enough to feel the compressible effects, but low enough to stay outside the trans-sonic zone.

9 Geometry

9.1 Geometry 3D modelling

The 3D model of the oscillator that has been used in this project for the simulation is the same used in previous projects carried out by the Aeronautics department of UPC ETSEIAT. It is shown in the following figures:

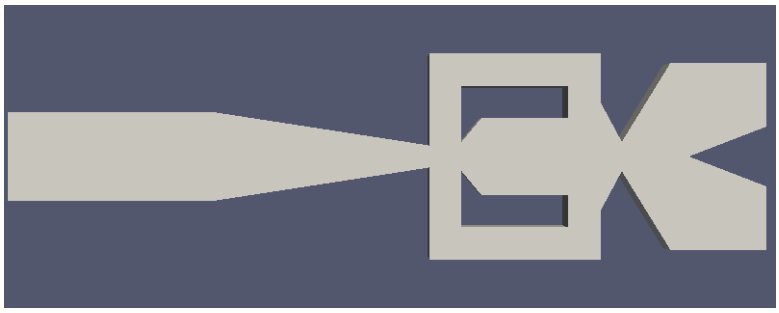

Figure 9.1: Geometry of the fluidic oscillator

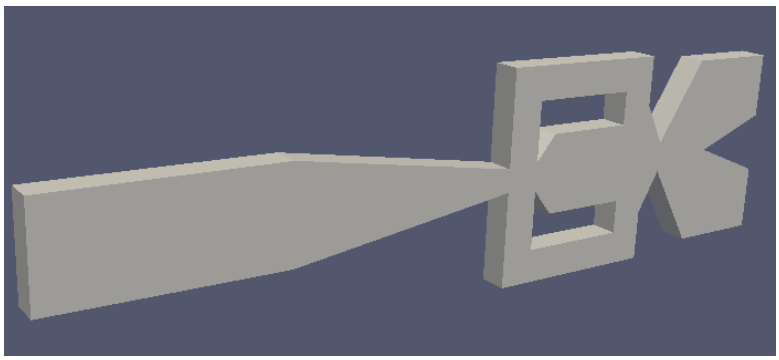

Figure 9.2: Geometry of the fluidic oscillator

It has the following dimensions: 0.075 m (x-direction), 0.02165 m (y-direction) and 3.25×10^{-3} m (z-direction).

9.2 Geometry meshing

9.2.1 Meshing progress

At first, the `snappyHexMesh` utility of OpenFOAM® was used. It generates a 3D mesh from triangulated surface geometries (tri-surfaces) in .stl (stereolithography) or .obj (object) format.

The mesh contains hexahedra and split-hexahedra elements, which are iteratively refined from the starting mesh and morphing the resulting mesh to the surface.

The starting mesh or background mesh is created with the `blockMesh` utility. The basic usage of this utility allows the creation of a hexahedric (hex) mesh, which contains the surfaces of the geometry. It defines the base level mesh density. Once the background mesh is created, the next step taken by the `snappyHexMesh` utility is splitting the cells at feature edges and surfaces, and removing the cells that are placed outside the computational domain. This way, a castellated mesh is created 9.3. In the same step, a first refinement process can be performed, by adding increasing the refinement level or what is the same, splitting several times some cells in particular. All this features are specified and can be modified in the `castellatedMeshControls` sub-dictionary of the `snappyHexMesh` utility.

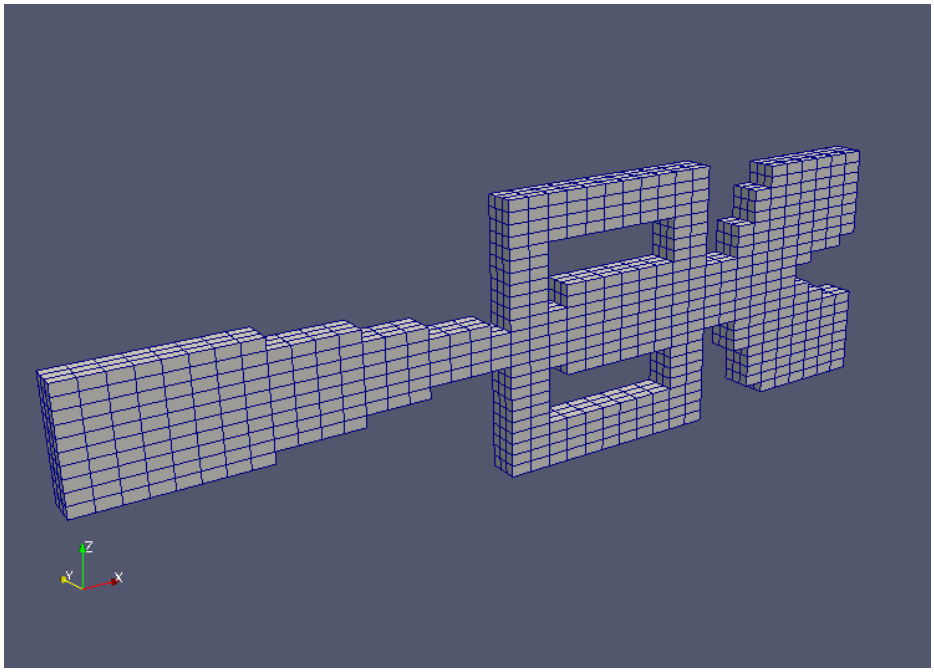

Figure 9.3: Basic castellated mesh using `snappyHexMesh` utility

In order to smooth the surfaces of the castellated mesh, another sub-dictionary of the `snappyHexMesh` is used: the `snapControls`. Its working principle consists on moving cell vertex points onto surface geometry 9.4.

