APPRAISAL OF FLOW SIMULATION BY THE LATTICE BOLTZMANN METHOD

Guillermo Izquierdo Bouldstridge

Imperial College of London
Department of Aeronautics

Master's Thesis

Supervisor:
Dr. Joaquim Peiró

September 2016 - January 2017

INDEX

ABSTRACT

This research project presents an overview of the Lattice Boltzmann Method (LBM), an alternative numerical approach to conventional CFD. LBM has increased in popularity among the scientific community in recent years, due to its promising abilities. Namely, it claims to achieve the same level of accuracy as that of traditional CFD, while offering new benefits such as easy parallelization and the possibility of implementing complex and multiscale flows. Unlike conventional CFD which focuses on the numerical solution of the Navier Stokes Equations, the Lattice Boltzmann Method focuses on microscopic particle interactions to represent the macroscopic behaviour of the fluid.

The aim of this project is to appraise the ability of the Lattice Boltzmann Method to accurately simulate incompressible flows and to analyse its accuracy performance and stability. This report presents the theoretical basis of this novel method, as well as a verification of its convergence results through some examples. These examples are implemented through an open-source code (*Palabos*). This project not only focuses on matching the LBM solutions with analytical or existing solutions, but it also focuses on studying the effect that the parameters of the model have on the results provided, on stability and on computational cost.

The results and their analysis show that LBM is an accurate method for representing incompressible flows. The report also describes how to implement the Lattice Boltzmann Method and suggests some ways to continue the work further.

1. INTRODUCTION

Computational Fluid Dynamics (CFD) traditionally has simulated flows by considering the fluid as a continuum model governed by the non-linear Navier Stokes equations, which are based on the conservation of macroscopic properties (momentum and mass). These equations are then discretized in space and time to obtain a numerical solution.

However, in recent years, a new approach to solve flow problems in the field of CFD is increasing in popularity. That new approach is the Lattice Boltzmann Method (LBM). Just to illustrate this increase in popularity, scientific articles written between 1950 and 1985 included the words “Lattice Boltzmann” 29,700 times, while articles between 1985 and 2017, 332,000 times (information from Google Scholar). It was firstly introduced by Ludwig Boltzmann in 1872, but it was not until the second half of the XX century and especially in the last twenty years that the scientific community has noticed the advantages that this model can produce.

Instead of focusing on the macroscopic representation of the flow, LBM uses a microscopic representation of the flow, as well as a discretization in space and time. LBM is a model that works under the basic idea that a fluid is composed of interacting molecules that can be described by classical mechanics. The method reproduces the physics of the fluid by applying simple physical phenomena such as streaming in space and billiard-like collision interactions between microscopic particles (C.Sukop & T.Thorne, 2006).

LBM has been proved to provide some computational advantages compared to traditional CFD. LBM can easily represent complex physical phenomena, ranging from multiphase flows to chemical interactions between the fluid and the surroundings. LBM is particularly suited for very efficient parallel processing, which considerably reduces computational cost.

These advantages have increased the interest in Lattice Boltzmann Method in the last years. Theory behind LBM is well documented, for instance the book written by Succi (2001) provides a good up-to-date account of the LBM theory. Concerning LBM code, open-source code provided by *Palabos* offers an accurate implementation of LBM for incompressible flows.

This project aims to compare the accuracy and performance of LBM to examine whether LBM is indeed a reasonable replacement for conventional CFD. Firstly, the basic theory of LBM is explained and secondly some real examples of flow simulations are tested to analyze results provided by LBM. Throughout this process, key aspects from LBM will be studied.

2. THEORY

This is a brief summary of the basic elements of Lattice Boltzmann Method, for further information I refer to references (Succi, 2001), (Chen & D.Doolen, 1998) or (C.Sukop & T.Thorne, 2006).

2.1 Boltzmann Equation

LBM evolves from an equation proposed by Ludwig Boltzmann in 1872 known as the Boltzmann equation (BE). Therefore, it is important to understand the basic idea behind the Boltzmann equation to be able to comprehend how LBM works.

Boltzmann equation focuses on the microscopic representation of the fluid to analyse its behaviour. As a fluid is composed by many particles, a statistical treatment is necessary to avoid massive number of calculations. That is the probability density function $f(\vec{x},\vec{v}$,t), which describes the probability of finding a particle around position $\vec{x}$ at time t with velocity $\vec{v}$. Therefore, the sum of all the probability density function at one specific position and time will result in the total number of particles in that position and time. It is precisely that probability density function that is the main object of the Boltzmann equation and consequently of LBM.

The Boltzmann equation describes the behaviour of $f(\vec{x},\vec{v}$,t) as a function of the microdynamic interactions, which refer to collisions between particles of the fluid (Succi, 2001). This equation is referred to as the Boltzmann equation and the basic ideas behind its development are described in the following.

