Tutor/s

Dr. Joan Llorens Llacuna

UNIVERSITAT DE BARCELONA

Grau en Enginyeria Química

Treball Final de Grau

CFD. Conventional and Lattice Boltzmann methods. Study of LBM: Basic theory and applications. Simulation of a fluid in a wind tunnel.

Mariona Nicolàs Codina

June 2022

The very ink in which history is written is merely fluid prejudice.

Mark Twain

CONTENTS

SUMMARY

The aim of this work is to examine Lattice Boltzmann method (LBM) to see their possible application in Chemical Engineering problems. LBM is considered an efficient alternative method to conventional computational fluid dynamics (CFD) in some fluid dynamics problems, in which conventional numerical methods are limited.

LBM does not rely on numerical resolution of Navier-Stokes equation (NSE) like conventional CFD does. Contrary to CFD, LBM does not need fluid mechanics equations to describe fluid behaviour. Thus, mathematical complexity is reduced, and the efficiency of the method increases.

LBM is based on statistical probabilities of movement of fluid, specifically, particles, in streaming and collision processes. This method is founded, mainly, on the kinetic theory, which will be analysed in depth to understand the basis of the technique. Thus, LBM can describe fluid behaviour without solving

LBM, as CFD have application on Transport Phenomena simulation in fluid flow. In particular, conventional CFD is mainly implemented for industrial, environmental and physiological fluid fields.

Once studied the basis of the main representative methods of either microscopic and mesoscopic level and analysed its concepts entailed, a comparison among conventional CFD and emerging LBM, included in the work, permits simulating fluid behaviour efficiently by choosing the befitted technique. Likewise, limitations and applications of each method will be remarked. Small errors, simple application of the method, extensibility, higher parallelization and small-time procedure are some of the characteristics required for a well suitable method to simulate fluid flows.

Keywords: Lattice Boltzmann method (LBM), computational fluid dynamics (CFD), Navier-Stokes equation (NSE), kinetic theory, statistical probabilities, macroscopic and microscopic scale, Transport phenomena

RESUM

La dinàmica de fluids computacional (de l’anglès, CFD) es basa en l’anàlisi i la resolució de sistemes, els quals expressen en termes matemàtics la dinàmica de fluids, per mitjà de la simulació a través d’un ordinador.

Aquest treball se centra en l’estudi exhaustiu del mètode anomenat “Lattice Boltzmann”(LBM) i la seva aplicació en problemes d’Enginyeria Química. El mètode de Lattice Boltzmann és considerat un mètode eficient alternatiu als mètodes convencionals quan aquests presenten limitacions a l’hora de resoldre la dinàmica de certs fluids.

A diferència dels mètodes convencionals de CFD, el mètode de Lattice Boltzmann (LB) no es basa en la resolució numèrica de l’equació de Navier-Stokes (NSE) i, per tant, és capaç de descriure el comportament de fluids sense utilitzar les equacions mecàniques de fluids. Gràcies a aquesta particularitat, el mètode LB és un bon substitut dels convencionals. La complexitat matemàtica involucrada en el mètode LB és menor i l’eficiència és elevada en comparació amb altres mètodes.

El mètode LB es basa en probabilitats estadístiques del moviment dels fluids, representats com a partícules, en els quals es donen dos processos: col·lisió i transmissió. El mètode es basa, principalment, en l’anomenada teoria cinètica, la qual s’analitzarà detalladament per tal de comprendre els conceptes fonamentals en què es basa aquesta tècnica.

Tant el mètode LB com els convencionals són implementats per simular el flux de fluids relacionats amb fenòmens de transport. Concretament, els mètodes convencionals CFD intervenen en els camps de la indústria, el medi ambient o fisiològic.

Un cop estudiats els conceptes bàsics dels mètodes més representatius a escala microscòpica i macroscòpica s’analitzen les característiques dels mètodes convencionals i el mètode LB, els quals són tècniques de CFD. D’aquesta manera és possible determinar el mètode més adient per simular els fluids en les diferents situacions presentades. Un mètode apropiat serà aquell mètode fàcil d’usar a l’hora de simular, el qual presenti temps de simulació petits, elevada eficiència i errors matemàtics petits.

Paraules clau: mètode Lattice Boltzmann (LB), dinàmica de fluids computacional (de l’anglès, CFD), equació de Navier-Stokes (de anglès, NSE), teoria cinètica, probabilitats estadístiques, escala macroscòpica i microscòpica, fenòmens de transport.

1. INTRODUCTION

1.1. NUMERICAL FLOW SOLUTION: CONVENTIOTNAL CFD AND LBM

Most real fluid flows behaviours do not have mathematical analytical solutions due to the complexity involved in the resolution procedure, like non-linear equations. Only simple situations with simple geometries can be solved using analytical methods. However, numerical methods can be applied to get solutions for the most real fluid flow problems, and it can deal with complex boundary conditions.

Among these numerical methods, there are two important techniques to highlight: the conventional CFD, based on the numerical solution of the Navier-Stokes equation, and the emerging LBM, based on statistical particle-based methods. [Krüger, T. et al. 2017].

The Lattice Boltzmann method, historically developed from the lattice gas automata (LGA), is considered a new and highly efficient numerical method in the area of high performance computational fluid dynamics. [Körner, C., Pohl, T., Rüde, U., Thürey, N., & Zeiser, T. Parallel Lattice Boltzmann Methods for CFD Applications. 2006.]

In this work, both methods will be analysed and compared. Considering the advantages and disadvantages of each method, it is possible to decide the best technique for studying fluid behaviour in a wide range of situations.

Many processes and plants considered in Chemical Engineering field involve transport phenomena with fluid flow. Nowadays, the recent appearance of LBM could bring some benefits to solve certain types of chemical engineering problems. The wide range of LBM applications leads to an increased interest in studying this new approach.

1.2. CASE STUDY. SIMULATION FLUID FLOW

Once introduced LBM and studied the background and basic theory of the method, it is explained the implementation of the standard algorithm. Using computational tools it is possible to describe fluid behaviour. As a case study, it will be simulated a fluid flow inside a wind tunnel, using a software system called “Wolfram Mathematica” program. Moreover, other fluid situations such as a flow in a pipe with bends and obstacles and a flow over an airfoil will be shown.

1.3. TIME EVOLUTION OF CFD. NUMERICAL METHODS RESEARCH

To introduce the main object of study of this work, an interesting question is raised: Why Lattice Boltzmann methods as an efficient tool for simulating fluid behaviour? LBM as an attractive befitted method for covering CFD limitations. Computational Fluid Dynamics as a revolution tool for simulating fluid flow.

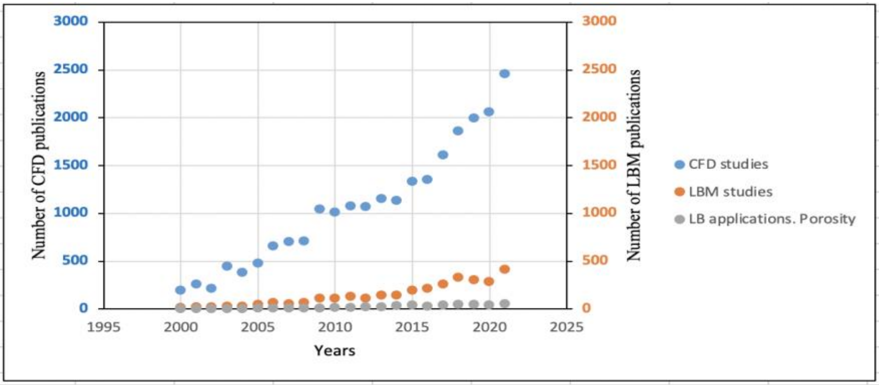


Figure 1. *Representation of the number of work publications from the year 2000 until now. Methods based on Computational Fluid Dynamics. Comparison among CFD, represented by the blue line, and Lattice Boltzmann method, represented by the orange line. It is observed a growing interest in both methods, emphasizing the amount of CFD work publications and the captivation of researchers for studying conventional methods. The grey line represents LBM main applications: porosity models. Implementation of LBM with porosity and porous materials.*

As shown in Figure 1, since the invention of computers, the interest in computational fluid dynamics (CFD), which solves the fluid mechanic equations, has steadily increased over the time. The number of published studies has been accelerated in the recent years because of the raise of suitable methods for simulating fluids.

Apart from the CFD conventional numerical methods another numerical method to be highlighted due to the large number of works published is called Lattice Boltzmann method (LBM). Research in both CFD and LBM have become object of study over the last 25 years.

In the Figure 1 it is shown a similar increasing tendency in both CFD and LBM. Even the higher interest in LBM over the last years, conventional CFD methods, based on the famous Navier-Stokes equation, are still studied. Researchers are attracted for comparing pros and cons of each technique and deciding the best method for simulate fluid behaviours.

The ease of implementation, extensibility and the higher efficiency and parallelization are the major reasons for LBM's growing field of application and increasing popularity. These properties make LBM promising approaches to model complex physical systems, especially fluid flow. Though it will not be a replacement for well-established CFD technology, it may have advantages in certain application areas.

The relatively low cost of CFD, in relation to experimental data, and the limited availability of analytical data, made it an integral part of design and analysis loops. Moreover, it is easy to adapt to new challenges and problems.

1.4. LIMITATIONS OF THE METHODS. NEW MODEL APPROACH

One fundamental problem in CFD is the discretization of the geometry, the expensive and time-consuming procedure of mesh generation. However, because of the rapidly increasing computational power and the development of new numerical methods like the Lattice Boltzmann method (LBM), the implementation of this tool on high performance supercomputer systems provides accurate flow solutions whilst avoiding time-consuming mesh generation. [Bernsdorf J., Durst F. and Schäfer M. (1999) Int. J. Numer. Met. Fluids, 29: 251–264.]

Lattice gas models suffers from some drawback, highlighting the statical noise. This disadvantage entails space and time averaging procedures to extract macroscopic quantities. Thus, new methods for covering these methods limitations of hydrodynamics should be studied. [Körner, C., Pohl, T., Rüde, U., Thürey, N., & Zeiser, T. Parallel Lattice Boltzmann Methods for CFD Applications. 2006.].

Due to the rapidly development of the LBM technique, LBM lacks from a systematic and flexible construction of numerical schemes in some situations such as mixture models with a given Schmidt number, or thermal models with a given Prandtl number. Consequently, a new recently LB method for simulating fluid behaviour with tailored transport coefficients has been introduced. This new method, called “Quasi-equilibrium” kinetic model is considered a befitted technique for dealing with this problem. [Gorban, A. Quasi-Equilibrium Lattice Boltzmann Method. The European Physical Journal B. 2007.]

2. OBJECTIVES

The main objective of this work is the study of LBM method, compared with other existing methods, and its main field applications in chemical engineering.

Even a fluid behaviour can be described simulating the problem involved in a computer, using conventional CFD methods like the “finite volume method”, fluid mechanic equations entailed can be solved. With the aim of covering conventional methods limitations, other attractive methods, such as lattice gas models like LBM, or multi-particle collision dynamics has been researched.

The implementation of new suitable techniques for dealing with complex geometries schemes leads to study the called Lattice Boltzmann method (LBM), as a new approach for covering conventional CFD limitations.

CFD is still an attractive field for researchers and the invention of supercomputers has increased the interest of simulating fluid behaviour. In this work it will be simulated several fluid flows in different situations such as a flow in a box, a fluid travelling through a wind tunnel or over an airfoil. Hence, the influence of a moving object, vorticity and differences pressure issues will be analysed.

3. CONVENTIONAL NUMERICAL METHODS FOR FLUID FLOW

3.1. HISTORY AND EVOLUTION

The conventional numerical methods, based on discretizing the equations of fluid mechanics using a method of approximation, represents the fluid variables as values at different points of a volume. The node values are used to approximate the derivatives in the corresponding partial derivative equation (PDE). [Krüger, T. et al. 2017].

The aim of these methods is to express the fluid behaviour of complex geometries schemes, by equations, which cannot be solved without a computer. Due to the required technology for solving the equations, field of computational fluid dynamics (CFD) started soon after the advent of electronic computers.

Several conventional methods used for finding fluid solutions such as finite difference, finite volume and finite element method, have been invented.

3.2. CHARACTERISTICS AND CONVENTIONAL NUMERICAL METHODS

Thanks to conventional methods, the fluid behaviour can be described using the equations of fluid mechanics through its conservation of mass and momentum. The macroscopic fluid mechanics equations are directly discretized, and solved on a computer, using a particular method of approximation, among them, Euler approximation of a time derivative, considered the fastest one to step the solution forward in time.

Among the conventional methods, still available today, the ones to be mentioned due to its higher importance in research field are the finite difference, the finite volume and the finite element methods, underlining the finite volume as the most commonly applied method.

The conventional methods mentioned are distinguished for representing the fluid variables as values at different nodes over a lattice. The interpolation of this points varies according to the method applied. Whereas the finite difference method is based on a field defined as a square grid of nodes, the finite volume method represents a node value by averaging the variable in a small volume around the node. In contrast, the finite element method approximates the continuous field by an interpolation of the node values.

Next, will be included a brief explanation of each of the conventional methods most used by researchers.

3.1.1. Finite difference method

On each of the nodes of the square, the solution variables are represented by a number, represented by λ. The derivatives of λ are approximated by a linear combination ("finite differences") of λ_j, where "j"denotes the nodes of the grid. Considering the Taylor series of $\lambda\ (x)$, where "x" denotes nodes positions, there are obtained three approximations for the first-order derivative and one for the second-order derivative. Additionally, the truncation errors of the corresponding approximations can be obtained by using the Taylor expansion.

In general terms, the finite method is based on taking a set of equations and replace the derivatives by finite difference approximations. This fact leads to consider finite difference method as one of the simplest to be used.

Even the easy application of the method there are also some weaknesses, important to be mentioned. Due to the non-conservative scheme, the numerical errors entailed leads to imperfections in mass, momentum and energy conservation terms. But the most disadvantage of the method, for which other CFD methods for simulating fluids are considered is the complexity of the geometries derived from its irregular grids, hardly to use in practice.

3.1.2. Finite volume method

The finite volume method (FVM) is the most used conventional Navier-Stokes solvers and is characterized for subdividing the space into smaller irregular volumes.

Compared with the finite difference method, the irregular elements involved in the finite volume method allows a better representation of complex geometries.

This method is not as general as FD method because the control volume formulation leads to a perfect conservative system, for mass and momentum terms. Even this methos is well suited for complex domain geometries the difficulty increases when generating the fitting irregular grid.

3.1.3. Finite element method

In finite element method (FEM) the PDEs is solved by integrating over the domain. Even FEM can be efficiently applied in unstructured grid and for increasing the order of accuracy, it is not a conservative method as FV methods are and the complexity involved is higher than in FD and FV methods. Thus, the integrals are tricky to work with and the mathematical complexity in Navier-Stokes equations grows.

The conventional methods present limitation in some cases. The difficulty involved in the equations of fluid mechanics leads to a non-linear, simultaneous system of equations, tricky to solve in some situations. Generally, difficulty increase when dealing with fluid turbulence or in complex geometries. Moreover, due to the implicit pressure from the incompressible Navier-Stokes simulations, the conventional methods mentioned are abandoned and substituted by other CFD methods. The suitable alternative ones are the particle-based methods, in which its not needed to solve the equations of fluid mechanics directly.

4. PARTICLE-BASED METHODS

Whereas conventional "Navier-Stokes" solvers, from a macroscopic point of view, are based on discretizing the equations of fluid mechanics the new approach for finding fluid solutions, called "particle-based methods", use particles to represent the fluid. These techniques are based on microscopic and mesoscopic scale.