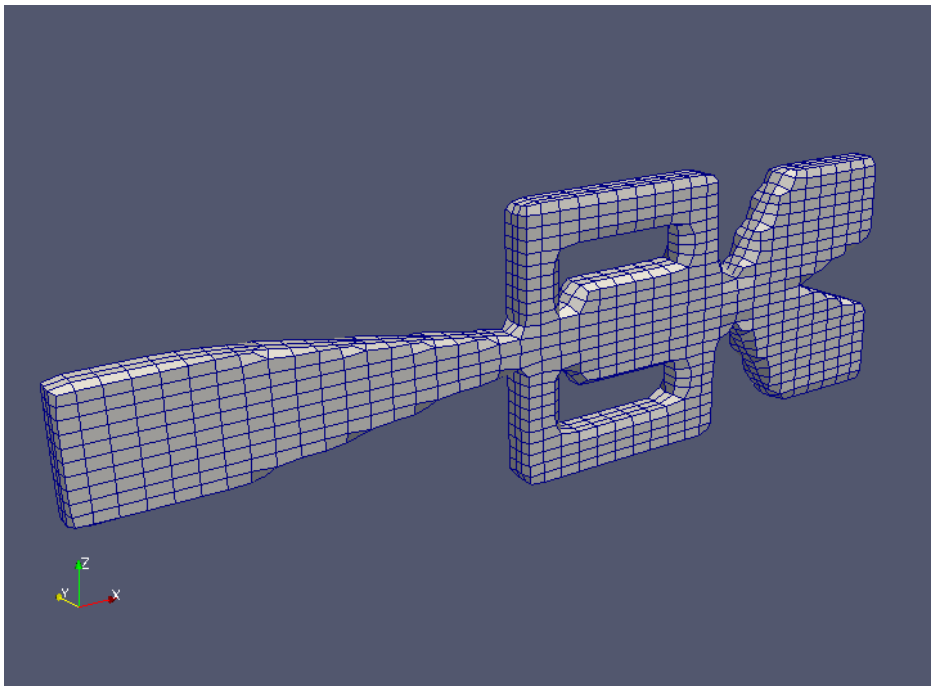

Figure 9.4: Basic snapped mesh using `snappyHexMesh` utility

Finally, although the resulting mesh of the snapping stage may be suitable enough, it can also produce some irregular cells along boundary surfaces. Solving this problem is the main purpose of the sub-dictionary `addLayersControls`. It introduces additional layers of hex cells aligned to the boundary surfaces. The `snappyHexMesh` utility also has a mesh quality control sub-dictionary called `meshQualityControls`, which purpose is to prevent any errors in the meshing that could lead the simulations to fail.

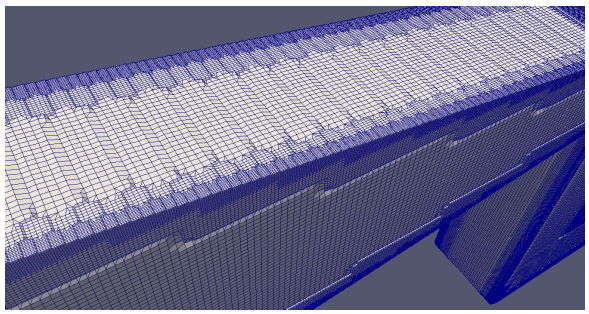

Figure 9.5: Finest mesh created using `snappyHexMesh` with irregularities on the external faces (a)

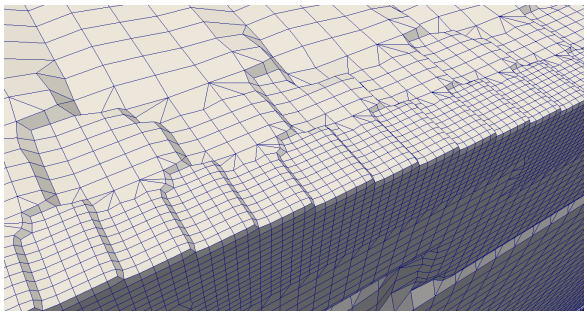

Figure 9.6: Finest mesh created using `snappyHexMesh` with irregularities on the external faces (b)

After several computations with parameters variations, the best mesh obtained with `snappyHexMesh` presented some problems regarding the geometric accuracy of the faces, as

it can be seen in figures 9.5 and 9.6. For this reason, the mesh that was finally used for the computations, was done entirely using `blockMesh` utility, by Baghaie, M. for his PhD in fluidic oscillators.

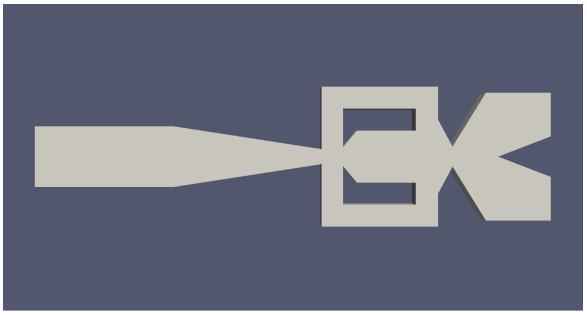

Figure 9.7: Final mesh by Baghaie, M. using `blockMesh` utility (a)

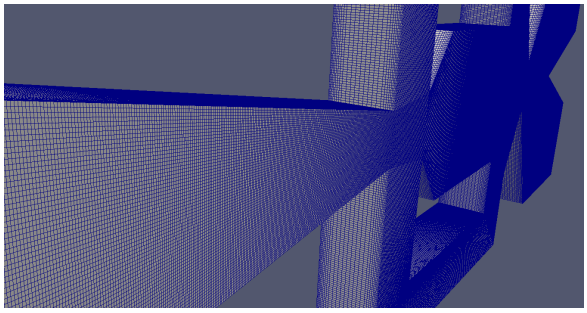

Figure 9.8: Final mesh by Baghaie, M. using `blockMesh` utility (b)

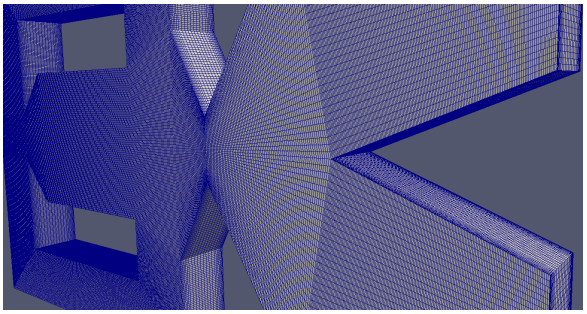

Figure 9.9: Final mesh by Baghaie, M. using `blockMesh` utility (c)

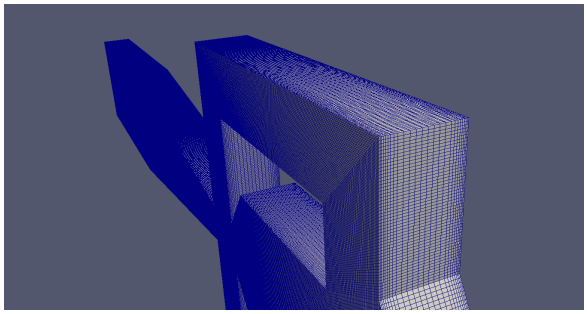

Figure 9.10: Final mesh by Baghaie, M. using `blockMesh` utility (d)

9.2.2 Mesh characteristics

The fluctuations in space of the turbulent flow has a wide range of scale. As bigger the Re number is, the smaller the fluctuations can be, so the mesh has to be finer, that is to say more refined. But in the other side, the mesh resolution has a strict limitation set by the computational power of the used hardware. For this reason, the hardware sets the maximum Re number in which the case can be computed. Although the computing power of the supercomputers is exponentially increasing, and the Re number in which the direct simulation of the Navier-Stokes equations can be computed is becoming larger, not everyone has access to this huge computational capacity.