Assuming that there are no collisions between particles and that the flow evolves only due to its own velocity, the evolution of the distribution function $f(\vec{x},\vec{v}$,t) during an interval of time dt is

$$f\left(\vec{x}+\vec{v}\cdot dt,\vec{v}+\frac{\vec{F}^{\,ext}}{m}\cdot dt,\mathrm{t}+\mathrm{dt}\right) = f\ (\ \vec{x},\vec{v},t)\ , \tag{1}$$

that is referred to as the *streaming process*. Assuming $f\ (\ \vec{x},\vec{v},t)$ is constant, a first-order Taylor series expansion around $(\vec{x},\vec{v},t)$ leads to

$$\left(\frac{\partial}{\partial t}+\vec{v}\cdot\partial_{\vec{x}}+\frac{\vec{F}^{\,ext}}{m}\cdot\partial_{\vec{v}}\right)\cdot f\ (\ \vec{x},\vec{v},t) = 0\ . \tag{2}$$

However, equation (2) does not consider collisions that may occur between moving particles. Collisions taking particles in or out the streaming trajectory can be represented by adding a term, $C_{[f]}$, on the right-hand side of equation (2):

$$\left(\frac{\partial}{\partial t}+\vec{v}\cdot\partial_{\vec{x}}+\frac{\vec{F}^{\,ext}}{m}\cdot\partial_{\vec{v}}\right)\cdot f\ (\ \vec{x},\vec{v},t) = C_{[f]}\ . \tag{3}$$

This collision term of equation (3) needs to represent how particles at same position change their velocities due to collision. Boltzmann made the following important assumptions: only binary collisions are considered, influence of container walls are not considered, influence of the external force is neglected when treating collisions and there is no relation between velocity and position of the molecule (known as molecular chaos) (Succi, 2001).

Using these assumptions, Boltzmann proposed a collision operator as follows. Given two particles with relative speed g=v_1-v_2 (see Figure 1) and solid angle Ω (defined by the scattering angle *X* from Figure 1 and the azimuthal angle around collisional plane), the differential cross section is defined as the total number of particles with relative speed g and solid angle Ω.

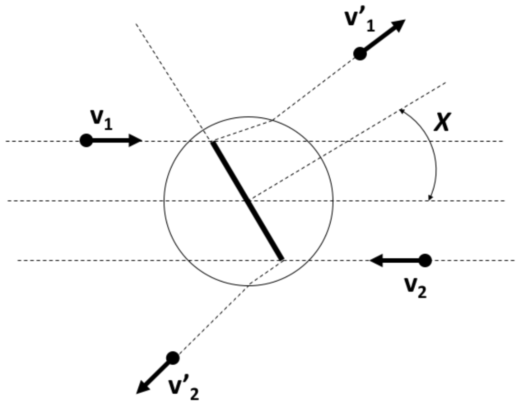


Figure 1 Scattering angle associated with a binary collision

Those parameters represent the underlying particle dynamics of the collision. Therefore, the change in the probability distribution function due to collision is a function of those parameters, which is given by

$$C_{[f]} = \int\int (f_{1'} \cdot f_{2'} - f_1 \cdot f_2) \cdot g \cdot \sigma(\mathrm{g}, \Omega) \cdot \mathrm{d}\Omega \cdot \mathrm{d}\overrightarrow{v_2} \ , \qquad (4)$$

where $f_{1'}$ refers to the distribution function after collision, f_1 to the distribution function before collision and $\sigma(g, \Omega)$ to the differential cross section, which is the total number of particles with relative speed g and solid angle Ω.

Substituting equation (4) in equation (3) leads to the Boltzmann equation, which takes the form

$$\left(\frac{\partial}{\partial t} + \vec{v} \cdot \partial_{\vec{x}} + \frac{\vec{F}^{ext}}{m} \cdot \partial_{\vec{v}}\right) \cdot f = \int\int (f_{1'} \cdot f_{2'} - f_1 \cdot f_2) \cdot g \cdot \sigma(\mathrm{g}, \Omega) \cdot \mathrm{d}\Omega \cdot \mathrm{d}\overrightarrow{v_2} \ , \qquad (5)$$

where the left-hand side refers to the streaming motion of particles, while the right-hand side of the equation represents the effect of collisions between particles.

2.2 BGK Lattice Boltzmann Method

The Boltzmann equation is hard to solve by itself due to its continuum form and its complex non-linear collision operator. Many authors have proposed alternative forms to simplify the collision operator. The simplest and most effective one was proposed by Bhatnagar, Gross and Krook in 1954 (Bhatnagar et al., 1954). For instance, *Palabos* open-source code uses this version of Lattice Boltzmann Method.

2.2.1 BGK Lattice Boltzmann Equation

Bhatnagar, Gross and Krook proposed in 1954 a collision operator for the Boltzmann Equation that consists of a relaxation of the probability distribution function towards a local equilibrium, that takes the form

$$\left(\frac{\partial}{\partial_t} + \vec{v} \cdot \partial_{\vec{x}}\right) \cdot f\ (\vec{x}, \vec{v}, t) = \frac{1}{l} \cdot (f\ (\vec{x}, \vec{v}, t) - f^{eq}\ (\vec{x}, \vec{v}, t))\ . \tag{6}$$

External forces are not considered. This relaxation process can be interpreted physically as the tendency of the probability distribution function to approach its equilibrium state $f^{eq}\ (\vec{x}, \vec{v}, t)$ after a time l (the specific value for the equilibrium function is discussed in the following sections).