Next, the most important particle-based models will be studied, highlighting the LBM for its higher interest in research.

The simplest microscopic method, called the molecular dynamics (MD) method, represents particles as atoms or molecules and is founded on finding the neighboring position of the atom using the previous and current position of the scheme. The MD method is generally used for solving fluid involved in chemical reactions and phenomena like phase change.

Other methods for simulating fluid behaviour are available today and, due to its higher efficiency, MD can be substituted for other particle-based methods.

Among these alternative methods, direct simulation Monte Carlo (DSMC), dissipative particle dynamics (DPD), multi-particle collision dynamics (MPC) and lattice gas models, highlighting the last one for being the background of LBM, which is the object of study of this work.

4.1. LATTICE GAS MODELS

The first Lattice gas models called HPP model, was introduced by Hardy, Pomeau and de Pazzis in 1973. It is based on a 2-dimensional gas model, consisting in a lattice in which streaming and collision steps take place, respecting the mass and momentum conservation. In 1986 appeared another lattice gas model, named FHP, due to the inventors of this method (Frisch, Hasslacher and Pomeau). While HPP model is based on the square lattice and four velocities, the FHP model has a triangular lattice and six velocities. [Krüger, T. et al. 2017].

The algorithm of lattice gas models is based in Boolean variables. The mainly variable that characterizes the description of fluids, associated to velocity, is called “occupational number”. The possible values are 0 or 1, which means the non-existence or existence, respectively, of the particle in the lattice.

During the simulation of a fluid behaviour take place two steps in the lattice gas: streaming and collision.

It is represented, graphically, the two processes that take place in the lattice of the fluid.

Figure 2 shows the collision process of two particles and its possible redistribution, which both diffusive ways have the same probability to occur. When particles collide in a lattice node they are redistributed, considering the conservation fluid laws. The collision step is a simple and algebraic local operation.

Once collision has occurred, the streaming process takes place. The representation of the streaming process of the lattice gas model is shown in Figure 3. Particles travel from a node to a neighbour node with an associated velocity, c_i. The particles velocities, c_i, move exactly from one node to its neighbour in exactly one time step.

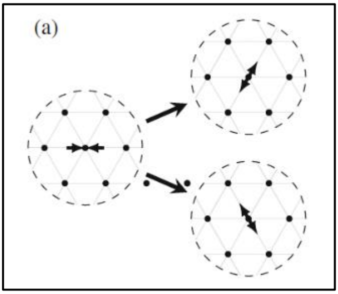


Figure 2. *Two particles from neighbour nodes meet in the same node of the lattice and, then, collide. Conservation laws are taken into account. The post-collision node is decided aleatory from the two possible paths, graphically represented. Lattice gas models.*

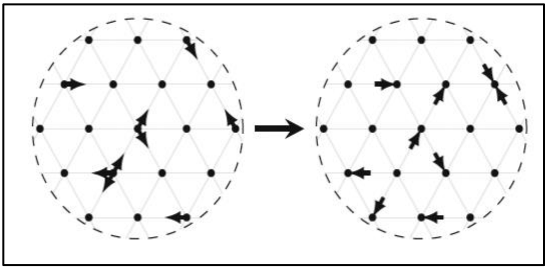


Figure 3. *Representation of the streaming step. Lattice gas models. Particles with an associated velocity move due to the collision.*

Equations 1 and 2 expresses, mathematically, both processes in which the method is founded.

$$n_i^*(x,t) = n_i(x,t) + \ \Omega_i(x,t) \qquad (1)$$

$\mathrm{n_i^*}$ is the post-collision occupation number

$\Omega_\mathrm{i} \in \{-1,0,1\}$ represents a collision operator, based on all occupation numbers from a node

$$n_i(x + c_i \cdot \Delta t) = n_i^*(x,t) \quad (2)$$

Both processes are mathematically expressed in Equation 3. [Krüger, T. et al. 2017].

$$n_i(x + c_i \cdot \Delta t, t + \Delta t) = n_i(x,t) + \Omega_i(x,t) \qquad (3)$$

The main advantage of the lattice gas model resides in the small error involved in simulating fluids, compared with other CFD methods. Thanks to Boolean variables, it is assumed that a particle is either residing or not residing in the lattice. Thus, particles collision, considered a perfect process, does not generate much error. Consequently, lattice gas models are considered high-powered methods, distinguished for being extremely paralyzed.

However, for a huge number of velocities, the complexity of the method increases, and other methods, more suitable, can be a better alternative for simulating fluids. Moreover, lattice gas models are not a good choice when dealing with high Reynolds numbers.

The statistical noise is the principal limiting point in the lattice gas models. Although the lattice gas remains in the equilibrium, the macroscopic variables involved will fluctuate. This difficulty was mostly resolved with the invention of Lattice Boltzmann method (LBM) in the late 1980s.

4.2. MULTI-PARTICLE COLLISION DYNAMICS

The multi-particle collision dynamics (MPC), which shares similitudes with Monte Carlo simulation method, DSMC, is a particle-based solver introduced in 1999 by Malevanets and Kapral. The ease of the method implementation and facility to be paralyzed steams from the stability involved, thanks to the mass and momentum conservation. [Krüger, T. et al. 2017].

Compared with other particle-based methods already mentioned, MPC should consider thermal fluctuations and hydrodynamic interactions.

MPC is commonly applied in systems with small mean free path. It works with efficient results for systems with Reynolds and Péclet numbers ranged from 0.1 to 10, not being well suited for Reynolds values near zero.

When hydrodynamic interactions are not required, other methods like Langevin or Brownian Dynamics are good alternatives to substitute MPC. Furthermore, even LBM and other conventional Navier-Stokes solvers are suitable methods compared with MPC when thermal issues are not desirable, it fits accurately in couple systems of solvents and solutes. It is also used for describing polymers and biological cells in equilibrium, likewise, external flow fluids.

5. CFD. BACKGROUND, BASIS AND ANALYSIS OF THE MAIN FLUID EQUATIONS. BASIC THEORY

5.1. BACKGROUND AND BASIS OF CFD

The high difficulty in some fluid simulations due to the lattice geometries complexity leads to the implementation of computational fluid dynamics.

Computational fluid dynamics, usually abbreviated as CFD, is a branch of fluid mechanics that uses numerical analysis and algorithms to solve and analyze problems that involve fluid flows.

The main equations to be solved are the continuum one and the NSE. However, depending on the simulated fluid conditions, additional equations, like the equation of state, will be introduced to define the system.

With the advent of CFD in recent decades, it has become an important tool in dealing with complex issues on different fields, especially, in engineering applications. Although in the beginning it was only used by limited number of researchers and academics in aeronautical and astronautics, today it is an integral available tool with a wide variety of applications. Among them, CFD is mainly implemented for industrial, environmental and physiological fluid fields. In Table 1 are mentioned the common areas in which CFD is mostly applied.

5.2. ANALYSIS OF THE MAIN FLUID EQUATIONS: BASIC THEORY

In this section it’s given an overview of fluid dynamics, in which will be studied the continuum equation, the Navier-Stokes equation (NSE), and the state’s one. The ones mentioned are the main equations of fluid mechanics used for describing a fluid behaviour. They are direct consequence of mass, momentum and energy conservation. Before analyzing these equations, it’s important to remark the “continuum” concept, which should be understood as operating at length and time scales sufficient large that the atomistic picture can be averaged out. Because of describing fluid dynamics terms from a macroscopic point of view, any fluid is considered continuum.

5.2.1. Continuity equation

In this section it will be analyzed, step-by-step, the continuum equation of a fluid, which reflects mass conservation. [Krüger, T. et al. 2017].

The mass of a fluid element can be expressed as shown in Equation 4.

$$\int_{V_o} \rho \, dV \qquad (4)$$

It is important to bear in mind that a fluid mass cannot be created or destroyed. Hence, the responsible for the change of this mass per unit time must be the fluid flow into or out of the stationary volume, V_o.

In mathematical terms, this statement is demonstrated in Equation 5.

$$\frac{\partial}{\partial t}\int_{V_o} \rho \, dV = \oint_{\partial V_o} \rho u \, dA \qquad (5)$$

$$\rho u\!: mass\ flux\ density\ or\ momentum\ density$$

Considering the divergence theorem for transforming the surface integral on the right-side of the equation into a volume integral. The resulting expression is shown in Equation 6.

$$\int_{V_o} \frac{\partial \rho}{\partial t} \, dV = \int_{V_o} \nabla \cdot (\rho u) \cdot \, dV \quad (6)$$

The result is the mass conservation of a continuum fluid can be expressed as a partial differential equation (PDE), Equation 7, commonly known as “continuity equation”.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \qquad (7)$$

5.2.2. Navier-Stokes equation (NSE)

The change of net momentum of an ideal fluid element is a consequence of either of the followed factors:

- Flow of a momentum into or out of the fluid element because of mass conservation
- Pressure differences, p
- External body forces, F

Thus, the momentum balance equation can be expressed as a sum of these factors' contributions, as shown in Equation 8.

$$\frac{\partial}{\partial t}\int_{V_o} \rho u \, dV = -\oint_{\partial V_o} \rho uu \cdot \, dA - \oint_{\partial V_o} p \, dA + \int_{V_o} F \, dV \qquad (8)$$

Using the divergence theorem to transform the surface integrals into volume integrals, the momentum balance equation can be expressed as seen it in the Equation 9.

$$\int_{V_o} \frac{\partial(\rho u)}{\partial t} \, dV = -\int_{V_o} \nabla(\rho uu) \, dV - \int_{V_o} \nabla p \, dV + \int_{V_o} F \, dV \qquad (9)$$

This leads to a new PDE, called the "Euler equation" or more commonly known as "Navier-Stokes equation", whereby the conservation of the momentum of an ideal fluid can be described in the Equation 10.

$$\frac{\partial(\rho u)}{\partial t} + \nabla(\rho uu) = -\nabla p + F \qquad (10)$$

The Cauchy momentum equation, Equation 11, uses a momentum flux density tensor term, Π, to express the Euler equation in a more general way.

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot \Pi = F \qquad (11)$$

The momentum flux density tensor is defined by Equation 12.

The stress term implicated, $\sigma_{\alpha\beta}$, is associated to simple isotropic fluids from the Euler equation. Whereas in the case of simple fluids the momentum flux transfer is reversible, for real fluids dissipative and irreversible issues requires a viscous term. [Krüger, T. et al. 2017].

$$\Pi_{\alpha\beta} = \rho u_\alpha u_\beta - \sigma_{\alpha\beta} \qquad (12)$$

$$\sigma_{\alpha\beta} = -\rho \; \delta_{\alpha\beta} \qquad (13)$$

The Equation 12 only includes reversible momentum transfer, considering either mass flow or pressure forces for being the conservative terms.

The description of a simple fluid involves an isotropic stress term in Euler's equation. Mathematically expressed in Equation 13.

Hence, the general viscous stress tensor, σ', can be defined as shown in Equation 14.

$$\sigma'_{\alpha\beta} = \eta \left((\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right) + \zeta \delta_{\alpha\beta} \frac{\partial u_\gamma}{\partial x_\gamma} \quad (14)$$

η and $\zeta$are viscosity coefficients

For simple uniform fluids and considering the stress tensor as a contribution of pressure and viscosity terms, it is obtained a new expression, Equation 15, to represent NSE.

$$\frac{\partial(\rho u_\alpha)}{\partial t} + \frac{\partial(\rho u_\alpha u_\beta)}{\partial x_\beta} = -\frac{\partial \rho}{\partial x_\alpha} + -\frac{\partial}{\partial x_\beta} \cdot \left[\eta \left((\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right) + \left(\eta_\beta - \frac{2\eta}{3} \right) \frac{\partial u_\gamma}{\partial x_\gamma} \delta_{\alpha\beta} \right] + F_\alpha$$

$$(15)$$

Assuming constant viscosity, the resulting NSE can be defined as shown in Equation 16.

$$\rho \frac{D u_\alpha}{Dt} = -\frac{\partial \rho}{\partial x_\alpha} + \eta \frac{\partial^2 u_\alpha}{\partial x_\beta \partial x_\beta} + \left(\eta_\beta + \frac{\eta}{3} \right) \frac{\partial^2 u_\alpha}{\partial x_\alpha \partial x_\beta} + F_\alpha \qquad (16)$$

The incompressible NSE, Equation 17, is a simplified expression of the NSE, assuming constant fluid density. [[Krüger, T. et al. 2017].

$$\rho \, \frac{Du}{Dt} = -\nabla \rho + \eta \Delta u + F \qquad (17)$$

The "Laplace operator", Δ, can be expressed as shown in Equation 18.

$$\Delta = \frac{\partial^2}{\partial x_\beta \partial x_\beta} \qquad (18)$$

5.2.3. Equation of state

The fluid behaviour can be studied through continuity and Navier-Stokes equations (NSE), which describes mass and momentum conservation, respectively. The system is composed of four equations, three of them derived from the NSE due to the three spatial components from the three-dimensional scheme.

The system is not closed because of the insufficient number of equations (only four equations, already mentioned) and the amount number of unknows (density, pressure and three velocity components) involved in the continuity and NSE. Hence, the system cannot be solved unless variables are fixed. An additional equation is needed and, hence, the system will be well characterized.

Generally, the new expression introduced in the system is founded on the “state principle of equilibrium thermodynamics”. This new concept relates the state variables that describes the local thermodynamic state of the fluid, such as density, pressure, temperature, internal energy, and entropy, through an equation of state.

The most renown equation of state is the “ideal gas” law, Equation 19. In the “ideal gas” law the pressure is related to the density and the temperature.

$$p = \rho RT \qquad (19)$$

The equation mentioned, Equation 19, can be developed to the isothermal equation of state, meaning that the temperature is constant (T ~ To).

Considering constant temperature (T ~ To) the Equation 20 is transformed to Equation 23, known as:

$$p = \rho RT_o \qquad (20)$$

This expression evidences the linearity relationship between the pressure and the density.

In case of incompressible fluid, $\rho = \rho_o$, there is no need of any other equation to define the system because the density is a fixed variable, and then the system is completely specified.

5.3. FIELD APPLICATIONS OF CFD

Over the time CFD has gain power thanks to the innovative technology and the invention of high-speed supercomputers. Thus, better solutions are achieved, and complex issues can be solved. Consequently, CFD has expand the field of applications. Apart from getting information of fluid behaviour, is now considered a numerical efficient tool for predicting transfer of heat and mass (such as dissolution), phase change, chemical reaction and mechanical movement, among other applications.

The main fields in which CFD implemented these days are shown in Table 1.

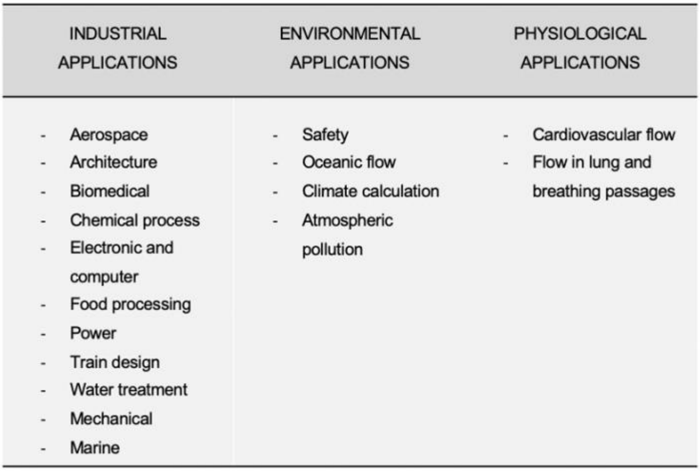
5.4. LIMITATIONS OF CFD

CFD is limited by propagation issues in which PDE is required on an open lattice. Thus, for unsteady-inviscid flow and steady-state inviscid supersonic CFD is difficult to implement. It also implies problems in equilibrium terms, related to boundary condition. Incompressible inviscid flow and steady-state temperature distribution are clear examples for this second situation.