The optimal size of the mesh cells, and its level of refinement is also dependant of the turbulence model used. $y+$ is a non dimensional parameter which value is a ratio between

turbulent and laminar influences in a cell. It is used to describe how fine or coarse a mesh should be for a particular flow pattern. It is important in turbulence modelling to determine the proper size of the cells near domain walls. If $y+$ is small then the cell is totally laminar and the next cell in has some laminar flow in it, the wall functions are not applied to this cell so bad modelling assumptions are made. High value of $y+$ implies that the cell is turbulent. If $y+$ is too big then the laminar/turbulent problem can be correctly treated, but other assumptions are invalidated in return.

In this particular study, the influence of $y+$ in the calculations has not been taken into account, since the used mesh belongs to the PhD thesis work of Baghaie, M. and it has been considered that it is optimal enough.

The mesh has 2242000 cells, most of them (2239250) are hexahedral, and the rest (2750) are polyhedral, and presents 25 layers of cells in the z-direction (width). It has 5 different patches:

- inlet
- outlet1
- outlet2
- walls
- topBottom

The only problem that can be found is the 100 high skew faces detected by the `checkMesh` utility, although it is not a critical issue.

10 Simulation

10.1 Simulation parameters and implementation

In order to get valid values from the computations, it is important to set the right values of the simulation parameters. In the root folder of the case, there is three different directories: `/constant`, `/system`, and the initial time directory `/0` where the initial conditions of the case are (after the computations, there will be more time directories, according to the `writeControl` parameters).

10.1.1 Constant parameters

The `constant` directory contains a full description of the case mesh in the `polyMesh` subdirectory, and some other files specifying the physical properties for the application of the simulation:

10.1.1.1 Thermophysical properties

This dictionary defines the thermophysical model characteristics, or how the energy, heat and physical properties of the fluid are calculated.

First of all, the thermophysical model and the mixture model have to be defined. In this case, as the solver belongs to the `compressible` family of solvers, the model used is the psiThermo (defined in the library as `hePsiThermo`) a model for fixed composition based on compressibility:

$$\psi = (RT)^{-1} \tag{10.1}$$

where R is the gas constant, and T the temperature.

As the oscillator does not present any reaction, the mixture type can be assumed as `pureMixture`, the general thermophysical model calculation for passive gas mixtures. The specie in the oscillator is air, with a molecular weight (`molWeight`) of 28.966 g/mol, and the parameter related to the number of moles of the component is set to 1.

Regarding the thermodynamic parameters, the model chosen is `hConst`, which assumes a constant C_p and a heat of fusion H_f.

- $C_p = 1004.8$
- $H_f = 2.544e + 06$

The form of energy to be used in the solution is the enthalpy (`sensibleEnthalpy`), and the equation of state of the fluid correspond to the perfect gas equation:

$$\rho = \frac{1}{RT}p \tag{10.2}$$

The transport properties of the fluid are included in the thermophysical properties dictionary, so it is not necessary to add an specific dictionary as it is with solvers that do not include energy or heat libraries. In this case, a constant (`const`) transport model has been selected, which assumes a constant μ and Prandtl number $Pr = C_p\mu/\kappa$. The values for those constant variables are:

- $\mu = 1.83632e - 05kg/ms$
- $Pr = 0.7$

10.1.1.2 Turbulence properties

As it has been explained, the turbulence model analysed in this study is the RAS `kOmegaSST`. For the extra required parameters, the default values are used.

10.1.2 Initial conditions

The initial conditions are specified in the directory of the first time step evaluated by the solver. As it has been set up the `startTime` as 0, the files related to those initial conditions are placed in the 0 directory.

The required parameters for the RAS k-ω SST calculations are the following: U, T, p, k, ω, α_t, ν_t.

10.1.2.1 U

The file U correspond to the fluid velocity at the various boundaries. It is set to 0 in all of them except for the inlet, where it takes the value of 173.034 m/s (the necessary velocity to reach $M = 0.5$ at the experiment temperature conditions).

10.1.2.2 T

The temperature is set to 298 K at all the boundaries and the internal field.

10.1.2.3 p

The initial and boundary values of the pressure is 10^5 pa (or $\frac{kg}{m\Delta s}$). It is important to notice that this OpenFOAM® does not work with reduced pressure values (p/ρ).

10.1.2.4 k

k represents the turbulent kinetic energy of the fluid. It can be initially approximated with the following expression:

$$k = 3/2(UI)^2 = 71.858m^2/s^2 \tag{10.3}$$

where U is the fluid reference velocity and I the turbulence intensity, defined as the ratio of the root-mean-square of the velocity fluctuations with appropriate values between 1

$$I = 0.16Re_{D_h}^{-\frac{1}{8}} = 0.04 \tag{10.4}$$

Re_{D_h} is the Reynolds number applied to the hydraulic diameter:

$$Re_{D_h} = \frac{UD_h}{\nu} = 55281.57 \tag{10.5}$$

where $D_h = 4,952mm$, $U = 173.034m/s$ and the kinematic viscosity at 298 K is $\nu = \frac{\mu}{\rho} = 1.55 \times 10^{-5}$

10.1.2.5 ω

ω is the specific turbulent dissipation rate. It is an essential parameter for the k-ω turbulent model. It can be estimated as:

$$\omega = \frac{k^{1/2}}{C_\mu l} = 446259.9 s^{-1} \tag{10.6}$$

10.1.2.6 α_t

α_t represents turbulent thermal diffusivity. It plays a very important role in the wall functions of the boundary conditions, and the turbulent heat flux modelling. Its dimensions are kg/ms.