Equation (6) is discretized in space, velocity and time (variables that define f) to obtain a numerical solution. Space is discretized by defining a lattice with cells of dimension Δx. Time is discretized by defining a time step Δt. The lattice speed is therefore given by $\vec{v} = \Delta x / \Delta t$. The direction of this velocity can also be discretized into a specific range ($\alpha = 1 \ldots M$) without modifying the solution. By doing this, the hydrodynamic moments of $f\ (\vec{x}, \vec{v}, t)$, which represent the physics of the flow, remain the same, since the moment integral can be evaluated exactly using a quadrature up to a certain order in the velocity space directions (Mei & Yu, 2002). By applying these discretizations, equation (6) evolves to

$$f_\alpha\ (\vec{x} + \vec{v_\alpha} \cdot \Delta t, t + \Delta t)\ -\ f_\alpha\ (\vec{x}, t) = \frac{1}{\tau} \cdot \left(f_\alpha\ (\vec{x}, t) - {f^{eq}}_\alpha\ (\vec{x}, t)\right), \quad \alpha = 1 \ldots M. \tag{7}$$

where $f_\alpha\ (\vec{x}, t) =\ f\ (\vec{x}, \vec{v_\alpha}, t),\ \ \vec{v_\alpha} \cdot \Delta t = \Delta x$ and $\tau = {}^{l}/_{\Delta t}$.

2.2.2 The equilibrium distribution function

Determining the value of the equilibrium distribution function is important because it defines the behaviour of the model. This function is normally established as a truncated Taylor series expansion up to second order of the Maxwellian distribution function. The Maxwellian distribution function proposes a distribution probability for the particle speeds in idealized gases as

$$f_{Maxwellian} = \rho \cdot \left(\frac{1}{2 \cdot \pi \cdot R \cdot T}\right)^{D/2} \cdot e^{\left(-(v-u)^2 / R \cdot T\right)}\ , \tag{8}$$

where D refers to the number of dimensions of the domain. To discretize equation (8) into the specific range of velocities ($\alpha = 1 \ldots M$), the distribution function is approximated as

$$f_{Maxwellian} = \sum_\alpha w_\alpha \cdot {f^{eq}}_\alpha \quad ; \qquad \sum_\alpha w_\alpha = 1 \tag{9}$$

where ${f^{eq}}_{\alpha}$ are obtained via a Taylor series expansion of equation (8) up to second order, which using equations (9) leads to

$$f_{\alpha}^{eq} = \rho \cdot w_{\alpha} \cdot \left(1 + \frac{v_{\alpha} \cdot u}{c_s^2} + \frac{(v_{\alpha} \cdot u)^2}{2 \cdot c_s^4} - \frac{u^2}{2 \cdot c_s^2}\right), \tag{10}$$

where w_{α} is a specific weight for each velocity equilibrium distribution function (equation (9)), c_s the sound speed, v_{α} the lattice velocity (Δx/Δt) and u the macroscopic speed of the flow.

The weights w_{α} are adjusted to recover the incompressible Navier-Stokes equations. A proof that LBGK satisfies Navier Stokes equations is developed in the article (Chen & D.Doolen, 1998). Depending on the choice of the discrete velocity space, weights w_{α} and c_s take different values. (Qian, et al., 1992) proposed a whole family of solutions for specific discrete velocities and their w_{α} and c_s values, by imposing conservation of mass, momentum and entropy. For instance, the case of 2 dimensions and 9 velocity directions has the following w_{α} and c_s values:

$$w_{\alpha} = \begin{cases} 4/9 & \alpha = 0 \\ 1/9 & \alpha = 1,2,3,4 \\ 1/36 & \alpha = 5,6,7,8 \end{cases} \quad ; \quad c_s = 1/\sqrt{3} \ . \tag{11}$$

Velocity discretization space takes the name of D*n*Q*m*, meaning *m* velocities in a *n*-dimension space. Among the solutions proposed, D1Q3, D2Q9 and D3Q19 are the most popular ways of discretizing velocity space (see Figure 2).

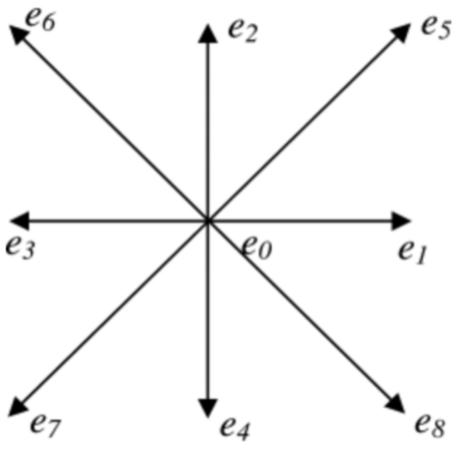


Figure 2 D2Q9 velocity space

2.2.3 The relaxation parameter

The relaxation parameter, τ from equation (7), is an important parameter of Bhatnagar-Gross-Krook Lattice Boltzmann Method (LBGK). It establishes how fast the particle distribution function achieves the equilibrium distribution function. The solution given by the model will depend on its value, which cannot be arbitrary, since it influences the viability of the model to recover the Navier-Stokes equations. An analysis proposed by (Chapman & Cowling, 1992) confirms that LBGK reproduces exactly the Navier Stokes equations for quasi-incompressible flows when

$$\tau = \frac{v}{c_s^2} + {}^1\!/_2 \; , \tag{12}$$

where v is the kinematic viscosity and c_s is the speed of sound.