The Mach Number is a dimensionless quantity defined as the ratio of the flow velocity past a boundary to the local speed of sound. Depending on its value it is measured the efficiency of the method. The Mach Number is mainly used to determine the approximation with which a flow can be treated as an incompressible flow.

Segregated or couple algorithms will also be analyzed, whose equations are solved in a different way. Whilst segregated fluids solvers are based on a sequential process, for couple fluids the procedure is simultaneous. Generally, incompressible flows are solved using segregated solvers and high-speed compressible flows were associated to couple algorithms. Coupled flow requires more computational time and high memory but also is more stable in high density fluctuations like in supersonic flows with collisions.

Table 1. *Applications of CFD in different industries: industrial, environmental and physiological.*



INDUSTRIAL APPLICATIONS	ENVIRONMENTAL APPLICATIONS	PHYSIOLOGICAL APPLICATIONS
- Aerospace - Architecture - Biomedical - Chemical process - Electronic and computer - Food processing - Power - Train design - Water treatment - Mechanical - Marine	- Safety - Oceanic flow - Climate calculation - Atmospheric pollution	- Cardiovascular flow - Flow in lung and breathing passages

6. STUDY OF LBM: THE NEW BRANCH FOR STUDYING FLUID BEHAVIOUR

6.1. INTRODUCTION OF LBM

The lattice Boltzmann method is a new and highly efficient numerical method in the field of high performance CFD. The main limitation of CFD comes from the geometry's discretization of the fluid, where mesh generation consume an excessive amount of time. When using regular orthogonal lattices mesh generation can be done almost automatically by regular discretization of arbitrary complex geometries. To achieve that situation is important to work with high technology, and powerful computers are needed.

By the rapidly increasing of computational power and the development of new numerical methods like LBM over the last recent years, it is meanwhile possible to handle with several tens of millions of grid points on a supercomputer. Thus, the time involved in creating the grid is extremely reduced. Consequently, an efficient tool for constructing these regular lattices and solving challenging flow problems, called lattice Boltzmann method, has emerged.

The lattice Boltzmann method, based on the evolution of statical distribution on lattices, has recently emerged as a powerful tool for solving challenging flow problems. It is an alternative approach to Navier-Stokes equations for computational fluid dynamics (CFD).

While Navier-Stokes equations, as commented before, consider fluid motion on a macroscopic scale, the lattice Boltzmann method describes particle physics on a mesoscopic level. The mesoscopic scale tracks representative collections of molecules, instead of individual particles, as occur in macroscopic and microscopic description. In contrast to Navier-Stokes solvers, which need to treat the nonlinear convection term, LBM does not include such linearity.

In this chapter will be explained the basis of the Lattice Boltzmann method, considered a particle-based method founded on the kinetic theory. Thus, kinetic theory, related to the mesoscopic scale, will be analysed in detail.

Interest in the lattice Boltzmann method has become a vast research field in the last 25 years and it has steadily increase, since it grew out of lattice gas models in the late 1980s. These methods are applied with the aim to simulate the flow of liquids and gases through the behaviour of a gas.

Due to the simplicity of the LBM, its scalability on parallel computers and the ease with which it can deal with complex geometries the interest for doing research in Lattice Boltzmann field is constantly increasing and extending.

6.2. **HISTORY AND EVOLUTION**

The Lattice Boltzmann method, introduced in 1988 thanks to Mc. Namara and Zanetti, has origin in LG (Lattice Gas), a microscopic model for gases that allows the simulation of fluid flows with simple fluid models. In LGA, the fluid is treated as a set of simulated particles residing on a regular lattice, in which, mainly, two processes take place: streaming and collision.

Due to the locality mentioned above, the relevant computer programs may be effectively parallelized. This fact allows the reduction of time simulation, compared with other numerical methods.

Even LBE, as LGA (Lattice Gas Automata), could also be derived from the Boltzmann equation, LBE has quite difference compared with the traditional CFD algorithms, which makes LBM to be considered as an improved method. Boltzmann equation is based on the description of the interaction of a single fluid particle or an ensemble averaged particle density distribution function.

6.3. ANALYSIS OF LBM: KINETIC THEORY

In this section some essential concepts as the assumption of local equilibrium in LBM, the Boltzmann equation, including the collision operator and its approximations, and the Boltzmann $\mathcal{H}$-Theorem will be analyst.

The kinetic theory can be understood as a mesoscopic fluid description on which the LBM is founded and describes a fluid behaviour by using mathematical expressions. The mesoscopic concept, which lies between the microscopic and macroscopic scale, refers to a distribution of several molecules among a gas scheme.

Even kinetic theory can be used to describe any fluid flow, it is most applied to the simplest case of a dilute gas, meaning that the time spend for the collision process is very little. In other words, the molecules collide, almost always, one-on-one. However, this assumption is not valid for all liquids and gases due to the increase of collisions. In liquids with strength intermolecular attracting forces or dense gases the molecules are closer and, therefore, the collision time increases.

The collision of simple atoms is complete elastically. Hence, considering the kinetic for dilute monoatomic gases, the translational energy evolved in a collision is conserved. However, a collision between two molecules might be inelastic or superelastic. While the inelastic concept is used when translational energy is transformed in rotational or vibrational energy, a superelastic impact means that rotational or vibrational energy are converted into translational energy.

6.3.1. Distribution function

The particle distribution function, expressed by $f(x, \xi, t)$, is the main variable in the kinetic theory. The particles system is described by three variables: position, velocity and time, respectively. The "f" function represents the density of particles mass, $\rho = (x, t)$, in three-dimensional physical space, t, and, simultaneously, in three-dimensional velocity space, $\xi = (\xi_x, \xi_y, \xi_z)$, in the microscopic scale.

Even the distribution function is understood from a microscopic point of view, macroscopic variables, among them, density and velocity, can be related with the distribution function, f, by using its moment (see Annex 1).

The discrete-velocity distribution function, $f_i(x,t)$, similar to the distribution function already mentioned, f, represents the density of particles with velocity $c_i = \left(c_{ix}, c_{iy}, c_{iz}\right)$.

6.3.2. Equilibrium distribution function

Once defined and analyzed the distribution function, $f(x,t,\xi)$, the equilibrium distribution function can be explained. The distribution function of a gas can reaches the equilibrium distribution function, $f^{eq}(x,t,\xi)$, after a considerable period of time. This equilibrium distribution functioin allows recovering the time-dependent BSE in the low number limit. [Bernsdorf J., Durst F. and Schäfer M. (1999) Int. J. Numer. Met. Fluids, 29: 251–264.]

To avoid mathematical complexity concepts the analysis of the equilibrium distribution function will be left.

The Equation 21 expresses this equilibrium distribution function. The equilibrium distribution term can also be called “Maxwell-Boltzmann distribution”. [Krüger, T. et al. 2017]

$$f^{eq}(x,|v|,t) = \rho \left(\frac{3}{4\pi e}\right)^{\frac{3}{2}} e^{-3|v|^2/(4e)} = \rho \left(\frac{1}{2\pi RT}\right)^{3/2} e^{-|v|^2/(2RT)} \quad (21)$$

6.3.3. Boltzmann equation and collision operator

Even the Boltzmann equation is founded from a mesoscopic point of view, it is also linked with the macroscopic equations mentioned. Hence, once obtained a solution of the Boltzmann equation it is easy to find it to NSE.

The Boltzmann equation, based on the equilibrium distribution function $f(x,t,\xi)$, describes the evolution in time t of a set of particles, residing in a specific place, which have an associated velocity, ξ. Hence, understanding the distribution function, f, as a function of position x, particle velocity and time t, the derivation with respect to time t will be expressed as seen it in Equation 22.

$$\frac{df}{dt} = \left(\frac{\partial f}{\partial t}\right)\frac{dt}{dt} + \left(\frac{\partial f}{\partial x_\beta}\right)\frac{dx_\beta}{dt} + \left(\frac{\partial f}{\partial \xi_\beta}\right)\frac{d\xi_\beta}{dt} \qquad (22)$$

By converting the right-hand side terms of the resulted expression, Equation 22, in other mathematical terms (see Annex 2), and introducing a new concept to express the total differential it is obtained a new expression, Equation 23. The new term introduced is called the collision operator, $\Omega(\mathrm{f})$, which represents the local redistribution of "f" function due to collisions. This source term is defined as a contribution of three terms, from the distribution function, as expressed in the Equation 23. The first and second terms refer to advection due to particles velocity. The other term represents the force, which influences velocity.

$$\frac{\partial f}{\partial t} + \xi_\beta \frac{\partial f}{\partial x_\beta} + \frac{F_\beta}{\rho}\frac{\partial f}{\partial \xi_\beta} = \Omega(f) \qquad (23)$$

$\Omega(f)$: source term, called "collision operator"

Due to the independence of the collision term with the gradient of the distribution function, f, the scheme for the Boltzmann equation is easy to implement and parallelize.

The collision operators can be approximated with some simplified models, being the discrete BGK (Bhatnagar-Gross-Krook) collision model the most popular one, used in the LBM. The collision operator can be expressed, as seen in Equation 24, once being reduced its mathematical complexity by the approximation mentioned.

$$\Omega(f) = -\frac{1}{\tau}(f - f^{eq}) \qquad (24)$$

τ: relaxation time

This expression captures the relaxation of the distribution function towards the equilibrium distribution, and its velocity is determined by the relaxation time. This new parameter is defined as a time constant that determines the speed to reach the equilibrium. It also determines the transport coefficients such as viscosity and heat diffusivity.

The macroscopic equation of fluid mechanics can be found directly from the Boltzmann equation. The moments of the Boltzmann Equation are multiplied with the functions of velocity, ξ, and integrated over the velocity space. Thus, mass, momentum and energy conservation equations can be found.

Boltzmann equation behaves according to continuity equation. So, the continuity equation, which describes conservation of mass, can be simply obtained from Boltzmann equation. In that case, the BE is integrated over velocity. Likewise, momentum and energy conservation equations can be obtained. Whereas the continuum equation is exact and invariable, both momentum and energy expressions depend on the form of function f. Hence, the difficulty increases. Assuming $f \cong f^{eq}$, the distribution function, f, is commonly approximated by using the called “Chapman-Enskog analysis”.

The Chapman-Enskog analysis links the kinetic and continuum picture through the non-equilibrium contributions of the distribution function. In other words, LBE and NSE can be associated using this new concept and LBE is assumed to be macroscopic according to NSE. Thus, the lattice Boltzmann equation can be used to simulate the Navier-Stokes equations.

Boltzmann equation has some advantages compared with conventional methods like finite volume or finite element. The most important to be mentioned is the exact advection that Boltzmann equation guarantees when discretizing the advection term, $(u \cdot \nabla)u$. Consequently, and contrarily to conventional methods, the complexity of iterative numerical processes is avoided and, also are errors involved.

6.3.4. Boltzmann $\mathcal{H}$-theorem

Boltzmann $\mathcal{H}$-theorem, defined in Equation 25, evidences that molecular collisions invariably lead to a distribution function towards equilibrium. This state will be corroborated in this section.

A new concept is needed to be introduced when explaining the Boltzmann $\mathcal{H}$ -theorem, called the entropy density, ρs, and units $\left[\frac{J}{kg \cdot m^3}\right]$.

The Equation 26 expresses the Boltzmann $\mathcal{H}$-theorem in mathematical terms.

$$\mathcal{H} = \int f \; lnf \; d^3\xi \qquad (25)$$

Some considerations to bear in mind, resulted from the expression, Equation 25:

- This quantity can only decrease
- When the distribution function f reaches equilibrium the Boltzmann $\mathcal{H}$ -theorem gets the minimum value

By reformulating this expression and applying some rules, a new equation is obtained. The balanced equation for the quantity $\mathcal{H}$ expressed as a moment of the Boltzmann equation is shown in Equation 26.

$$\frac{\partial}{\partial t}\int f \; lnf \; d^3\xi + \frac{\partial}{\partial x_\alpha}\int f \; lnf \; d^3\xi = \int lnf\Omega(f) \; d^3\xi \qquad (26)$$

Considering some approximations, the last step leads to Equation 27.

$$\frac{\partial \mathcal{H}}{\partial t} + \frac{\partial \mathcal{H}_\alpha}{\partial x_\alpha} \leq 0 \qquad (27)$$

The main comment to keep in mind for Boltzmann $\mathcal{H}$-theorem is the fact that $\mathcal{H}$ is not conserved in the system, which never increases, instead it decreases, while the particle distribution doesn’t reach the equilibrium. This statement is the basic meaning of Boltzmann $\mathcal{H}$-theorem. The same observation leads to conclude that entropy increase in a system until the distribution function reaches the equilibrium, characterized by an entropy minimum. Hence, for ideal gases, $\mathcal{H}$ is proportional to the entropy density, as seen in Equation 28.

$$\rho s = -R \; \mathcal{H} \quad (28)$$

6.3.5. Law of similarity: Knudsen (Kn), Reynolds number (Re), Mach number (Ma)

Fluids flows which share dimensionless numbers provide the same physics upon a simple scaling by the typical length and velocity scales in the problem. [Krüger, T. et al. 2017].

High Reynolds number flows are dominated by turbulence and are characteristic for vehicle aerodynamics and building design, among other applications. However, low Reynold numbers are also attractive in microfluidics and biophysics fields.

6.4. ADVANTAGES AND DISADVANTAGES OF LBM

Each method mentioned, implemented for finding fluid flow solution, has its own advantages and disadvantages.

Among all the advantages of Lattice Boltzmann the ones to be highlight and the responsible for gaining researchers' attention are mentioned below: [Krüger, T. et al. 2017], [Mohamad, A. Lattice Boltzmann Method: Fundamentals and Engineering Applications with Computer Codes. Second Edition. Springer. 2019.], [Guo, Z., C. Shu. Lattice Boltzmann Method and its Applications in Engineering. Advances in Computational Fluid Dynamics, Volume 3. World Scientific.2013].

- Simplicity of the method application
- Its extensibility, scalability and performance on parallel computers
- The ease with which it can handle complex geometries.
- Local character of the LB algorithm allowing higher parallelization.
- For incompressible fluids, the Navier-Stokes equation, which resolution with CFD methods was complex, it's able to be solved using LBM in a simple way by allowing artificial compressibility.
- The Poisson equation, involved in CFD methods, does not appear when simulating fluid using LBM. This is an advantage for LBM because the non-locality of this method could have result in a mathematical complex resolution.

Due to the disadvantages LBM presented in some fluid conditions, alternative methods are required for solving fluid problems, and so, the partial difference equation (PDE) involved.

A numerical solution, understanding it as an approximation of the exact solution of continuum equations, leads to truncational errors, inevitably involved, which are proportional to some parameters, as time and spatial steps. Thus, order of accuracy can determine if a method is precise enough to be implemented.

LBM, even being considered an attractive method to be implemented in a wide fluid scene is limited in some circumstances. The main limitation in LBM which will be highlighted is the statical noise involved in final simulation results.

Another important disadvantage that involves LBM is the amount of memory required for simulating. When dealing with steady flows other methods, different from the lattice Boltzmann one should be contemplated.