10.1.2.7 ν_t

ν_t ($\nu_t = m^2/s$) represents the kinematic variable of the eddy viscosity, also called turbulent viscosity ($\nu_t = \mu_t/\rho$). It comes from the Boussinesq assumption that states that the momentum transfer caused by turbulent eddies can be modelled with an eddy viscosity 7.2.1.

10.1.3 System parameters

The `system` directory contains the corresponding files for setting the parameters associated with the solution procedure.

10.1.3.1 controlDict

The `controlDict` dictionary contains the essential input parameters for the creation of the database that controls the input and output of the case.

First of all, the solver application must be defined. In this case, `rhoPimpleFoam` (a transient solver for turbulent flow of compressible fluids for HVAC -Heating, ventilation and air conditioning- and similar applications).

The time control is one of the most important parameters. It controls the start and end time of the simulation, as well as the time step (Δt). Then it comes the Data writing controls, that enables the user to set the timing and format of the output files writing among other parameters, and the graph controls for plotting the solution.

Additional libraries and functions can be added to the simulation in the `controlDict` dictionary, such as the probes function that writes out the values of each time step of the specified fields (p and U in this case) in some specific previously defined points of the domain, displayed in table 10.1.

Probe	Coordinates (m)			Location
	x	y	z	
1	0.087	0.007125	0.001625	exit of the upper outlet channel
2	0.087	- 0.007125	0.001625	exit of the lower outlet channel
3	0.0585	0.01	0.001625	middle of the upper feedback channel
4	0.0585	- 0.01	0.001625	middle of the lower feedback channel
5	0.0504	0.00185	0.001625	upper side of the mixing chamber
6	0.0504	- 0.00185	0.001625	lower side of the mixing chamber
7	0.0665	0.0048	0.001625	inlet of the upper feedback channel
8	0.0665	- 0.0048	0.001625	inlet of the lower feedback channel

Table 10.1: Location of the probes

10.1.3.2 fvSchemes

The `fvSchemes` file contains the specification of which numerical discretisation schemes are used by the solver. It can be selected at run-time, or what is the same, it can be modified without being necessary to restart the simulation.

The used schemes for the RAS simulation are displayed in table 10.2:

Schemes	methods
ddtSchemes	Euler
gradSchemes	Gauss linear
divSchemes	Gauss linear or Gauss limitedLinear 1
laplacianSchemes	Gauss linear orthogonal
interpolationSchemes	linear
snGradSchemes	orthogonal
wallDist	method meshWave

Table 10.2: Schemes used for the RAS K-ω SST simulation

10.1.3.3 fvSolution

In the `fvSolution` file, the user can control the equation solvers, tolerances and algorithms.

The first sub-dictionary is `solvers`. It specifies each linear-solver that is used for each discretised equation. The solver can be:

- PCG/PBiCG: p reconditioned (bi-)conjugate gradient, with PCG for symmetric matrices, PBiCG for asymmetric matrices.
- smoothSolver: solver that uses a smoother.
- GAMG: generalised geometric-algebraic multi-grid.
- diagonal: diagonal solver for explicit systems.

Some of the solvers use a smoother, and it has to be specified which one among all the available smoothers is going to be used:

- GaussSeidel
- symGaussSeidel
- DIC/DILU
- DICGaussSeidel

The solver has some other parameters that have to be taken into account. First of all, the `tolerance` and `relTol` (relative tolerance), force the solver to stop if the residual falls below the `tolerance` value or if the ratio of current to initial residuals falls below the `relTol` value. Thus, the tolerance represents the level at which the residual is small enough so the solution reaches the desired accuracy. Another parameter of the solver is the `preconditioner`. Its definition and types can be found at Hirsch, C. "Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics" [16].

Moreover, if a relaxation factor is to be used in some of the solvers, it has to be introduced in the `fvSolution` dictionary.

Finally, a very important choice has to be made: the algorithm. There is three types of algorithms.

- PISO: (pressure-implicit split-operator)

- SIMPLE: (semi-implicit method for pressure-linked equations)
- PIMPLE: (PISO and SIMPLE combined algorithm)

PISO and PIMPLE are used for transient problems, whereas SIMPLE is used for steady-state. The used algorithm for the RAS k-ω SST simulation is PISO, while for the LES-RAS Spalart-Allmaras DDES simulation, the selected algorithm is PIMPLE. Whatever the selected algorithm is, some additional parameters need to be introduced, such as `nCorrectors`, `nNonOrthogonalCcorrectors`, `nOuterCcorrectors` or `momentumPredictor`. All of those parameters are set in their default values.

10.1.3.4 decomposeParDict

This dictionary is aimed to decompose the case to analyse in several separated parts, to be able to run it in parallel using various processors (or machines, if working in a local network).

10.2 Variations on the simulation parameters

In order to achieve a better convergence of the calculations, the different variations on the simulation parameters were done are the following:

- Originally, the time step (Δt) was not dynamic. It was set to 1.0e-6 s. To solve the floating point error, it must be reduced. For this reason, after setting the appropriate maximum Courant number, it was defined as dynamic, were its value is automatically calculated by the solver itself.
- Increment of the residuals tolerance: During the first simulations, the solver crashed several times after approximately 800 time steps. The tolerance in all the residuals was then set to 1.0e-04, instead of the default values of 1.0e-06 and 1.0e-05.
- Since the `rhoPimpleFoam` solver was not able to compute under compressible supersonic flow conditions (the flow gets supersonic inside the fluidic oscillator due to its geometry, mainly at the inlet nozzle), the first solution that was sought was to reduce the velocity inlet value. Moreover, some research was done on the compressible supersonic solver `sonicFoam`. Further explanation of this velocity reduction process is given in section 11.
- Other minor parameters were changed in order to optimize them for the calculations.

11 Results

In this section, a detailed report of the obtained results of the simulation is given.