2.2.4 Computational aspects

When this model is programmed, streaming and collision are calculated as separate processes. This means dividing equation (6) into the following two steps (Mei & Yu, 2002):

$$\text{Collision:} \quad \widetilde{f_\alpha}\,(\vec{x},t) = f_\alpha\,(\vec{x},t) - \frac{1}{\tau}\cdot\,(f_\alpha\,(\vec{x},t) - f^{eq}{}_\alpha\,(\vec{x},t)) \tag{13}$$

$$\text{Streaming:} \quad f_\alpha\,(\vec{x} + \overrightarrow{v_\alpha}\cdot\Delta t, t + \Delta t) = \widetilde{f_\alpha}\,(\vec{x},t) \tag{14}$$

Firstly, the collision process is calculated and particles at the same position redistribute their velocities due to interactions between them; secondly, streaming is calculated and particles change position due to their velocity.

Another interesting aspect concerning the theory of LBM that should be noted and that simplifies computational calculations is that the method enables us to calculate macroscopic parameters directly from the distribution function. Density (ρ) and momentum density (ρu) can be calculated as follows (K. Aidun & R.Clausen, 2010):

$$\rho = \textstyle\sum_\alpha f_\alpha \, , \tag{15}$$

$$\rho \mathrm{u} = \textstyle\sum_\alpha f_\alpha \cdot v_\alpha \, . \tag{16}$$

2.3 Lattice Boltzmann Method stability

LBM is a numerical method that may develop instabilities depending on the parameters selected to run the model. I present only a brief overview of LBM stability. Further information can be found in the papers (D. Sterling & Chen, 1996) and (Worthing, et al., 1997).

Since the collision operator includes an equilibrium function that is required to satisfy the Maxwell-Boltzmann distribution function (equation (8)), as well as the Navier

Stokes equations, entropy is not guaranteed to not decrease during collision. In case entropy decreases, instabilities in the model may occur.

The article (D. Sterling & Chen, 1996) develops a study of LBM stability by linearizing the Lattice Boltzmann scheme, which is non-linear due to the form of the equilibrium function in the collision term (equation (8)). To linearize the system, they expand f_α as

$$f_\alpha\,(\vec{x},t) = \overline{f_\alpha^{(0)}} + f'_\alpha(\vec{x},t)\ , \tag{17}$$

where $\overline{f_\alpha^{(0)}}$ refers to the global equilibrium particles which are constant in space and time. Term $f'_\alpha(\vec{x},t)$ does not necessarily match with $f_\alpha^{(neq)}(\vec{x},t)$. By applying that change, Lattice Boltzmann Equation becomes

$$g_\alpha\,(f_\alpha) = f_\alpha(\vec{x},t) - \frac{1}{\tau}(f_\alpha(\vec{x},t) - f_\alpha^{(0)}(\vec{x},t))\ . \tag{18}$$

A Taylor expansion of g_α about $\overline{f_\alpha^{(0)}}$ leads to the following linearized system:

$$f'_\alpha\,(\vec{x} + \vec{v_\alpha}\cdot dt, t+dt) = G\,\cdot\,f'_\alpha\,(\vec{x},t)\ , \tag{19}$$

where G is the Jacobian matrix of the coefficients of the Taylor expansion and it is independent of space and time. The eigenvalues of that matrix determine the stability of the system, since the model will be stable if all eigenvalues have modulus less than the unity. The eigenvalues of G are {1, 1-1/τ} and stability is therefore guaranteed when τ > 0.5 (D. Sterling & Chen, 1996).

However, when testing values for τ slightly higher than 0.5 the model also generates instabilities. That it is due to the nonlinear terms in the equilibrium distribution function. When τ approaches 0.5, kinematic viscosity decreases significantly (equation (12)) and high Reynolds number is obtained (equation (20)). Equation (20) represents the way LBM calculates the Reynolds number, but this is not necessarily the physical Reynolds number of the case that the model is testing. To avoid confusions with the physical Reynolds number, that relationship of parameters that appears in equation (18) will be named Re_{LBM} :

$$Re_{LBM} = \frac{N\cdot v}{\mathrm{v}}\ , \tag{20}$$

where N refers to the resolution of the lattice (1/Δx), v is the lattice velocity (Δx/Δt) and v is the kinematic viscosity of the fluid.

To avoid those instabilities, lattice velocity (Δx/Δt) should be small enough to retain a stable scheme and to keep higher-order terms from the Chapman-Enskog expansion negligible (D. Sterling & Chen, 1996). Since the velocity is limited, if high Reynolds number is needed, resolution should be increased or relaxation parameter decreased until its boundary limit.

3. SOFTWARE

Beyond providing an appraisal to Lattice Boltzmann Method theory, this project aims to execute some practical examples and analyse the results that LBM provides. To perform this analysis, a code is used provided by *Palabos (Parallel lattice Boltzmann open source)*, which is an organization focused on developing LBM software. This project uses the latest version of Palabos (version 1.5), which was launched in January 2015.

Palabos provides a C++ code that simulates flow behaviour using Lattice Boltzmann Method. It provides a set of functions and algorithms to be able to implement different kinds of geometries and flows. Some examples are already programmed by Palabos to illustrate how the code works. It must be emphasized that Palabos always calculates Reynolds number with equation (18) and if the geometry requires another calculation for Reynolds number, it should be adapted to that of equation (18). Since not all geometries concerning this project are implemented by *Palabos* with its examples cases, this project has needed to program those new cases using Palabos structure. This has been done by adding code to the already existent C++ program of Palabos, in order to develop the cases and calculations included in this project.