6.5. FIELD APPLICATIONS OF LBM

The LBM is a powerful approach to hydrodynamics, with applications ranging from high Re number flows at a microscopic scale, porus media and multi-phase flows. [Körner, C., Pohl, T., Rüde, U., Thürey, N., & Zeiser, T. Parallel Lattice Boltzmann Methods for CFD Applications. 2006.]. LBM is also a befitting technique to simulate the behaviour of fluids from micro scale with high number of Reynolds. It may be implemented for simulating situations in which sound and flow interact. For example, aeroacoustics sound generation.

Moreover, LBM is well suited for simulating mass-conserving flows in schemes with complex geometries like automobile aerodynamics [Bernsdorf J., Durst F. and Schäfer M. (1999) Int. J. Numer. Met. Fluids, 29: 251–264.]

Even the wide range of LBM applications, this method does not fit in all fields of engineering. Likewise, LBM is not appropriated for directly simulating long-range propagation of sound at realistic viscosities. Thus, other suitable models should be considered.

Due to the rapidly development of the technique, LBM lacks from a systematic and flexible construction of numerical schemes in some situations such as mixture models with a given Schmidt number, or thermal models with a given Prandtl number. A new recently LB method called Quasi-equilibrium kinetic model has been introduced for dealing with this problem. [Gorban, A. Quasi-Equilibrium Lattice Boltzmann Method. The European Physical Journal B. 2007.]

6.6. LIMITATIONS OF LBM

The Knudsen number is defined as the ratio between the mean path and the representative physical length scale. It is the expansion parameter used in the Chapman-Enskog theory to derive the NSE from the Boltzmann equation. Moreover, the Knudsen number, is related to the Mach number and the Reynolds number, as expressed in Equation 29, commonly known as “von Kármán relation”.

$$Kn = \alpha \frac{Ma}{Re} \qquad (29)$$

$$\alpha: numerical\ constant$$

Lattice Boltzmann simulations are generally set-up in the limit of small Knudsen and Mach number. For small values of Kn, $Kn < 1$, a hydrodynamic system (Navier-Stokes) is valid. However, LBM is implemented for $Kn \geq 1$. [Krüger, T. et al. 2017].

Although the large advantages of LB that converts it in a wide method for dealing with several fluid flow situations it is limited in some geometry’s schemes. Mixture models with a given Schmidt number and thermal models with a given Prandtl number are examples of fluid conditions in which LBM generate problems. [Gorban, A. Quasi-Equilibrium Lattice Boltzmann Method. The European Physical Journal B. 2007.].

LBM is a weakly compressible Navier-Stokes solver. Thus, when simulating strongly compressible flows, for example in the case of supersonic fluids, alternative methods to LBM should be used. [Krüger, T. et al. 2017].

For the isothermal binary mixtures and for a weakly compressible flow a new method based on Lattice Boltzmann called “Quasi-Equilibrium Lattice Boltzmann Method”, (QELBM), has been applied. Quasi-Equilibrium (QE) kinetic models are considered a new approach to cover LBM limitations in hydrodynamic systems. [Gorban, A. Quasi-Equilibrium Lattice Boltzmann Method. The European Physical Journal B. 2007.].

As other kinetic models mentioned, the QE is founded on the constructions of continuous time-space QE models. Once the scheme is discretised the next step is the derivation of the system.

Other kinetic models are proposed to deal with LBM limitations like bulk viscosity issues and chemical reactions. [Ansumali, S., K. Iliya V., Consistent Lattice Boltzmann Method. ETH-Zürich, Institute of Energy Technology, CH-8092, Zürich, Switzerland. Physical review letters.2008.].

7. LATTICE BOLTZMANN IN A NEAR FUTURE

The lattice Boltzmann method is still undergoing development. Many models, including simulation of granular flows, viscoelastic flows, magnetohydrodynamics, and microemulsions were recently proposed. Although innovative and promising, these existent LBM methods, including multicomponent LBM models, require additional benchmarking and verification. The current lattice models for multiphase and reacting systems are most suitable for isothermal problems. The development of a reliable LBM for thermal systems will allow the simulation of heat transfer and surface phenomena simultaneously. This would open many new areas of application in a near future. [Gorban, A. Quasi-Equilibrium Lattice Boltzmann Method. The European Physical Journal B. 2007.].

Recently, a novel model with energy conservation, derived from LBM has been research and implemented for recovering LBM limitations. The problem of energy conservation can be solved with this new approach. This new model extends the LBM to a hydrodynamically consistent simulation tool for nearly incompressible flows. [Ansumali, S., K. Iliya V., Consistent Lattice Boltzmann Method. ETH-Zürich, Institute of Energy Technology, CH-8092, Zürich, Switzerland. Physical review letters.2008.].

In recent years, the lattice Boltzmann method has drawn considerable attention as a simulation method for flows at low Mach numbers. Especially popular are the called “isothermal lattice Boltzmann models” (ILBM) without energy conservation. The presence of bulk viscosity precludes the limit of incompressible hydrodynamics. Even the efficiency ILBM method in incompressible hydrodynamics, the spurious bulk viscosity of ILBM becomes a drawback when such models are applied to a weakly compressible or microscopic flow simulations. [Ansumali, S., K. Iliya V., Consistent Lattice Boltzmann Method. ETH-Zürich, Institute of Energy Technology, CH-8092, Zürich, Switzerland. Physical review letters.2008.

8. LATTICE BOLTZMANN SIMULATION WITH WOLFRAM MATHEMATICA. COMPUTATIONAL WIND TUNNEL. FLUID FLOW SIMULATION

8.1. BEGINNINGS

Wind tunnels are devices used to study the effect of an object as it moves through a fluid.

The most common and simple wind tunnel are the open-return ones. The design consists of a hollow pipe or rectangular box in which one end of the tunnel is fitted with a fan, whilst the other is open.

In this section it's going to be studied the fluid dynamics by developing a basic schematic of a 2D wind tunnel, in which the tops and bottom's walls are solid, and the fluid enter the tunnel from the left and travel through the tunnel to the right (see Figure 4).

Even Navier Stokes equations (NSE) are the main governing equations that dictate the behaviour of fluid flows, the Lattice Boltzmann method (LBM) can study the fluid dynamics without solving them. Thus, the massive system of nonlinear algebraic equations due to the partial differential equations (PDE) is avoid, and so, its mathematical complexity. Moreover, the local operation from LBM is another reason to consider this tool as a highly parallel technic to simulate fluid flow.

The simulations are performed in a microscopic, or "lattice", domain in which particles interacts with each other. Even this microscopic point of view, it is possible to describe macroscopic variables by focusing on the evolution of the particle's behaviour over the time.

8.2. **FLUID BEHAVIOUR. CASE STUDY. EXAMPLES**

In the following section it is illustrated a fluid in different scenes with the aim to understand the computational wind tunnel setup and its flexibility.

Firstly, it is going to be exemplified the simplest case, to make sure the fully understanding of the fluid flow simulation process.

8.2.1. Flow in the wind tunnel

The schematic of the domain consists of two walls with no velocity (see Figure 4).

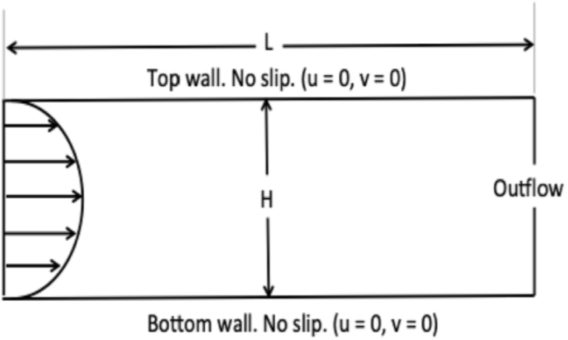
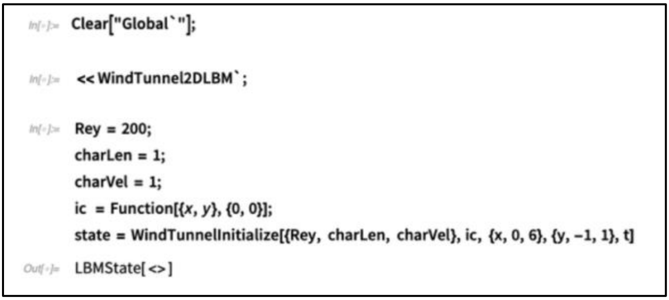

L
Top wall. No slip. (u = 0, v = 0)
H
Outflow
Bottom wall. No slip. (u = 0, v = 0)

Figure 4. *Scheme representation of the domain with a flow in a wind tunnel with boundary conditions. Fixed walls.*

The only length scale to the problem and the characteristic one is the height of the wind tunnel. The characteristic velocity refers to the maximum one, which comes from the inlet, and it will be assigned a value of 1. All that remains is to specify the Reynolds number at which the simulation is to be carried out. For a wind tunnel's length of 6 units (from 0 to 6) and a height of 2 units (from -1 to 1).

The simulation is set-up by using the corresponding code:



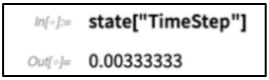
```
In[•]:= Clear["Global`"];

In[•]:= <<WindTunnel2DLBM`;

In[•]:= Rey = 200;
        charLen = 1;
        charVel = 1;
        ic = Function[{x, y}, {0, 0}];
        state = WindTunnelInitialize[{Rey, charLen, charVel}, ic, {x, 0, 6}, {y, -1, 1}, t]
Out[•]= LBMState[<>]
```

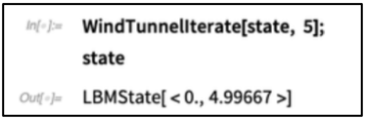
Up to now, it is not imposed any boundary condition information yet because the wind tunnel defaults to the flow in a channel case and, thus, all the boundary conditions are automatically imposed. Therefore, the specified parameters are only the characteristic information and the dimensions of the wind tunnel.

By using a fixed time step the simulation is performed:



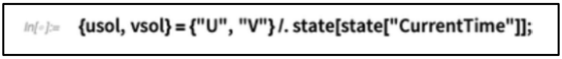
```
In[•]:= state["TimeStep"]
Out[•]= 0.00333333
```

However, for a period of 5 time units, the running simulation is:



```
In[•]:= WindTunnelIterate[state, 5];
        state
Out[•]= LBMState[< 0., 4.99667 >]
```

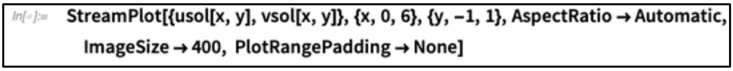
Once the simulation is finished it is possible to get the relevant results. This instruction is shown:



```
In[•]:= {usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
```

One way to extract that data is by visualizing the streamline plot (see Figure 5)

The code for these instructions is:



```
In[◦]:= StreamPlot[{usol[x, y], vsol[x, y]}, {x, 0, 6}, {y, −1, 1}, AspectRatio → Automatic,
          ImageSize → 400, PlotRangePadding → None]
```

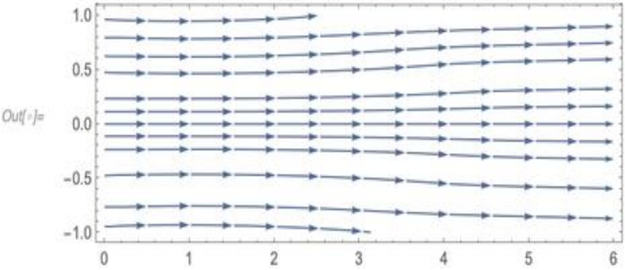
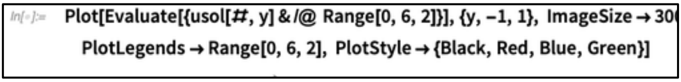


Figure 5. *Representation of the behaviour of the fluid for the simplest case of the wind flow simulation (flow in a wind tunnel). The blue arrows give information of the lengths and direction of the simulated fluid.*

The streamlines present some deviations. This phenomenon is explained by the velocity field (see Figure 6), in which are represented the profiles of the "u" component of the velocity field at various x locations.

The code implemented is:



```
In[◦]:= Plot[Evaluate[{usol[#, y] &/@ Range[0, 6, 2]}], {y, −1, 1}, ImageSize → 30
          PlotLegends → Range[0, 6, 2], PlotStyle → {Black, Red, Blue, Green}]
```

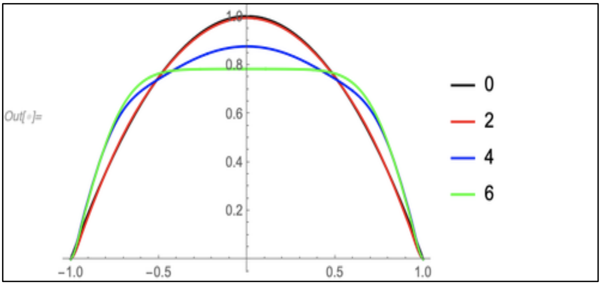
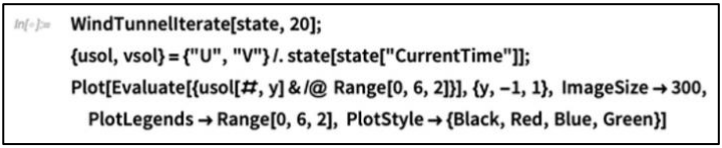


Figure 6. *Velocity profiles. Representation of fluid velocities for different positions of the fluid inside the wind tunnel. It is tendency to a steady state. Simulation for a 5 periods of time.*

As it is shown, the velocities have a spatial dependence. Thus, the flow will reach steady state. This fact can be verified by comparing the velocity profiles obtained from running the simulation for different time units (the simulation will be run for 5 and 20 time units).

For a period of 20 time:



```
In[•]:= WindTunnelIterate[state, 20];
        {usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
        Plot[Evaluate[{usol[#, y] & /@ Range[0, 6, 2]}], {y, -1, 1}, ImageSize → 300,
          PlotLegends → Range[0, 6, 2], PlotStyle → {Black, Red, Blue, Green}]
```

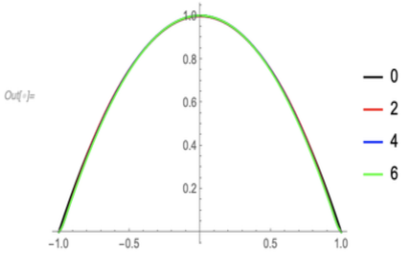


Figure 7. *Representation of the velocity profiles for different positions of the fluid inside the wind tunnel. Simulation for a 20 periods of time. Steady state.*

As it can be deduced (see Figure 7), the velocity profiles at the various x locations are almost the same with each other. Thus, this analysis reveals the idea of reaching steady state.

8.2.2. Flow in a box problem (see annex 3)

8.2.3. Observing disturbance caused by a moving object (see annex 4)

8.2.4. Flow in a pipe with bends and obstacles

What kind of flow pattern would be expected for the following geometry? (see Figure 20).

Once occurred the simulation the question will be answered.