The initial simulation (with an inlet Mach value of 0.50) crashed several times approximately at the time t = 0.0001725 s, as the maximum number of iterations was exceeded. After checking the results obtained so far it was clear what the problem was: the velocity values near the inner throat had reached supersonic values due to the convergent nozzle of the inlet, and the solver in use was not designed to compute under those conditions. In figure 11.1 the Mach values of the internal field are shown:

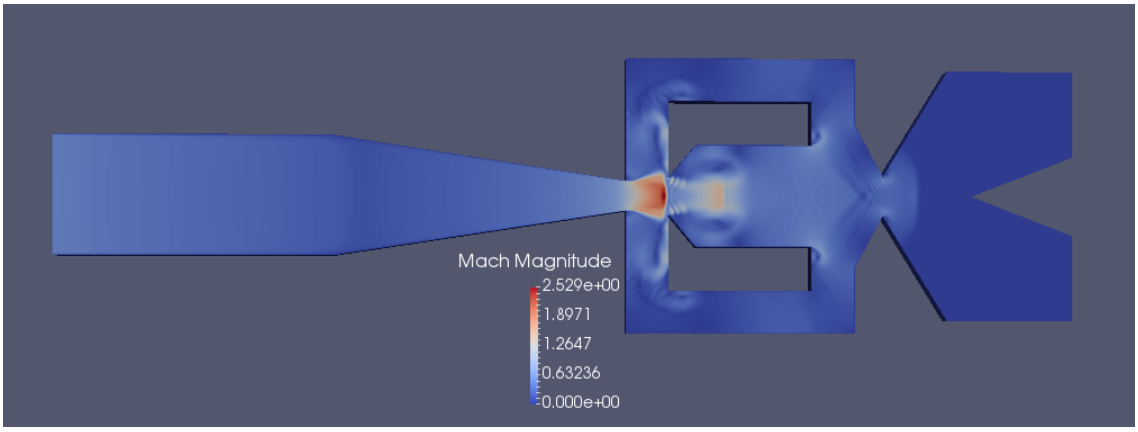

Figure 11.1: Mach values of the initial simulation internal field

The pressure field of figure 11.2 shows a very low pressure zone in the area where the highest Mach values are reached. Moreover, as a consequence of the supersonic flow, some pressure waves appear in the exit throat, and they propagate backwards blocking the entrance of the feedback channels.

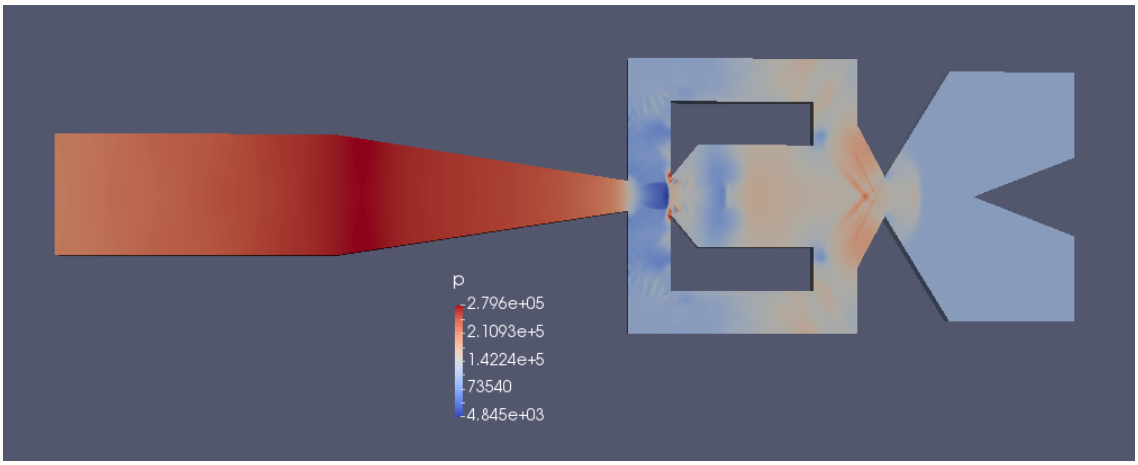

Figure 11.2: Pressure values of the initial simulation internal field

In order to face this problem and carry on with the simulation (not even a single oscillation had been simulated yet), two possible solutions were presented. On the one hand, the inlet velocity value had to be reduced to the point where all the internal flow field velocity could be treated as subsonic. On the other hand, further research of the OpenFOAM® solvers had to be made, to start working with a solver which enables the calculation of supersonic flows.

Reducing the inlet velocity value also reduces the maximum Mach value in the internal field. Table 11.1 shows the maximum values of the velocity for each simulation while varying the inlet value.

Inlet			**Internal field**	**Simulation**		
Mach	**U [m/s]**	**Re**	**Max. Mach**	**Last ts**	**Last written ts**	**Ending status**
0.50	173.034	55281.57	2.529	0.000172445	0.00017219	divereged
0.45	155.73	49753.223	1.611	0.00023632	0.000236078	aborted
0.40	138.43	44226.152	1.495	0.00022798	0.000223448	aborted
0.35	121.12	38695.886	1.415	0.000236312	0.00023601	aborted
0.30	103.82	33168.81	1.243	0.000281094	0.000275568	divereged
0.25	86.517	27640.786	-	-	-	failed
0.20	69.214	22112.757	1.339	0.000543004	0.00538636	divereged
0.15	51.90	16581.213	1.767	0.00105472	0.00104997	divereged
0.10	34.607	11056.378	1.298	0.000945026	0.000938398	divereged

Table 11.1: Inlet and internal field Mach values and simulation parameters

The increment of the maximum Mach value reached in the internal field on the $M = 0.15$ and $M = 0.10$ simulations despite decreasing the inlet velocity value is probably due to

reaching a more advanced stage of the simulations.

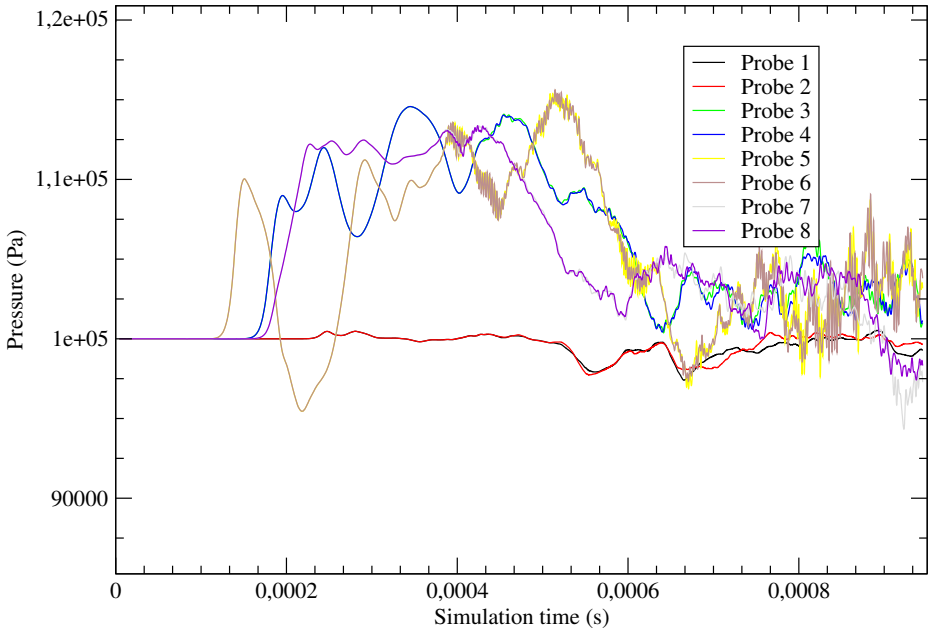

Figure 11.3: Evolution of the pressure field in the probe locations of Mach 0.10 simulation