It should be noted that the Palabos code can reproduce the physics of those flows: Incompressible Navier-Stokes equations, weakly compressible, non-thermal Navier-Stokes equations, flows with body-force term, thermal flows with Boussinesq approximation, single-component multi-phase fluids (Shan/Chen model), multi-component multi-phase fluids (Shan/Chen model) and static Smagorinsky model for fluid turbulence (Flowkit, 2011). More information about the code can be found on its website (www.palabos.org) or in the reference provided.

The *Palabos* code was run in parallel using a two-core 2.9 GHz processor with a Ram memory of 8 GB.

4. VERIFICATION OF LBM

This section discusses the verification of the Lattice Boltzmann Method (LBM) for flow simulation. A total number of five cases are simulated, all of them in two dimensions. All cases are run under double precision, unless the first one which is tested in single precision. Since LBM is implemented on a lattice with straight lines, the first cases analysed contain only cartesian geometries, which consist of non-curved geometries with either horizontal or vertical walls. Complex surfaces require introducing curved geometries that do not necessarily match with the lattice lines. The fluid implemented in all geometries is an incompressible, single phase and thermal.

Results are compared to analytical solutions or to results obtained by other CFD methods. To evaluate this comparison quantitatively, the error between both solutions is calculated. The results provided by Lattice Boltzmann Method will mainly depend on the number of iterations run, the mesh size selected and the relaxation parameter chosen. Those cases aim to give a general overview of how those variables influence on the accuracy of the results.

An analysis of the iteration convergence is performed to obtain the influence of the number of iterations. It is also indispensable to find a convergence criterion to know when to stop the code and then consider the solution steady in time. Computational cost will be another aspect to consider when analysing number of iterations.

An analysis of the mesh convergence is performed to obtain the influence of the mesh size selected. Therefore, different resolutions for the lattice are executed to find which relationship exists between Δx and accuracy of the results.

The same physical problem is tested with different values of τ to obtain the influence of the relaxation parameter. An analysis of the effect of the relaxation parameter on accuracy will be also carried out.

When analysing curved geometries, a new way of implementing boundary conditions is discussed in section 2, since the geometry does not fit exactly with the lattice lines, by the bounce-back boundary condition.

4.1 Cartesian geometries

This section presents a study of three cases that contain staircase geometries: Blasius flat plate, Poiseuille flow and backward facing step. They all conform to a cartesian grid, since they do not have curved lines on their geometries. I analyse iteration and mesh convergence, the influence of the relaxation parameter and the computational cost. The main goal is to illustrate the behaviour of the model according to how the case is implemented.

4.1.1 Blasius Flat Plate

This geometry consists of a single plate submerged in a flow that moves at a constant velocity. The flow has a Reynolds number of 10,000, an inlet velocity of 0.05 m/s, a kinematic viscosity of 0.000005 m^2/s and a density of 1 kg/m^3. The domain has dimensions 0.5m x 1m, the plate is allocated in the bottom wall and the plate has a length of 0.4m (see Figure 3). The domain is divided into cells and the resolution (number of cells per meter) varies from 600 to 3000.

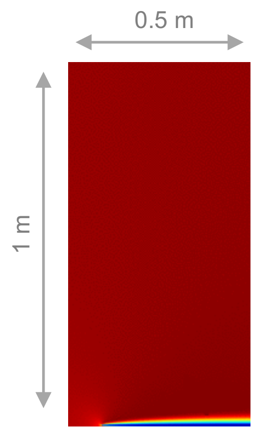


Figure 3 Dimensions of the Blasius flat plate

Since Reynolds number is quite high, resolution needs to also be high due to the reason exposed in the section concerning LBM stability. The domain is divided into 1400 cells per meter to increase the value of the relaxation parameter and to avoid instabilities.

The analytical solution of the flat plate and its boundary layer has been well documented (Fang et al., 2006). To validate the accuracy of the results provided by LBM, the solution is compared to the exact analytical solution proposed by Blasius, which describes the velocity flow as a function of the Reynolds number and the quotient between the height and the boundary layer thickness.

I analyse the iteration convergence to find the convergence criterion, the mesh convergence to find which mesh is small enough to represent the model properly and I evaluate computational cost per mesh size.

4.1.1.1 Iteration convergence

Lattice Boltzmann Method is an iterative method that needs a condition to stop iterating. Therefore, first thing to do is to establish a criterion to know when to stop the code. The condition to stop iterating will be fixed when difference in results between two consecutive iterations is smaller than a certain value.

To determine this value, the code is run for 40,000 iterations. At each iteration, the program calculates the difference in results (error) with previous iteration (Figure 4). Error is calculated as the average difference of the velocity in the N points allocated in the vertical line at the middle of the flat plate, which reads as