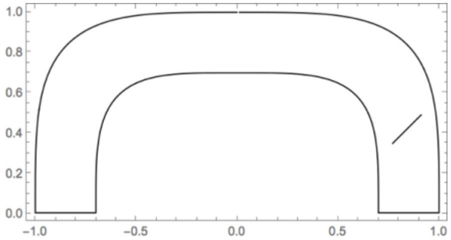
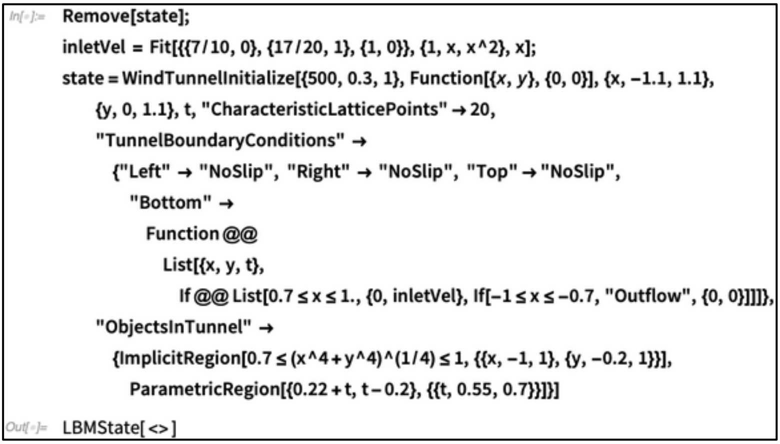


Figure 20. *Scheme representation of a pipe with bends and obstacles.*

Fluid enters the pipe from the right end, moves up the pipe and then gets discharged from the left end. What would be the effect of that stopper at the right end? How will it impact the discharge?

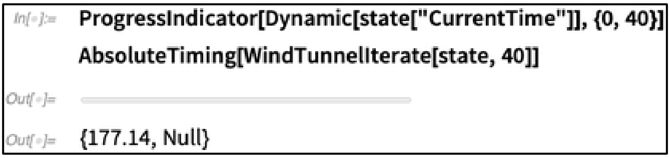
The best way to figure out is by setting up the problem. Only the pipe will be immersed and the obstacle within it into the wind tunnel. The left, right and top boundaries of the device are given a 0-velocity condition. The bottom boundary is given an outflow condition from –1<= x <=-0.7, a 0-velocitycondition from –0.7<= x <=0.7 and a parabolic velocity profile from 0.7<=x<= 1.

The code implemented is:



```
In[•]:= Remove[state];
        inletVel = Fit[{{7/10, 0}, {17/20, 1}, {1, 0}}, {1, x, x^2}, x];
        state = WindTunnelInitialize[{500, 0.3, 1}, Function[{x, y}, {0, 0}], {x, −1.1, 1.1},
            {y, 0, 1.1}, t, "CharacteristicLatticePoints" → 20,
            "TunnelBoundaryConditions" →
             {"Left" → "NoSlip", "Right" → "NoSlip", "Top" → "NoSlip",
              "Bottom" →
               Function @@
                List[{x, y, t},
                 If @@ List[0.7 ≤ x ≤ 1., {0, inletVel}, If[−1 ≤ x ≤ −0.7, "Outflow", {0, 0}]]]},
            "ObjectsInTunnel" →
             {ImplicitRegion[0.7 ≤ (x^4 + y^4)^(1/4) ≤ 1, {{x, −1, 1}, {y, −0.2, 1}}],
              ParametricRegion[{0.22 + t, t − 0.2}, {{t, 0.55, 0.7}}]}]
Out[•]= LBMState[<>]
```

If the simulation runs for 40 time units:



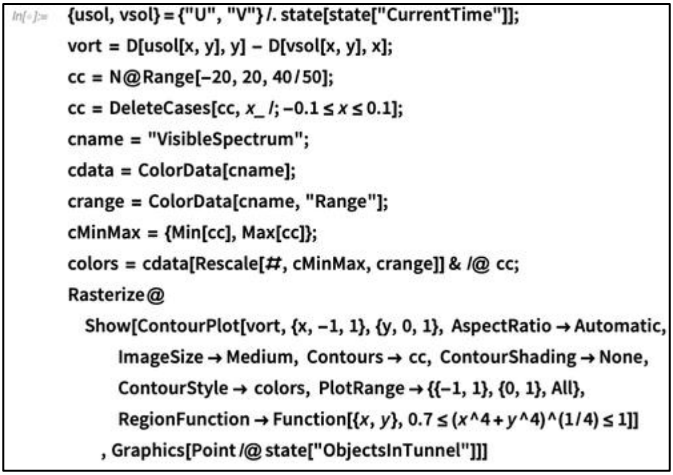
```
In[◦]:= ProgressIndicator[Dynamic[state["CurrentTime"]], {0, 40}]
        AbsoluteTiming[WindTunnelIterate[state, 40]]

Out[◦]=

Out[◦]= {177.14, Null}
```

Taking vorticity into account, the derived plot is represented below (see Figure 21).

The code for running the simulation is:



```
In[◦]:= {usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
        vort = D[usol[x, y], y] − D[vsol[x, y], x];
        cc = N@Range[−20, 20, 40 / 50];
        cc = DeleteCases[cc, x_ /; −0.1 ≤ x ≤ 0.1];
        cname = "VisibleSpectrum";
        cdata = ColorData[cname];
        crange = ColorData[cname, "Range"];
        cMinMax = {Min[cc], Max[cc]};
        colors = cdata[Rescale[#, cMinMax, crange]] & /@ cc;
        Rasterize@
          Show[ContourPlot[vort, {x, −1, 1}, {y, 0, 1}, AspectRatio → Automatic,
              ImageSize → Medium, Contours → cc, ContourShading → None,
              ContourStyle → colors, PlotRange → {{−1, 1}, {0, 1}, All},
              RegionFunction → Function[{x, y}, 0.7 ≤ (x^4 + y^4)^(1/4) ≤ 1]]
           , Graphics[Point /@ state["ObjectsInTunnel"]]]
```

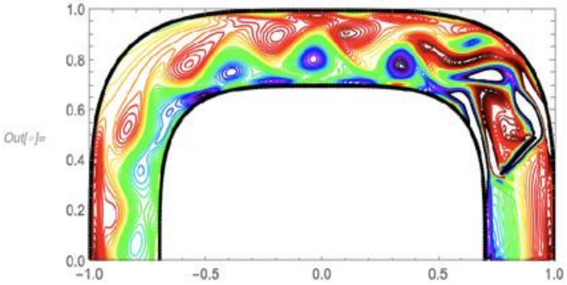
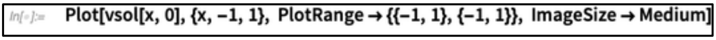


Figure 21. *Representation of the fluid behaviour. Result of the simulation taking vorticity into account. Vortex shedding caused by the object. Flow in a pipe with bands and obstacles.*

The velocities at y=0 is represented as follows (see Figure 22):



```
In[ ]:=  Plot[vsol[x, 0], {x, -1, 1}, PlotRange → {{-1, 1}, {-1, 1}}, ImageSize → Medium]
```

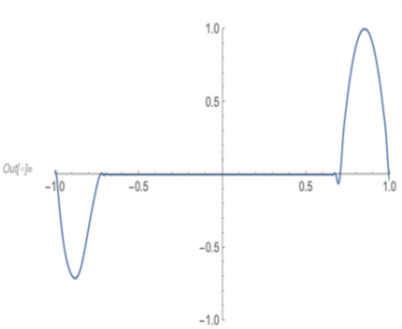


Figure 22. *Velocities profiles at y=0. Comparison among the outlet and inlet velocities profiles. The outlet velocity is half of the inlet velocity.*

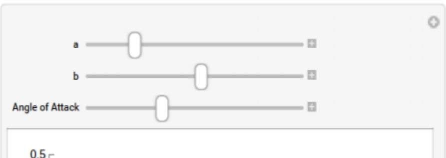
Comparing the outlet and inlet velocities profiles it is observed that the value of the outlet velocity is half of the inlet one. This fact evinces the disturbance the object causes to the fluid, resulting in a reduction in fluid discharge from the left end of the pipe.

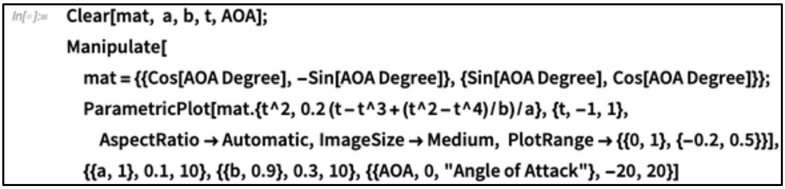
8.2.5. Simulating the flow over an airfoil

In this case, the flow is simulated around an airfoil (see Figure 23). The mathematical expression, Equation 30, defines this kind of the airfoil (see Figure 23).

$$\left\{t^2, 0.2\left(t - t^3 + \frac{t^2-t^4}{b}\right)/a\right\}, -1 \le t \le 1 \quad (30)$$

where parameter "a" controls how thick an airfoil should be, and parameter "b" controls the curvature of the airfoil





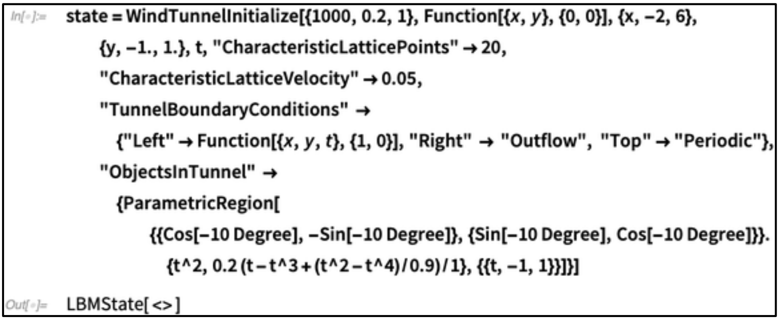
```
In[ ]:= Clear[mat, a, b, t, AOA];
        Manipulate[
          mat = {{Cos[AOA Degree], −Sin[AOA Degree]}, {Sin[AOA Degree], Cos[AOA Degree]}};
          ParametricPlot[mat.{t^2, 0.2 (t − t^3 + (t^2 − t^4) / b) / a}, {t, −1, 1},
            AspectRatio → Automatic, ImageSize → Medium, PlotRange → {{0, 1}, {−0.2, 0.5}}],
          {{a, 1}, 0.1, 10}, {{b, 0.9}, 0.3, 10}, {{AOA, 0, "Angle of Attack"}, −20, 20}]
```

For the aircraft to being able to get off the ground, the top surface of the airfoil should have a pressure distribution, lower than the bottom surface. This pressure difference causes the wing to lift upward (along with anything attached to it). This pressure difference is achieved by having wind blow over its surface at significantly high speeds.

A second consideration is that the wing generally needs to be tilted or have an “angle of attack” to it. By doing this, we ensure greater lift. We will also give the airfoil a value of -10 for the angle of attack.

Even a typical Reynolds number for small aircraft is around 1 million, the simulation will be run for a Reynolds number of 1000 just because a full-scale simulation is not possible on a laptop due to the large grid size.

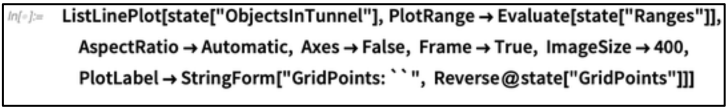
For this example, a uniform flow fill comes in from the left. The top and bottom tunnel boundaries are set to be periodic, and the right boundary is set to an outflow. The characteristic length here will be the thickness of the airfoil:



```
In[•]:=  state = WindTunnelInitialize[{1000, 0.2, 1}, Function[{x, y}, {0, 0}], {x, -2, 6},
           {y, -1., 1.}, t, "CharacteristicLatticePoints" → 20,
           "CharacteristicLatticeVelocity" → 0.05,
           "TunnelBoundaryConditions" →
             {"Left" → Function[{x, y, t}, {1, 0}], "Right" → "Outflow", "Top" → "Periodic"},
           "ObjectsInTunnel" →
             {ParametricRegion[
                 {{Cos[-10 Degree], -Sin[-10 Degree]}, {Sin[-10 Degree], Cos[-10 Degree]}}.
                   {t^2, 0.2 (t - t^3 + (t^2 - t^4) / 0.9) / 1}, {{t, -1, 1}}]}]

Out[•]=  LBMState[<>]
```

Before simulating, following the same pattern as other examples fluid flow, the discretized object is extracted (see Figure 24) to check the appropriate location of the object within the wind tunnel.



```
In[•]:=  ListLinePlot[state["ObjectsInTunnel"], PlotRange → Evaluate[state["Ranges"]],
           AspectRatio → Automatic, Axes → False, Frame → True, ImageSize → 400,
           PlotLabel → StringForm["GridPoints: ``", Reverse@state["GridPoints"]]]
```

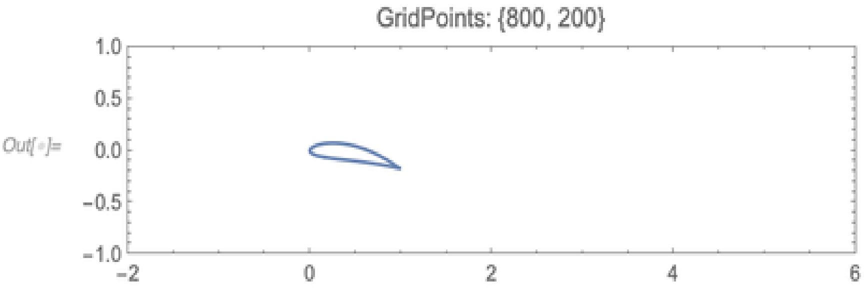
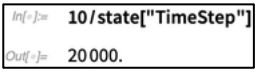


Figure 24. *Representation of the position of the object. Simulation of a flow over an airfoil. Large number of points due to the thin airfoil. It implies bigger time resolution.*

As it can be observed in the grid (see Figure 24), the number of points is large, because the thin airfoil is being solved by 20 lattice points.

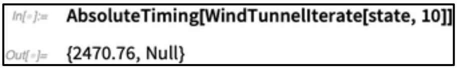
Running the simulation for 10 time units:



```
In[◦]:=  10/state["TimeStep"]
Out[◦]=  20000.
```

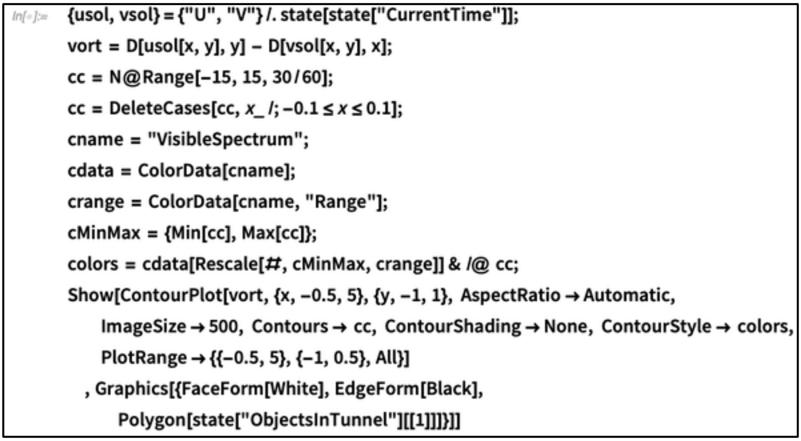
The resolution time is bigger compared with other simulations because the number of grid points (800x200) needed for running the simulation is quite large, and so it is the number of iterations (20000), to set up the problem.