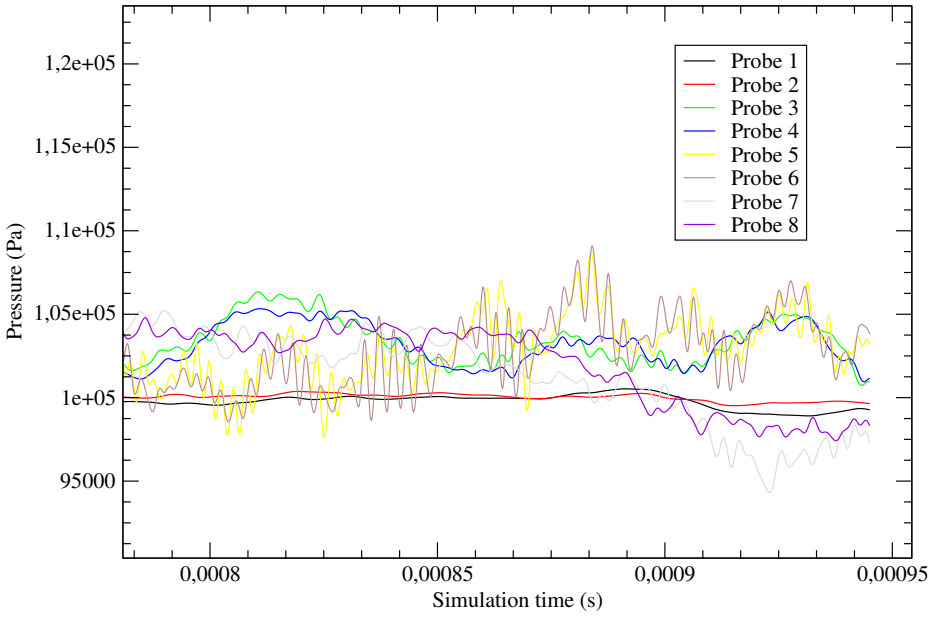

Figure 11.4: Evolution of the pressure field in the probe locations of Mach 0.10 simulation

As it can be seen, none of the simulations converged properly, as the internal field velocities kept increasing above sonic values, even though it was significantly reduced on the inlet. In figures 11.3 and 11.4 it is demonstrated that the stationary oscillatory stage is not

reached, as none of the probes placed in key locations of the oscillator presents a regular oscillation of the field values (i. e. the pressure field), although some of them present evidences of being on the edge of becoming oscillatory.

A new strategy was followed afterwards, by increasing the tolerance on the solver solutions during the transient phase, and decreasing it again when the stationary oscillation phase was reached, but the results were not appropriate at all. The high tolerance values induced the obtaining of some false results that progressively mislead to a total failure of the simulation. The previous, initial and final stage of this failure is shown in fig.11.5, 11.6and 11.7.

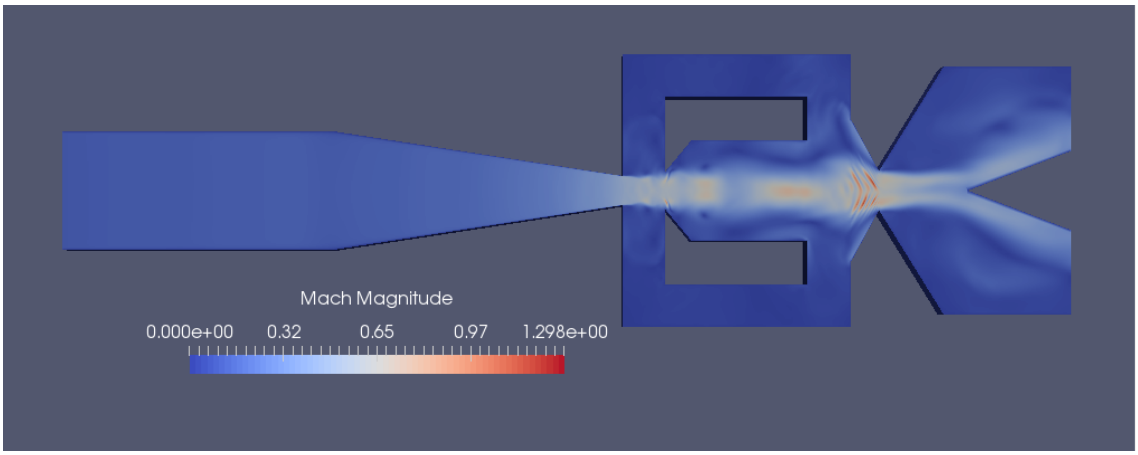

Figure 11.5: Mach values of the internal field of the M=0.10 simulation before failure

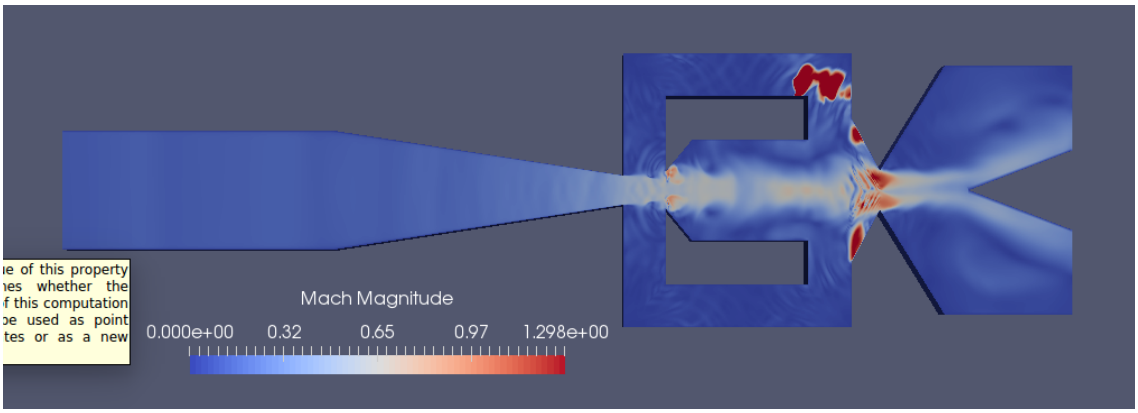

Figure 11.6: Initial false Mach values of the internal field of the M=0.10 simulation