$$error = \frac{\sum_i^N |u_{it} - u_{it-1}|}{N} \quad . \tag{21}$$

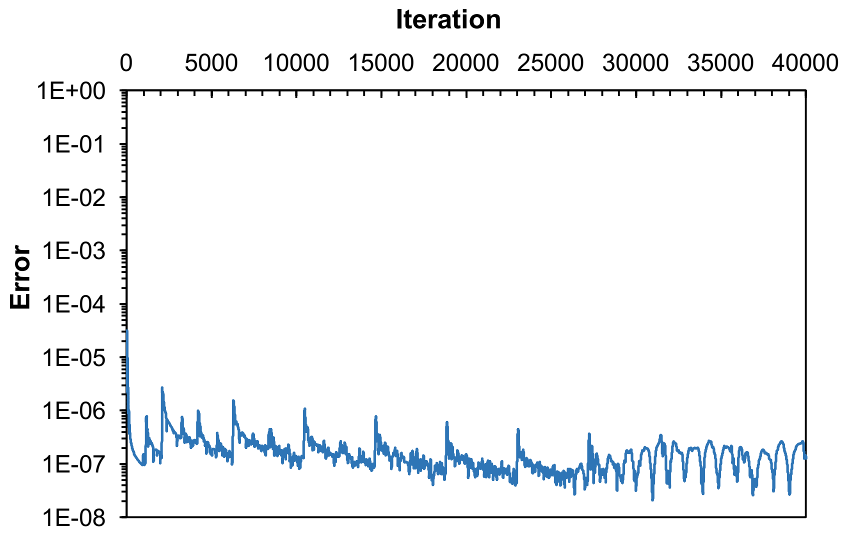


Figure 4 Error between consecutive iterations

Figure 4 shows that error between consecutive iterations decreases as number of iterations increases. Since the code is executed entirely in single precision, machine epsilon is of the order of 10^{-6}, which explains why error remains between 10^{-6} and 10^{-8} for high iterations. Error similar in magnitude to machine epsilon can be caused by the round-off error and not by progress of LBM.

However, Figure 4 does not provide an indication of when to stop the code. An analysis studying how close the results provided are to the exact Blasius solution among iterations is needed (Figure 5). The error of the iteration at which its solution is like the exact one will be the minimum error permitted to continue iterating.

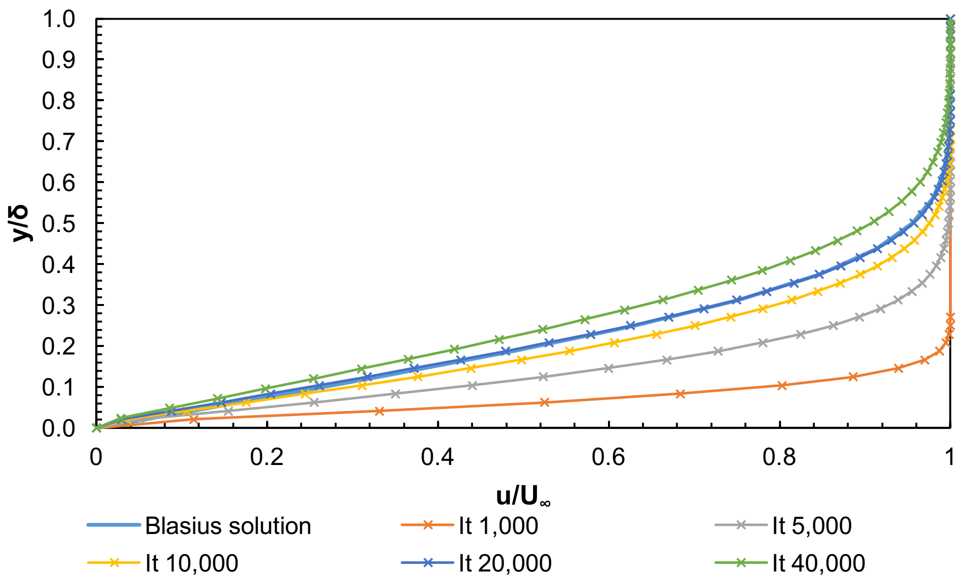


Figure 5 Velocity profile per iteration

Figure 5 shows that by iteration 20,000, results provided by the program are like the Blasius exact solution. Before iteration 20,000, error (Figure 4) is always higher than $5 \cdot 10^{-8}$. Therefore, the code needs to stop iterating when the error is smaller than $5 \cdot 10^{-8}$. Criterion of convergence is established.

4.1.1.2 Mesh convergence

LBM uses a lattice to represent the domain. Depending on the number of cells that represent the domain accuracy of the results will vary. Not only is needed to know when to stop iterating (previous section), but also which resolution apply to the domain to obtain enough accurate results.

To find resolution needed, several resolutions are tested and compared with the exact analytical solution. The model is run for the following resolutions: 600, 800, 1000, 1200, 1400 and 3000. For all of them, the code stops iterating when error between consecutive iterations is smaller than $5{\cdot}10^{-8}$. Figure 6 shows that results are closer to the exact solution when resolution increases.