By iterating:



```
In[◦]:=  AbsoluteTiming[WindTunnelIterate[state, 10]]
Out[◦]=  {2470.76, Null}
```

Taking vorticity into account (see Figure 25), the code for the simulation is:



```
In[◦]:=  {usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
         vort = D[usol[x, y], y] – D[vsol[x, y], x];
         cc = N@Range[–15, 15, 30/60];
         cc = DeleteCases[cc, x_ /; –0.1 ≤ x ≤ 0.1];
         cname = "VisibleSpectrum";
         cdata = ColorData[cname];
         crange = ColorData[cname, "Range"];
         cMinMax = {Min[cc], Max[cc]};
         colors = cdata[Rescale[#, cMinMax, crange]] & /@ cc;
         Show[ContourPlot[vort, {x, –0.5, 5}, {y, –1, 1}, AspectRatio → Automatic,
             ImageSize → 500, Contours → cc, ContourShading → None, ContourStyle → colors,
             PlotRange → {{–0.5, 5}, {–1, 0.5}, All}]
           , Graphics[{FaceForm[White], EdgeForm[Black],
               Polygon[state["ObjectsInTunnel"][[1]]]}]]
```

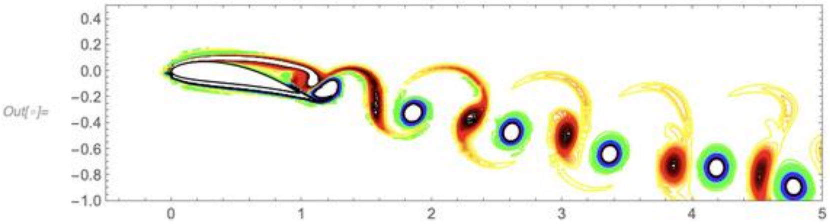


Figure 25. *Representation of the fluid behaviour taking vorticity into account. Flow over an airfoil. Vortex shedding generated. Aircraft getting off the ground.*

As seen in other examples, this situation generates a vortex shedding (see Figure 25). Nevertheless, in contrast to previous fluid scenes, the aim for the airfoil case is to flow to hug the surface.

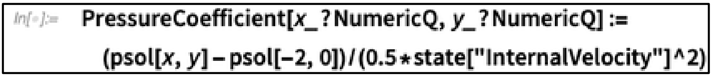
When the flow separates (as seen on the top surface of the airfoil), the pressure drop is not achieved properly, and the airfoil will be unable to achieve the lift.

Next, the pressure at the object's surface will be analyst.

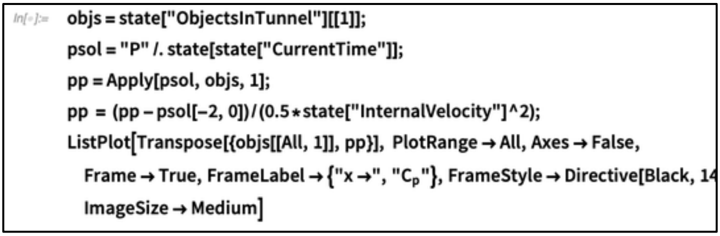
It will be plotted a non-dimensional parameter (see Figure 26), called pressure coefficient, defined below, Equation 31.

$$C_p = 2(p - p_\infty)/(\rho_{LBM} U_{LBM}^2) \qquad (31)$$

where p_∞ is the pressure far upstream.



```
In[=]:= PressureCoefficient[x_?NumericQ, y_?NumericQ] :=
          (psol[x, y] - psol[-2, 0]) / (0.5 * state["InternalVelocity"]^2)
```



```
In[∘]:= objs = state["ObjectsInTunnel"][[1]];
        psol = "P" /. state[state["CurrentTime"]];
        pp = Apply[psol, objs, 1];
        pp = (pp - psol[-2, 0]) / (0.5 * state["InternalVelocity"] ^ 2);
        ListPlot[Transpose[{objs[[All, 1]], pp}], PlotRange → All, Axes → False,
          Frame → True, FrameLabel → {"x →", "Cp"}, FrameStyle → Directive[Black, 14
          ImageSize → Medium]
```

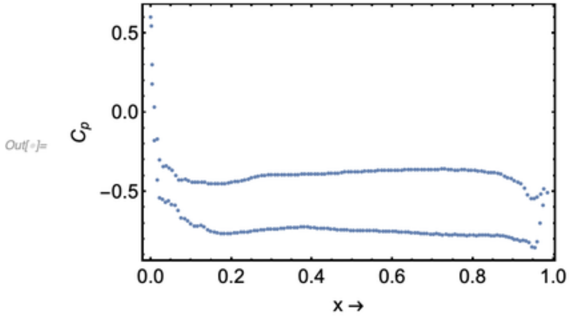
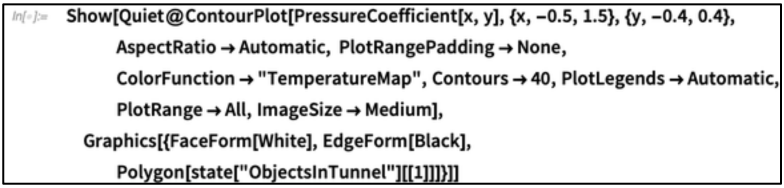


Figure 26. *Representation of the pressure near the airfoil. There are observed pressure differences from the top and the bottom of the surface. Pressure coefficient plotted.*

The resulting lower line represents the pressure on the top surface, while the top line represents the pressure on the bottom surface. It shows a clearly evidence of some pressure differences.

The pressure contours are plotted (see Figure 27). The simulation code is:



```
In[∘]:= Show[Quiet@ContourPlot[PressureCoefficient[x, y], {x, -0.5, 1.5}, {y, -0.4, 0.4},
            AspectRatio → Automatic, PlotRangePadding → None,
            ColorFunction → "TemperatureMap", Contours → 40, PlotLegends → Automatic,
            PlotRange → All, ImageSize → Medium],
          Graphics[{FaceForm[White], EdgeForm[Black],
            Polygon[state["ObjectsInTunnel"][[1]]]}]]
```

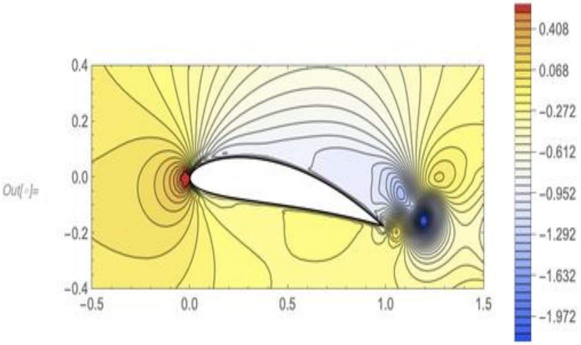


Figure 27. *Plot of the pressure contours from the airfoil. The resulting simulation shows a pressure difference between top and bottom surface of the airfoil. “Bermoulli principle”.*

By plotting the pressure contours (see Figure 27) and visualizing them near the airfoil it can be demonstrated that the top surface pressure is less than the bottom surface. Thus, this airfoil works well.

The fluid dynamic property explored is known as “Bermoulli principle” and has applications in aviation and in fields such as automotive engineering.

9. LIST OF SYMBOLS

c_i: particles velocity. Lattice gas models

f: distribution function

f_i^{eq}: equilibrium distribution function

F: external body forces

n_i^*: post-collision occupational number

p: pressure

p_∞: pressure far upstream

R: universal gas constant

t: time

T: temperature

T_o: initial temperature

u^*: velocity term once collision term computed and boundary-adjustment step

V_o: stationary volume

α: numerical constant

$\delta_{LBM} = 1$: time step in the Lattice Bolztmann. BGK approximation

Δ: Laplace operator

ζ : viscosity coefficient

η: viscosity coefficient

λ: solution of the variables. Finite difference method

ξ : particle velocity

Π: momentum flux density tensor

ρ: density

ρ^*: density term once collision term computed and boundary-adjustment step

ρs: entropy density

ρu: mass flux density or momentum density

$\sigma_{\alpha\beta}$: stress tensor for isotropic fluid

$\sigma'_{\alpha\beta}$: general viscous stress tensor

τ: relaxation time

ν_{LBM}: cinematic viscosity

Ω: collision operator

10. CONCLUSION

Interest in the lattice Boltzmann method has become a vast research field in the last 25 years and it has steadily increase, since it grew out of lattice gas models in the late 1980s. These methods are applied with the aim to simulate the flow of liquids and gases through the behaviour of a gas.

The aim of CFD methods is to express the fluid behaviour of complex geometries schemes, by equations, which cannot be solved without a computer.

The main limitation of CFD comes from the geometry's discretization of the fluid, where mesh generation consume an excessive amount of time. For unsteady-inviscid flow and steady-state inviscid supersonic CFD is difficult to implement. It also implies problems in equilibrium terms, related to boundary condition.

The lattice Boltzmann method, based on the evolution of statical distribution on lattices, is an alternative approach to Navier-Stokes equations for computational fluid dynamics (CFD). Kinetic theory, in which LBM is founded in the mesoscopic scale description of the behaviour of a fluid, using mathematical expressions.

The main advantages of LBM are the simplicity of the method, its scalability on parallel computers and the ease with which it can deal with complex geometries

Many processes and plants considered in Chemical Engineering filed involve transport phenomena with fluid flow.

LBM is well suited to simulate mass-conserving flows in schemes with complex geometries. Among them, LBM handle situations with efficient results in porus media and multi-phase flows. LBM is also a befitting technique to simulate the behaviour of fluids from micro scale with high number of Reynolds.

Although the large advantages of LB that converts it in a wide method for dealing with several fluid flow situations it is limited in some geometry's schemes. Mixture models with a given Schmidt number and thermal models with a given Prandtl number are examples of fluid conditions in which LBM generate problems. As an alternative method to cover this limitation a new method called "Quasi-Equilibrium Lattice Boltzmann Method", (QELBM) has been introduced. Apart from QELBM, other methods for dealing with LBM limitations are explored constantly to improve the quality of the simulations.

The law of similarity is, for instance, regularly used in the automotive or aircraft industries where models of cars or planes are tested in wind tunnels. Since the models are usually smaller than the real objects, the flow velocity should increase or decrease the viscosity in such a way that the Reynolds numbers in both systems match. For example, if the model of a car is five times smaller than the car itself, the velocity of the air in the wind tunnel should be five times larger than the real one.

REFERENCES AND NOTES

1. Krüger, T., H.Kusumaatmaja, A. Kuzmin, O. Shardt, G.Silva, E.Magnus Viggen. The Lattice Boltzmann Method. Principles and Practice. Springer. 2017.
2. Mohamad, A. Lattice Boltzmann Method: Fundamentals and Engineering Applications with Computer Codes. Second Edition. Springer. 2019.
3. Guo, Z., C. Shu. Lattice Boltzmann Method and its Applications in Engineering. Advances in Computational Fluid Dynamics, Volume 3. World Scientific.2013.
4. Huang, H., Michael Sukop and Xi-Yun Lu. Multiphase Lattice Boltzmann Methods: Theory and Application. Wiley Blackwell. 2015.
5. Gorban, A. Quasi-Equilibrium Lattice Boltzmann Method. The European Physical Journal B. 2007.
6. Körner, C., Pohl, T., Rüde, U., Thürey, N., & Zeiser, T. Parallel Lattice Boltzmann Methods for CFD Applications. 2006.
7. Bernsdorf J., Durst F. and Schäfer M. (1999) Int. J. Numer. Met. Fluids, 29: 251–264.
8. Qian, Y.H., d'Humie`res D. and Lallemand P. (1992) Europhys. Lett., 17(6): 479–484.
9. Bernsdorf J., Zeiser Th., Brenner G. and Durst F. (1998) Int. J. Mod. Phys. C, 9(8): 1129–1141.
10. Bernsdorf J., Günnewig O., Hamm W. and Mu¨nker, O. (1999) GIT Labor-Fachzeitschrift 4/99: 387–390.
11. Ansumali, S., K. Iliya V., Consistent Lattice Boltzmann Method. ETH-Zürich, Institute of Energy Technology, CH-8092, Zürich, Switzerland. Physical review letters.2008.
12. Mokhasi,P. Kernel Developer, Algorithms R&D. Building a Lattice Boltzmann\[Dash]Based Wind Tunnel with the Wolfram Language. Nov.2. 2019.

ACRONYMS

BGK: Bhatnagar-Gross-Krook

CFD: Computational Fluid Dynamics

DSMC: Direct simulation Monte Carlo

FDM: Finite Difference Method

FEM: Finite Element Method

FVM: Finite Volume Method

ILBM: Isothermal Lattice Boltzmann Method

LBM: Lattice Boltzmann Method

LBE: Lattice Boltzmann Equation

LGA: Lattice Gas Automata

MD: Molecular Dynamics

NS(E): Navier Stokes (Equation)

PDE: Partial Differential Equation

QE: Quasi-Equilibrium

QELBM: Quasi-Equilibrium Lattice Boltzmann Method

APPENDICES

APPENDIX 1: MOMENTUM OF THE DISTRIBUTION FUNCTION

In a few words it’s going to be explained the mathematical process to obtain the momentum density as an example. Taking into the procedure will be shown the mathematical expression of the total energy density.

The moment mathematical concept is expressed as an integral of the distribution function, weighted with some function of ξ. Integrating over velocity space, the macroscopic mass density can be defined as shown in the Equation 32:

$$\rho(x,t) = \int f(x,\xi,t)d^3\xi \qquad (32)$$

The Equation 33 shows a similar case for momentum density, also defined as the moment.

$$\rho(x,t)\ u(x,t) = \int \xi f(x,\xi,t)d^3\xi \qquad (33)$$

Once applied some mathematical concepts, the total energy density can be redefined as shown in Equation 34.

$$\rho(x,t)\ E(x,t) = \frac{1}{2}\int |\xi|^2 f(x,\xi,t)d^3\xi \qquad (34)$$

For monoatomic gases, the Equation 34 refers, only, to translational energy of molecules. Due to the inexistent inner molecular structure in the case of monoatomic gases, molecular vibrational and rotational energies are null.

APPENDIX 2: BOLTZMANN EQUATION TRANSFORMATION

The expressions represented by the Equation 36, 37 and 38 are the mathematical approximations implemented that leads to Equation 23.

$$\frac{\mathrm{dt}}{dt} = 1 \qquad (36)$$

$$\frac{dx_\beta}{dt} = \xi_\beta \qquad (37)$$

$$\frac{d\xi_\beta}{dt} = \frac{F_\beta}{\rho} \qquad (38)$$

APPENDIX 3: FLUID IN A BOX PROBLEM

Before the simulation are determined the schematic of the domain and its boundary condition (see Figure 8). All the walls are no-slip walls, stationary, except the top one, which moves with a horizontal velocity of 1 (length unit/time unit). As the top wall moves, the wall drags the fluid and a rotational movement is generated.

The big circle drawn in the schematic (see Figure 8) represents a vortex. If there is sufficient strength in the main vortex, it will generate smaller and secondary vortices, as in that case.

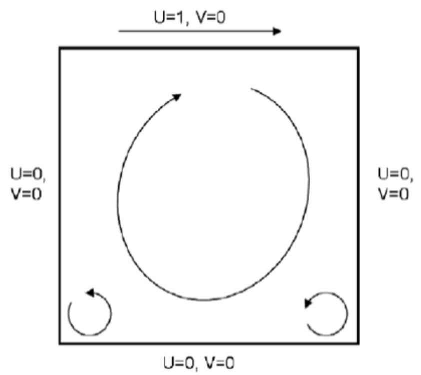
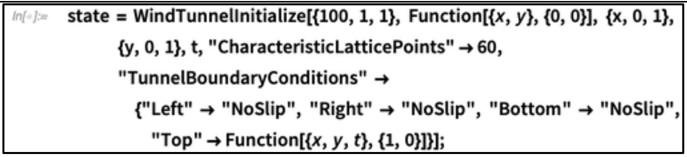


Figure 8. *Representation of the schematic of the domain and its boundary conditions. The velocity only affects the top wall of the wind tunnel, causing rotational movement in the fluid. Consequently, some vortices are generated in the domain.*

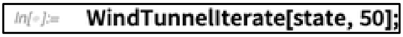
As a starting point for the code simulation, it is assumed a relationship between the strength of the vortex and the Reynolds number.