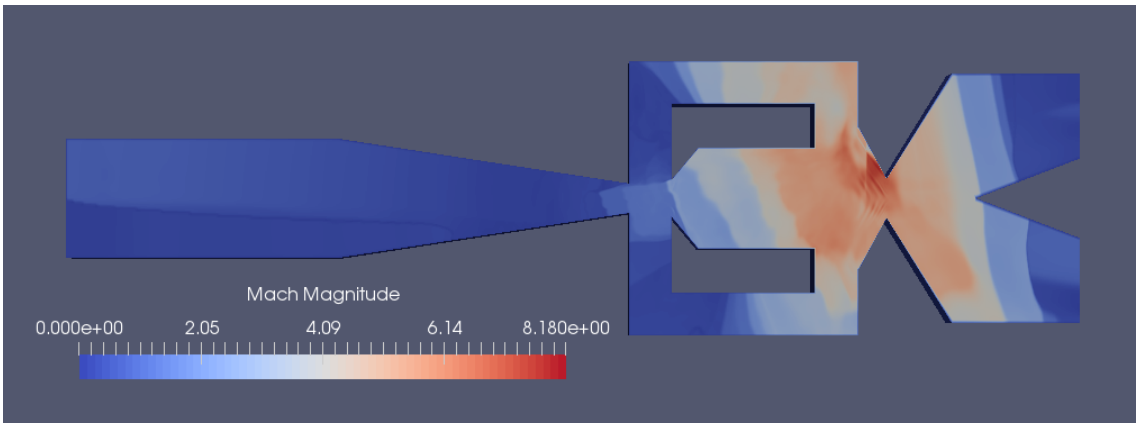

Figure 11.7: Final false Mach values of the internal field of the M=0.10 simulation

Some further simulations were launched with small variations of the parameters previously commented, but none of them presented better results that the ones shown in table 11.1.

Regarding the alternative to the `rhoPimpleFoam` solver, although a lot of time has been invested on the research in the literature of the `sonicFoam` solver, none of the simulations done with it were successful enough to take them into account in the results as all of them crashed during the 1st or the 2nd time step. It is the author's opinion, though, that further research on this topic is crucial for future work on fluidic oscillators with compressible flow.

12 Organization and planning

12.1 Work breakdown

In this section a brief description of the project's organization is made, comparing the scheduled and planned work structure with what has finally happened during the realization of this study.

The project was initially divided into different parts according to the different steps of the project. Those parts could be subdivided into tasks. This project was performed individually, so it was not possible to parallelize them. Some of those tasks presented several interdependency relationships among each other, that were defined in the Project Charter. After knowing the tasks and their interdependency relationships, a Gantt Chart was made in order to have a global vision of all the tasks of the study over the time-line. Its purpose was to determine the calendar of the project.

Although the original planning and organization was followed during the first weeks of work, there was an important change of plans that delayed the realization of the whole project a few months. After this period, in which some minor tasks were carried out, the main work was resumed until its completion some days before the due date, so that there was the possibility of calmly deal with possible delays or other unexpected problems.

Apart from this general delay of the realization of the project, it has been noticed while working on it that some of the planned tasks did not correspond to what they were thought to be at the beginning, when planning the project. There were several reasons for what this could happen such as a higher difficulty of the task, or an undiscovered interdependency relationship, etc. Although some further modifications had been done over the original planning, it has not deeply affected the work structure.

12.2 Future work

As this study related to the compressible behaviour of the flow inside fluidic oscillators has remained somehow uncompleted, further work is essential to accomplish the initial goals of the project. Some of the next steps to develop are the ones described below:

- Modify the mesh in two different aspects. On the one hand, reduce the remaining highly skew faces that the used mesh had, that can lead to potential failures, and in the other hand, increase the mesh domain, so that it does not end at the outlet edge of the oscillator. A bigger cavity should be added (connecting or not the two different outlets, depending on the final purpose of the designed oscillator), so that the flow stream does not abruptly reach the pressure conditions of the surrounding environment. Apart from being a more accurate description of the physical phenomena that take place in the oscillator, it would help the numerical calculations to converge faster.
- Modify not only the mesh, but the geometry of the oscillator itself (although it would diverge from the studies made so far with this particular geometry) to make it more suitable for compressible flow.
- Further research on the compressible supersonic solver `sonicFoam` should be done, in order to get a reliable results under those conditions, as the `rhoPimpleFoam` solver has been found to be useless in this case, where the velocity in the internal field rapidly increases to trans-sonic and sonic values despite the sub-sonic values on the inlet.
- Also, as it is proven in Calviño, P. study [13], the turbulence is far better modelled using a LES turbulence model rather than a RAS turbulence model. For this reason, a detailed study of the simulation of compressible flow under LES turbulent model conditions would be very useful, following the research line of the Aerospace Department.
- In addition, in order to validate the numerical results of this project and the compressible studies yet to come, an essential future work would be to empirically perform the experiment under the same conditions as in the numerical simulations. This would provide the needed tools to compare and evaluate the computational results, and vice-versa.

13 Environmental impact

The study of fluidic oscillators goes far beyond of being just an academic research subject. As it has been said, one of their countless uses is to work as a boundary layer control device on either wing or turbine airfoils, as well as other aerodynamic surfaces. Those boundary layer control devices are aimed to reduce flow separation on the boundary layer (an undesirable phenomena in aircraft high-lift systems and jet engine intakes), which translates into a parasite drag reduction.

The environmental benefits of the fluidic oscillators comes first of all from the drag reduction. The drag reduction implies a higher Lift/Drag ratio, one of the major goals in aircraft design. Delivering the same lift with lower drag leads directly to better fuel economy in aircraft, climb performance, and glide ratio.