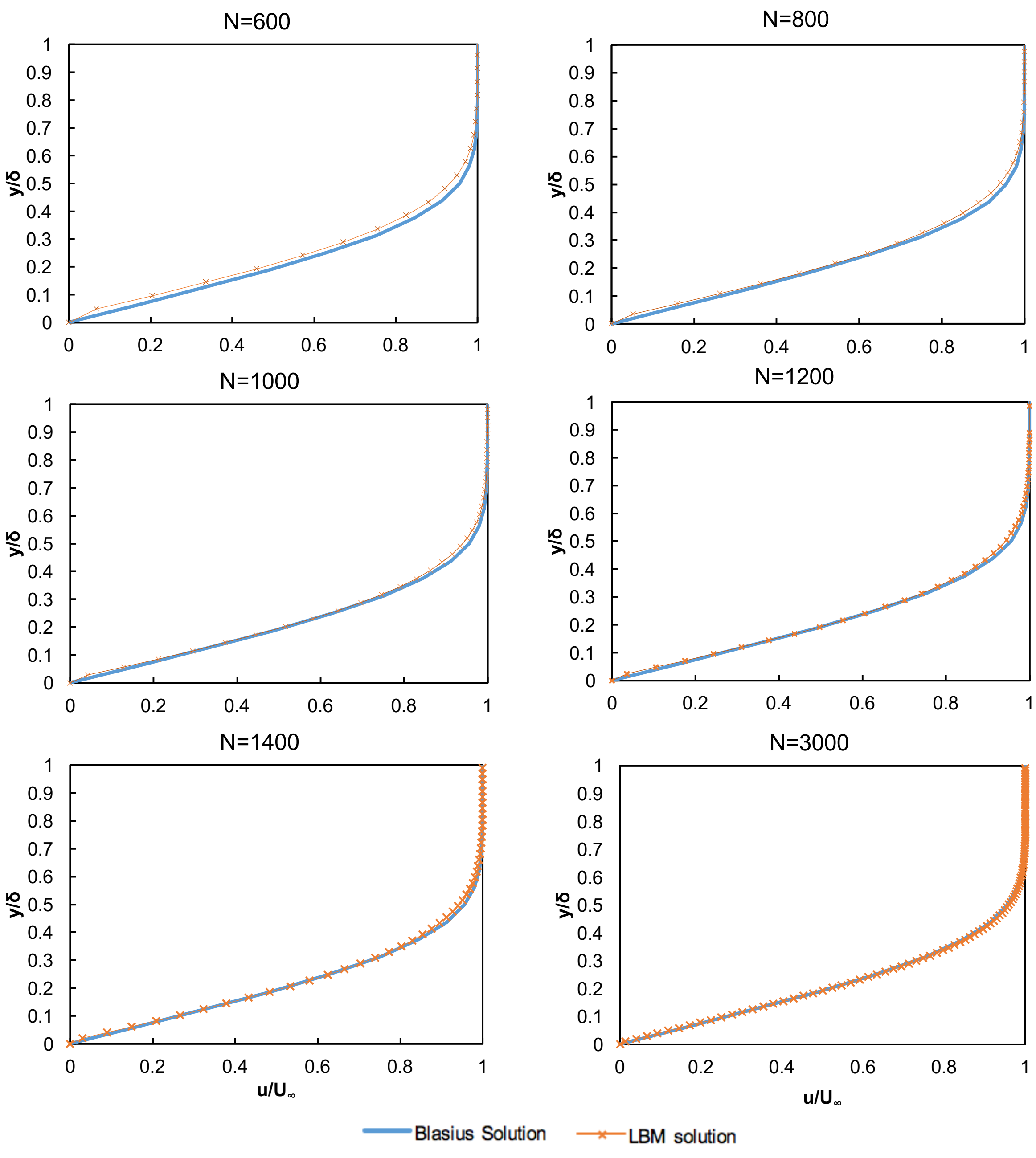


Figure 6 Blasius solution vs LBM solution for several resolutions

Figure 6 just illustrates qualitatively that higher resolution means more accurate results. However, an analysis studying quantitavely resolution against accuracy is developed. Figure 7 shows how the error between exact and LBM solution is related to resolution (number of cells per meter). To find the exact correlation between both parameters, both the resolution and the error are plotted with logarithm. Error is calculated as the average percentage difference in velocity between both solutions.

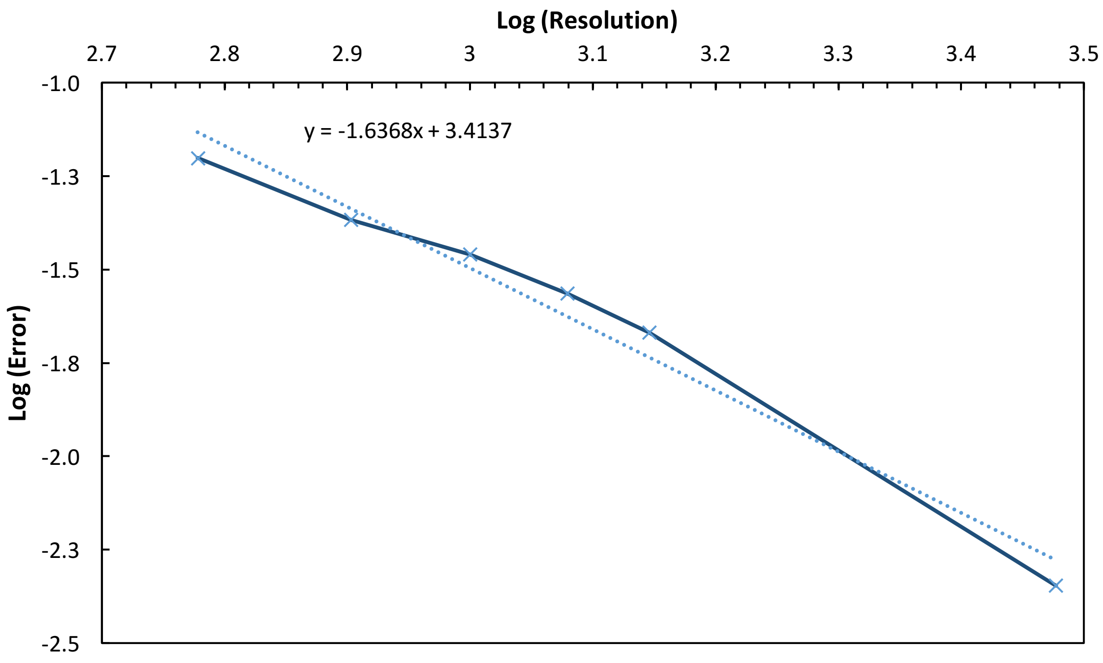


Figure 7 Error between exact and LBM solution per resolution

Figure 7 shows that correlation between the error and the resolution is

$$Error = 2592.4 \cdot (Resolution)^{-1.6368} \quad . \tag{22}$$

Therefore, error decreases when resolution increases by an exponent of 1.64. If user needs an error smaller than 1%, resolution should be higher than 1400. By introducing a resolution of 3000, accuracy in results are quite satisfactory (error of 0.45%).