To demonstrate this relationship the simulation will be run for different values of Reynolds. First simulation will run in a Reynolds number of 100, and the code is:



```
In[•]:=  state = WindTunnelInitialize[{100, 1, 1}, Function[{x, y}, {0, 0}], {x, 0, 1},
           {y, 0, 1}, t, "CharacteristicLatticePoints" → 60,
           "TunnelBoundaryConditions" →
             {"Left" → "NoSlip", "Right" → "NoSlip", "Bottom" → "NoSlip",
               "Top" → Function[{x, y, t}, {1, 0}]}];
```

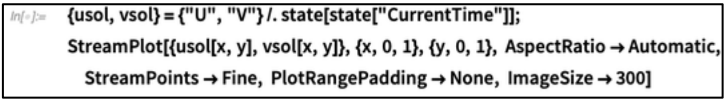
Iterating for 50 time units:



```
In[•]:=  WindTunnelIterate[state, 50];
```

As it is represented (see Figure 9), a primary vortex is formed in the middle of the domain. Additionally, a smaller secondary vortex appears at the bottom right of the box.

The code for running the simulation is shown below:



```
In[•]:=  {usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
         StreamPlot[{usol[x, y], vsol[x, y]}, {x, 0, 1}, {y, 0, 1}, AspectRatio → Automatic,
           StreamPoints → Fine, PlotRangePadding → None, ImageSize → 300]
```

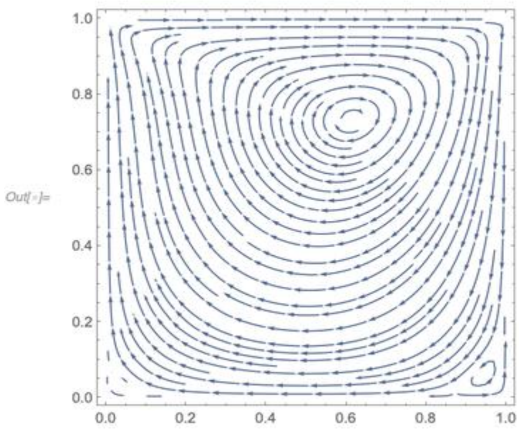
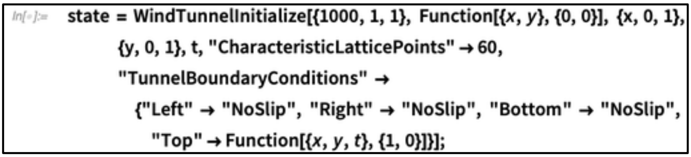


Figure 9. *Representation of the fluid behaviour. Rotational movements due to the moving top wall. Two vortices are generated. Due to the strength, one vortex is bigger than the other. Flow in a square box. Simulation at Re=100.*

If the Reynolds number is ramped up, these secondary vortices become stronger and larger. Thus, new vortices start developing in the corners of the field.

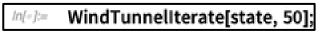
Now, the simulation will run at Reynolds number 1000.

The code used is:



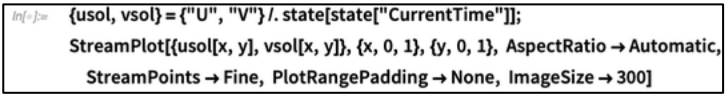
```
state = WindTunnelInitialize[{1000, 1, 1}, Function[{x, y}, {0, 0}], {x, 0, 1},
      {y, 0, 1}, t, "CharacteristicLatticePoints" → 60,
      "TunnelBoundaryConditions" →
        {"Left" → "NoSlip", "Right" → "NoSlip", "Bottom" → "NoSlip",
          "Top" → Function[{x, y, t}, {1, 0}]}];
```

Iterating, again, for 50 time units:



```
WindTunnelIterate[state, 50];
```

The mathematical language used for running the simulation is:



```
{usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
StreamPlot[{usol[x, y], vsol[x, y]}, {x, 0, 1}, {y, 0, 1}, AspectRatio → Automatic,
  StreamPoints → Fine, PlotRangePadding → None, ImageSize → 300]
```

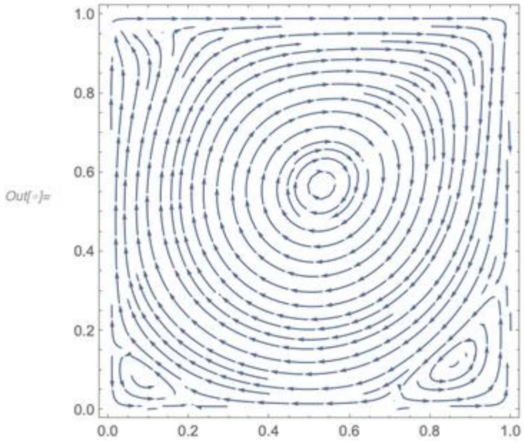
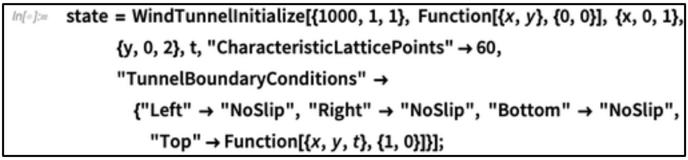


Figure 10. *Fluid behaviour represented by streamlines. Flow in a box. Generation of a primary bigger vortex in the middle of the domain and multiple secondary vortices in the corners. Streamlines from the top manifest the purpose of forming a new vortex. Flow in a square box. Simulation at Re= 1000.*

The fluid behaviour represented in the box (see Figure 10) shows the result of the simulation at number of Reynolds of 1000.

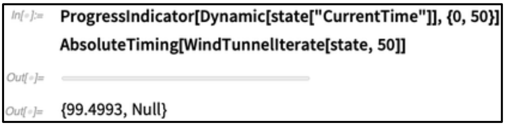
The resulted simulation (see Figure 10) shows that the primary vortex has moved closer to the center of the domain, and it has enough strength to be able to form secondary vortices. In the top part of the domain different behaviour of the fluid is presented. It is the initial step to generating the next vortex.

But what happens if the simulation is done it on a “tall” box instead of a square box? The boundary conditions remain the same but the domain changes in the y direction:



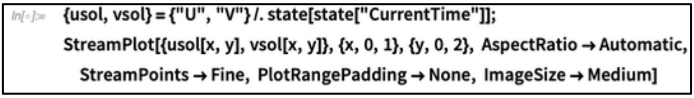
```
In[ ]:= state = WindTunnelInitialize[{1000, 1, 1}, Function[{x, y}, {0, 0}], {x, 0, 1},
          {y, 0, 2}, t, "CharacteristicLatticePoints" → 60,
          "TunnelBoundaryConditions" →
            {"Left" → "NoSlip", "Right" → "NoSlip", "Bottom" → "NoSlip",
              "Top" → Function[{x, y, t}, {1, 0}]}];
```

The instruction “Progressindicator” is used to track the simulation progress:



```
In[ ]:= ProgressIndicator[Dynamic[state["CurrentTime"]], {0, 50}]
        AbsoluteTiming[WindTunnelIterate[state, 50]]
Out[ ]=
Out[ ]= {99.4993, Null}
```

The streamlines, with the pertaining code, are visualized in the box (see Figure 11)



```
In[ ]:= {usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
        StreamPlot[{usol[x, y], vsol[x, y]}, {x, 0, 1}, {y, 0, 2}, AspectRatio → Automatic,
          StreamPoints → Fine, PlotRangePadding → None, ImageSize → Medium]
```

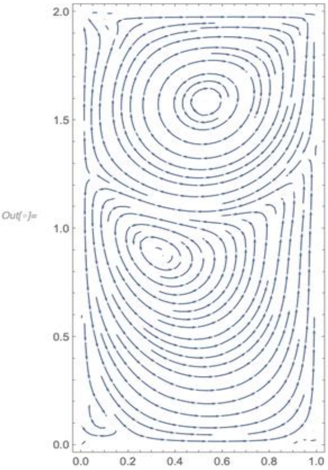


Figure 11. *Representation of the fluid behaviour. Flow in a “tall” box. Vortices are formed in the middle of the box. The strength associated is enough to obtain big vortices.*

- In the tall-box scenario, a primary vortex is developed near the top wall. This generated vortex, due to its strength cause a new vortex below the first one. If the second vortex generated is strong enough, it will create vortices at the bottom corners of the box.
- The flexibility wind tunnel concept mentioned will be studied by analyzing the flow behavior when a circular object is putted inside the wind tunnel (see Figure 12). The schematic (see Figure 12) is similar to the first fluid case studied (see Figure 4).

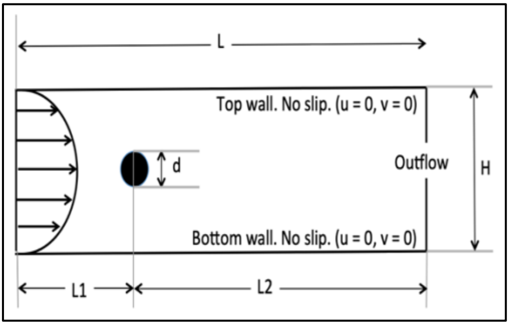
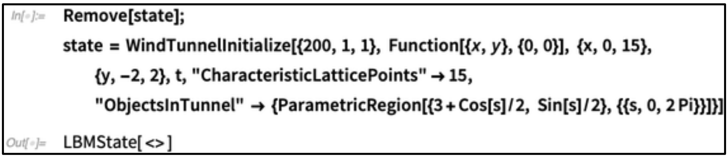

L
Top wall. No slip. (u = 0, v = 0)
d
Outflow
H
Bottom wall. No slip. (u = 0, v = 0)
L1
L2

Figure 12. *Representation of the scheme of the domain with immobile walls. Flow in a wind tunnel with an object introduced. Characteristic length of the domain: "d".*

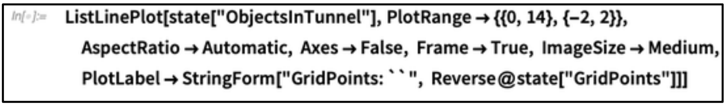
There are two length scales: "d" and "H". The choice of the characteristic length, though arbitrary, must tie back to some aspect of the physics of the flow. In this example, if the size of the object was to be increased or decreased, then the flow pattern behind it would be expected to change. Therefore, the natural choice is to use "d" as the characteristic length.

The cylinder will be placed at the position (3,0) in the domain and the size of it is 1 length unit. Therefore, the characteristic length will be 1. The domain size will be (0,15)x(-2,2). The object is specified as a ParametricRegion:



```
In[•]:= Remove[state];
        state = WindTunnelInitialize[{200, 1, 1}, Function[{x, y}, {0, 0}], {x, 0, 15},
           {y, −2, 2}, t, "CharacteristicLatticePoints" → 15,
           "ObjectsInTunnel" → {ParametricRegion[{3 + Cos[s]/2, Sin[s]/2}, {{s, 0, 2 Pi}}]}]
Out[•]= LBMState[<>]
```

The following code refers to the instructions to visualize the position of the object inside the device, to make sure the object is in the correct position.



```
ListLinePlot[state["ObjectsInTunnel"], PlotRange→{{0, 14}, {−2, 2}},
  AspectRatio→Automatic, Axes→False, Frame→True, ImageSize→Medium,
  PlotLabel→StringForm["GridPoints: ``", Reverse@state["GridPoints"]]]
```

As it is represented below (see Figure 13), the object is implemented as desired.

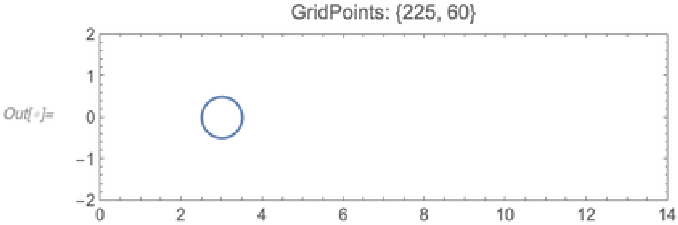
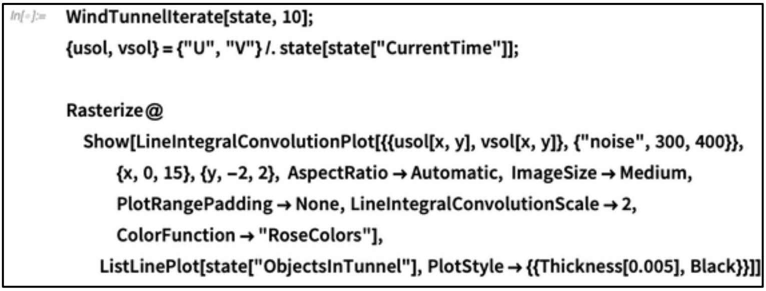


Figure 13. *Position of the added object inside the channel. Fluid in a wind tunnel. Unmoving walls. Study of the influence of an object in the behaviour of the fluid.*

As shown (see Figure 14), the result of running the simulation for 10 time:

The instructions for running the simulation for 10 times are shown next:



```
WindTunnelIterate[state, 10];
{usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];

Rasterize@
  Show[LineIntegralConvolutionPlot[{{usol[x, y], vsol[x, y]}, {"noise", 300, 400}},
     {x, 0, 15}, {y, −2, 2}, AspectRatio→Automatic, ImageSize→Medium,
     PlotRangePadding→None, LineIntegralConvolutionScale→2,
     ColorFunction→"RoseColors"],
   ListLinePlot[state["ObjectsInTunnel"], PlotStyle→{{Thickness[0.005], Black}}]]
```

The revealing results obtained from the simulation (see Figure 14) are mentioned below:

- Symmetric pair of vortices behind the cylinder
- The flow inside the cylinder

A close-up revels that there is some flow pattern inside the cylinder due to the implementation of LBM (see Figure 15).

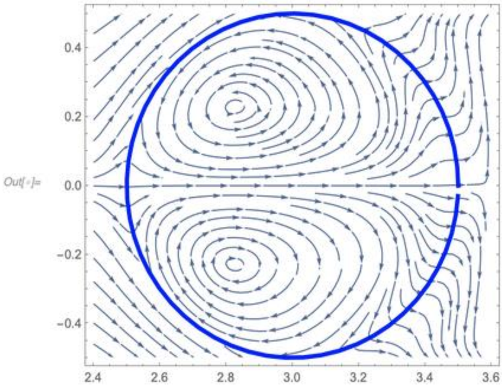
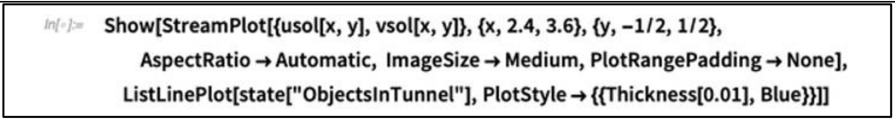


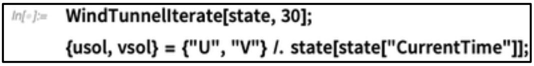
Figure 15. *Representation of the fluid behaviour and its cylinder influence. The streamlines representation confirms the creation of vortices inside the object. Incompressible fluid.*



```
In[•]:=  Show[StreamPlot[{usol[x, y], vsol[x, y]}, {x, 2.4, 3.6}, {y, -1/2, 1/2},
           AspectRatio → Automatic, ImageSize → Medium, PlotRangePadding → None],
         ListLinePlot[state["ObjectsInTunnel"], PlotStyle → {{Thickness[0.01], Blue}}]]
```

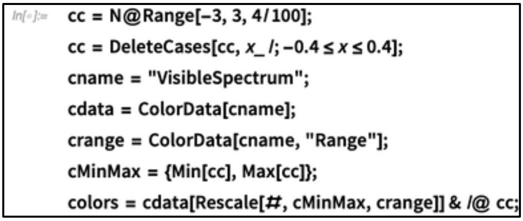
LBM computes a set of forces to be applied on the grid points such that the velocity at the surface representing the surface is 0. It does not specify what needs to happen inside the cylinder. Because of the incompressible flow inside the cylinder also exists a flow pattern. The important thing is that velocities at the boundaries of the object are 0 (no-slip).