The environmental impact of aviation occurs among other reasons because of the emission by the engines of particles and gases such as carbon dioxide (CO_2), water vapour, hydrocarbons, carbon monoxide, nitrogen oxides, sulphur oxides and black carbon. Some of them contribute to climate change and global dimming. A better fuel economy implies an emission reduction of those particles, which can help to reduce the environmental impact of aviation.

That being said, there is an other minor environmental benefit of the use of fluidic oscillators on aircrafts. As it has no moving parts, the risk of failure is lower, and also its maintenance, which can help to avoid using extra spare parts.

14 Conclusions and recommendations

The objective of this project was to go further on the line of investigation currently followed by the Aerospace Department, focusing on something new that still has not been the subject of many research: the compressible flow behaviour inside the fluidic oscillator devices.

The turbulent compressible flow case is much more complex than the incompressible. Hence, a lot of research on the literature has been made, as well as in the CFD specialized forum CFD-Online [25], were the author has found a lot of help that has been essential for the realization of the project.

Although all the research done on the literature, the obtained results have been quite disappointing according to the author's point of view. None of the simulations that have been carried out have reached convergence, so no reliable results have been obtained of the stationary oscillatory stage of the compressible turbulent flow inside a fluidic oscillator. Nevertheless, even though it could seem a paradox, the absence of results of this study is a result itself, as it will provide a basis on future research on the subject. The main hypothesis suggested by the author to give an explanation to this event is that maybe the used geometry of the fluidic oscillator was not adapted to the supersonic flow effects, possibly due to the sharp edges, that should be smoothed. For this reason, as it has been said, the geometry should be reconsidered in future work.

When working with computational fluid dynamics it must be understood that sometimes, for some unknown reason, the case just does not work, and the real job consists on finding the cause of the failure and a way to solve it, in order to launch again the simulation of the case. This process takes a considerable amount of time. An important thing to take into account when reviewing the results of this study is the available time for the simulations, which has not allowed the best of results.

Finally, it must be highlighted that there is still a lot of work to do in this field, a lot of research pending to be done, and this project aimed to contribute on this research, has definitely cleared and prepared the ground for further investigation and studies.

15 Bibliography

[1] B. C. Bobusch, R. Woszidlo, J. M. Bergada, C. N. Nayeri, and C. O. Paschereit, "Experimental study of the internal flow structures inside a fluidic oscillator," *Experiments in Fluids*, vol. 54, no. 6, 2013.

[2] M. Ruiz, "Estudio de osciladores fluídicos mediante mecanica de fluidos computacional. phd thesis," Dec. 12 2014.

[3] N. Tesla, "Valvular conduit," Feb. 3 1920. US Patent 1,329,559.

[4] L. N. Cattafesta and M. Sheplak, "Actuators for active flow control," *Annual Review of Fluid Mechanics*, vol. 43, no. 1, pp. 247–272, 2011.

[5] S. Raghu, "Fluidic oscillators for flow control," *Experiments in Fluids*, vol. 54, no. 2, pp. 1–11, 2013.

[6] P. Bauer, "Windshield washer," June 5 1979. US Patent 4,157,161.

[7] V. Tesař, *Pressure-driven Microfluidics*. Artech House integrated microsystems series, Artech House, 2007.

[8] W. Warren, "Negative feedback oscillator," Nov. 24 1964. US Patent 3,158,166.

[9] C. E. Spyropoulos, *Proceedings of the fluid amplification symposium*. No. v. 3 in Proceedings of the Fluid Amplification Symposium, Army Materiel Command, Harry Diamond Laboratories, 1965.

[10] S. Raghu, "Feedback-free fluidic oscillator and method," July 3 2001. US Patent 6,253,782.

[11] J. W. Gregory, J. P. Sullivan, and S. Raghu, "Visualization of jet mixing in a fluidic oscillator," *Journal of Visualization*, vol. 8, no. 2, pp. 169–176, 2005.

[12] B. Bobusch, "Experimentelle und numerische bestimmung der innendurchströmung eines fluidischen oszillators, diplomarbeit," Mar. 30 2010.

[13] P. Calviño, "Numerical study of three dimensional effects on fluidic oscillators. tfg thesis," June 12 2015.

[14] M. Saramiento, "Estudio de un oscilador fluídico mediante mecanica de fluidos computacional. tfg thesis," Dec. 9 2015.

[15] C. A. Candelo, "Estudio 3d de un amplificador fluídico mediante openfoam. tfg thesis," Jan. 20 2016.

[16] C. Hirsch, *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*. Elsevier Science, 2007.

[17] A. Sayma, *Computational Fluid Dynamics*. Bookboon, 2009.

[18] I. MD HELICOPTERS, "Notar technology®, www.mdhelicopters.com/v2/notar.php," 2014.

[19] M. Iguchi and O. J. Ilegbusi, *The Coanda Effect*, pp. 41–88. New York, NY: Springer New York, 2011.

[20] O. Krüger, "Cfd animation of the flow in a fluidic oscillator, http://fd.tu-berlin.de/en/research/projects/flow-control/fluidics," 2014.

[21] C. J. Campagnuolo and H. C. Lee, "Review of some fluidic oscillators," tech. rep., Washington DC 20438, USA, Apr. 1969.

[22] F. T. Brown, "A combined analytical and experimental approach to the development of fluid-jet amplifiers," *Journal of Basic Engineering*, vol. 86, no. 2, 1964.

[23] M. Sitter, "Internal flow of fluidic oscillators: A numerical and experimental investigation, master of science thesis," Dec. 8 2015.

[24] P. Gil, "Estudio de un amplificador fluídico mediante 3d-cfd. tfg thesis," Dec. 28 2015.

[25] "Cfd online discussion forums, main forum, url: http://www.cfd-online.com/," 1998-2016.

[26] W. Haase, B. Aupoix, U. Bunge, and D. Schwamborn, *FLOMANIA - A European Initiative on Flow Physics Modelling: Results of the European-Union funded project, 2002 - 2004*. Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer Berlin Heidelberg, 2006.

[27] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Comput. Phys.*, vol. 12, pp. 620–631, Nov. 1998.

[28] S. Raghu, S. A. Gokoglu, M. A. Kuczmarski, and D. e. Culley, "Numerical studies of a supersonic fluidic diverter actuator for flow control, technical report," tech. rep., Oct. 01 2010. Report/Patent Number: NASA/TM-2010-216805, AIAA Paper 2010-4415, E-17437.

[29] F. R. Menter, "Zonal two equation kappa-omega turbulence models for aerodynamic flows," *AIAA Fluid Dynamics Conference*, 1993.