Now the simulation runs for 30 time units to observe the behavior of the pattern behind the cylinder:



```
In[•]:=  WindTunnelIterate[state, 30];
         {usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
```

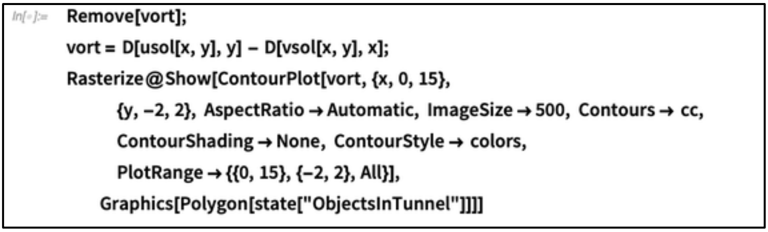
The instructions implemented below enable researchers to set up the color scheme for the contours:



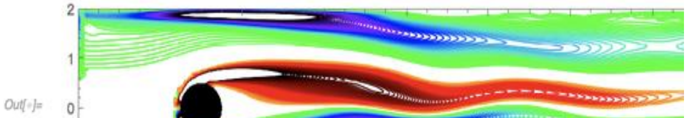
```
In[•]:=  cc = N@Range[-3, 3, 4/100];
         cc = DeleteCases[cc, x_ /; -0.4 ≤ x ≤ 0.4];
         cname = "VisibleSpectrum";
         cdata = ColorData[cname];
         crange = ColorData[cname, "Range"];
         cMinMax = {Min[cc], Max[cc]};
         colors = cdata[Rescale[#, cMinMax, crange]] & /@ cc;
```

The influence of an object in the channel is studied by focusing the running simulation on the vorticity variable.

The instruction code for the simulation is:



```
In[•]:=  Remove[vort];
         vort = D[usol[x, y], y] - D[vsol[x, y], x];
         Rasterize@Show[ContourPlot[vort, {x, 0, 15},
             {y, -2, 2}, AspectRatio → Automatic, ImageSize → 500, Contours → cc,
             ContourShading → None, ContourStyle → colors,
             PlotRange → {{0, 15}, {-2, 2}, All}],
           Graphics[Polygon[state["ObjectsInTunnel"]]]]
```

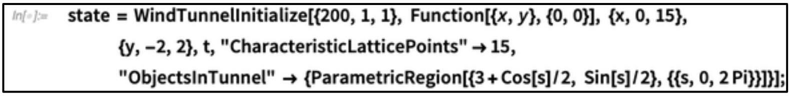


As it can be observed (see Figure 16) the symmetric pattern has been destroyed and the vorticity clearly shows a wavy behaviour, which means an instability in the wake of the cylinder. This instability continues to amplify, and, eventually, new vortices start forming behind the cylinder. This phenomenon is called “vortex shedding”. There is a shear layer generated at the surface of the cylinder that gets carried downstream.

This vortex shedding is also dependent on the Reynolds number. For small enough numbers there is no shedding. However, for values around 100-150, the shedding is observed. The vortex shedding dependence can be easily understood by observing the time evolution of the flow.

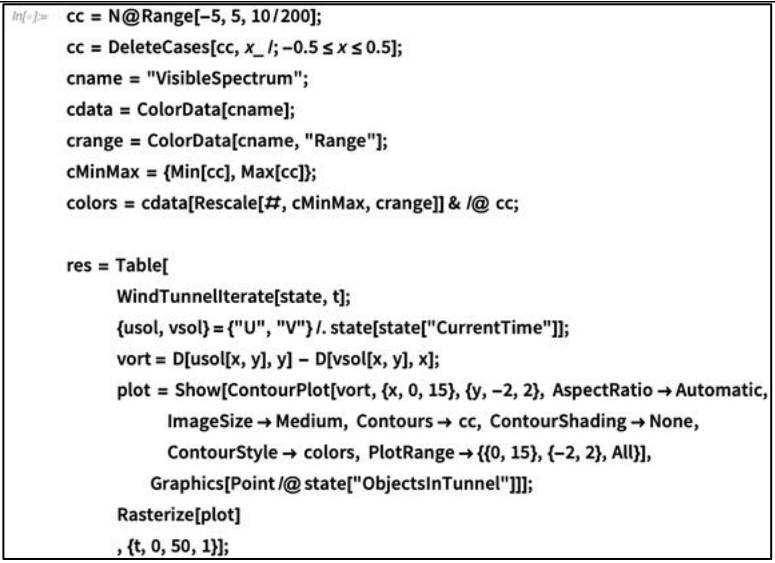
The first step is the definition of the characteristic terms and the objects in the tunnel.

The simulation code is:



```
In[•]:=  state = WindTunnelInitialize[{200, 1, 1}, Function[{x, y}, {0, 0}], {x, 0, 15},
           {y, −2, 2}, t, "CharacteristicLatticePoints" → 15,
           "ObjectsInTunnel" → {ParametricRegion[{3 + Cos[s]/2, Sin[s]/2}, {{s, 0, 2 Pi}}]}];
```

To produce a time evolution of the vorticity, a series of plots with the solution at each time will be generated, introducing the following code:

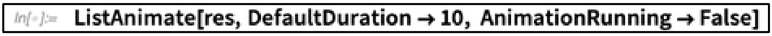


```
cc = N@Range[-5, 5, 10/200];
cc = DeleteCases[cc, x_ /; -0.5 ≤ x ≤ 0.5];
cname = "VisibleSpectrum";
cdata = ColorData[cname];
crange = ColorData[cname, "Range"];
cMinMax = {Min[cc], Max[cc]};
colors = cdata[Rescale[#, cMinMax, crange]] & /@ cc;

res = Table[
      WindTunnelIterate[state, t];
      {usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
      vort = D[usol[x, y], y] - D[vsol[x, y], x];
      plot = Show[ContourPlot[vort, {x, 0, 15}, {y, -2, 2}, AspectRatio → Automatic,
            ImageSize → Medium, Contours → cc, ContourShading → None,
            ContourStyle → colors, PlotRange → {{0, 15}, {-2, 2}, All}],
         Graphics[Point /@ state["ObjectsInTunnel"]]];
      Rasterize[plot]
      , {t, 0, 50, 1}];
```

Once the simulation has run it's easy to observe the two vortices forming in the back of the cylinder with the shear layer slowly getting perturbed. These vortices go increasing until finally breaking into a vortex shedding. (see Figure 17).

The animated representation is coded:



```
ListAnimate[res, DefaultDuration → 10, AnimationRunning → False]
```

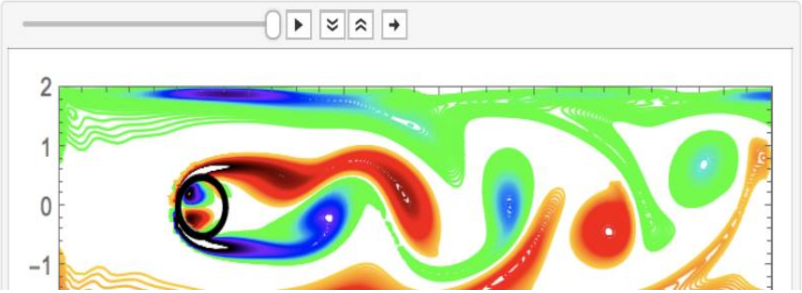


Figure 17. *Representation of the fluid behaviour over the time and its cylinder influence, focusing on the vorticity. Wavy behaviour leads to vortices formation. As times goes by, the dimension of these vortices increases until it breaks into a vortex shedding.*

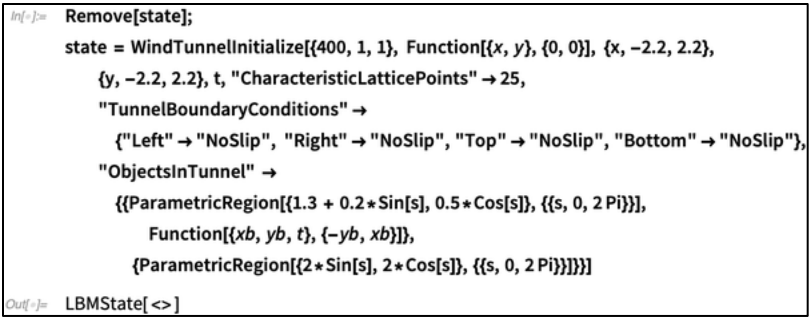
APPENDIX 4: OBSERVING DISTURBANCES CAUSED BY A MOVING OBJECT

The objective of this section is to study the kind of disturbances developed when an object moves through a still fluid.

The new case of fluid behavior lies object. This object is placed near the tank wall and follows the tank boundary in a circular path. The flexibility of LBM with immersed boundary allows us great flexibility with moving objects.

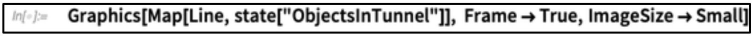
Before setting up the problem, the characteristic terms must be defined. The simulation will be performed at a Reynolds number of 400 and both characteristic length and velocity are specified as unity. Inside the tunnel there are two objects: the first object refers to the circular tank, held stationary, while the second one is the elliptical object, which moves through the tank.

The simulation runs as follows:



```
In[•]:= Remove[state];
        state = WindTunnelInitialize[{400, 1, 1}, Function[{x, y}, {0, 0}], {x, -2.2, 2.2},
           {y, -2.2, 2.2}, t, "CharacteristicLatticePoints" → 25,
           "TunnelBoundaryConditions" →
             {"Left" → "NoSlip", "Right" → "NoSlip", "Top" → "NoSlip", "Bottom" → "NoSlip"},
           "ObjectsInTunnel" →
             {{ParametricRegion[{1.3 + 0.2*Sin[s], 0.5*Cos[s]}, {{s, 0, 2 Pi}}],
                 Function[{xb, yb, t}, {-yb, xb}]},
               {ParametricRegion[{2*Sin[s], 2*Cos[s]}, {{s, 0, 2 Pi}}]}}]
Out[•]= LBMState[<>]
```

To check the geometry (see Figure 18) of the problem the discretized object will be extract when doing the initialization:



```
In[•]:= Graphics[Map[Line, state["ObjectsInTunnel"]], Frame → True, ImageSize → Small]
```

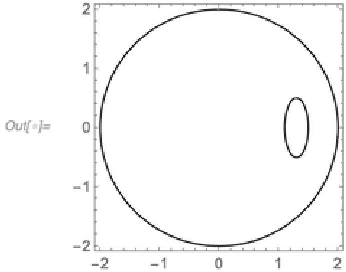
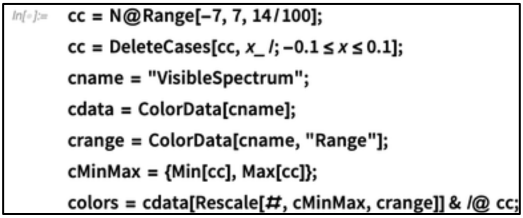


Figure 18. *Representation of the geometry of the object. Circular path. Two objects added: circular stationary tank and elliptical object which moves through the tank. Object moving through a still fluid.*

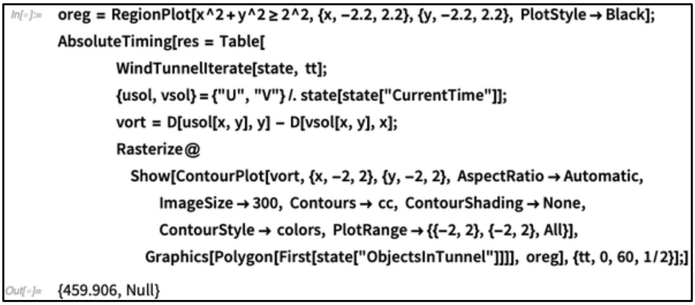
As did it in other fluid examples mentioned, vorticity will be analyzed in depth. The color scheme and the levels of contours that will be plotted should be defined.

The simulation code is:



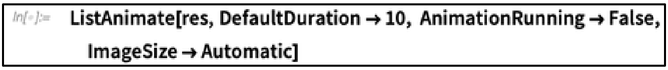
```
In[=]:=  cc = N@Range[−7, 7, 14/100];
         cc = DeleteCases[cc, x_ /; −0.1 ≤ x ≤ 0.1];
         cname = "VisibleSpectrum";
         cdata = ColorData[cname];
         crange = ColorData[cname, "Range"];
         cMinMax = {Min[cc], Max[cc]};
         colors = cdata[Rescale[#, cMinMax, crange]] & /@ cc;
```

For running the simulation 60 time units:



```
In[=]:=  oreg = RegionPlot[x^2 + y^2 ≥ 2^2, {x, −2.2, 2.2}, {y, −2.2, 2.2}, PlotStyle → Black];
         AbsoluteTiming[res = Table[
                 WindTunnelIterate[state, tt];
                 {usol, vsol} = {"U", "V"} /. state[state["CurrentTime"]];
                 vort = D[usol[x, y], y] − D[vsol[x, y], x];
                 Rasterize@
                   Show[ContourPlot[vort, {x, −2, 2}, {y, −2, 2}, AspectRatio → Automatic,
                       ImageSize → 300, Contours → cc, ContourShading → None,
                       ContourStyle → colors, PlotRange → {{−2, 2}, {−2, 2}, All}],
                     Graphics[Polygon[First[state["ObjectsInTunnel"]]]], oreg], {tt, 0, 60, 1/2}];]
Out[=]=  {459.906, Null}
```

By running the time evolution of the fluid disturbance, it is shown (see Figure 19) the geometric pattern generated within the tank, initially before settling down to a more uniform circular disturbance.



```
In[•]:=  ListAnimate[res, DefaultDuration → 10, AnimationRunning → False,
           ImageSize → Automatic]
```

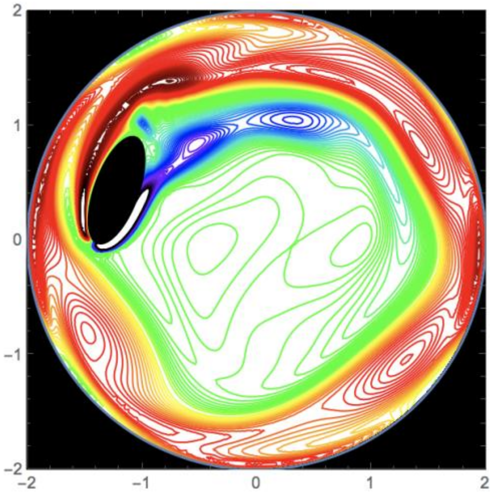


Figure 19. *Evolution of the fluid behaviour over the time. Result of the simulation. Geometric pattern generated before settling down a uniform circular disturbance.*