4.1.1.3 *Computational cost*

Previous section has proved that higher resolution means more accuracy. However, higher resolution needs from more computational time, which is a cost. User may be restricted by this cost, so resolution may be fixed not by accuracy needed, but by maximum cost allowed. Therefore, an analysis studying how computational cost increases as resolution increases is performed in Figure 8.

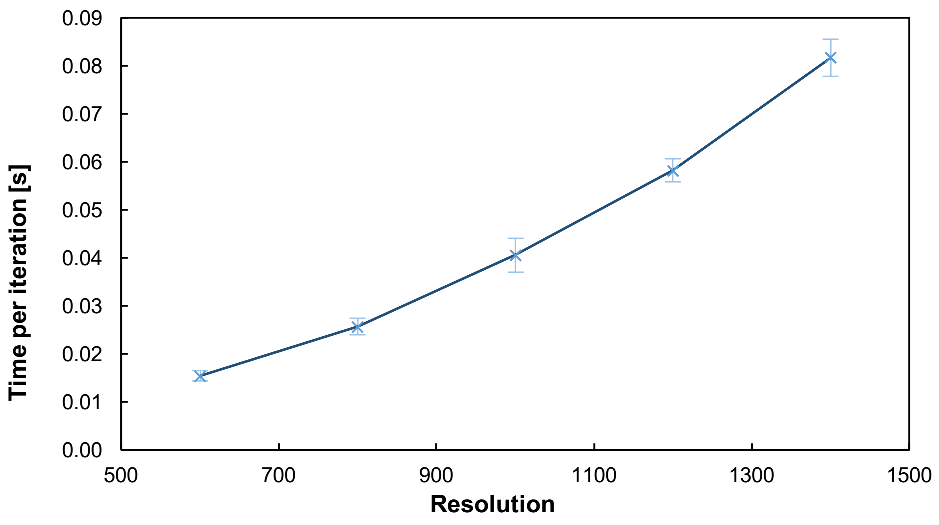


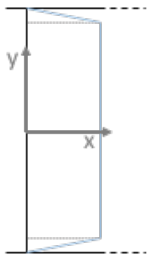
Figure 8 Computational cost per resolution

Correlation between resolution and computational cost is quadratic. Moreover, resolution of 1400 spends 0.0817 s/iteration (27.2 minutes for 20,000 iterations), while resolution of 600 only 0.01537 s/iteration (5.1 minutes for 20,000 iterations), which means that increasing approximately 5 times the computational cost leads to reducing approximately 3 times the error between solution given and exact solution.

4.1.2 Poiseuille flow

This case consists of a flow through two parallel plates. The flow moves horizontally due to a pressure decrease between both extremes of the channel (pressure-driven flow). The fluid has a Reynolds number of 5, a kinematic viscosity of 0.2 m^2/s and a density of 1 kg/m^3. The domain has a length of 3 metres and a height of 1m, while resolution varies from 60 to 300 cells per meter. Since Reynolds number is much lower than for the Blasius flat plate case, resolution needed is much smaller, because there is no need to increase resolution to avoid stability problems. Pressure gradient is fixed in 144 Pa/m.

To avoid initial problems near the walls, the case is initialised with a constant velocity for all the cells, except for those cells whose distance to the wall is smaller than 0.2m. For those cells, velocity decreases linearly from that constant velocity to 0, at the cell in the wall. Figure 9 shows how the lattice is initialised in terms of velocity, while Figure 10 shows the velocity profile at steady state.



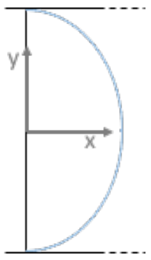


Figure 9 Initial velocity profile *Figure 10 Final velocity profile*

The geometry is implemented with periodic boundaries in the horizontal axis. This means that the right side is connected to the left side, so that the flow always remains in the domain. Pressure gradient is imposed by fixing a pressure both in the left wall and the right wall.

As for the Blasius flat plate case, the analytical solution for the Poiseuille flow has been well documented. That exact solution proves that the flow has a parabolic velocity profile, which remains constant in the whole channel. The equation of the parabola reads as

$$u(h) = \frac{\Delta P}{4 \cdot v \cdot L} \cdot (H^2 - h^2) \ , \tag{23}$$

where v is the kinematic viscosity, L is the length of the channel, ΔP is the pressure difference and H is the height of the channel.

4.1.2.1 Iteration convergence

Unlike Blasius flat plate case, Poiseuille flow is run under double precision, which means that machine epsilon is of the order of 10^{-12}. Therefore, the minimum error between two consecutive iterations permitted to continue iterating will have a different value. To determine it, the same procedure as the one of Blasius flat plate is done. Firstly, the error between consecutive iterations for a fixed amount of iterations is studied (Figure 11) and secondly the error between LBM results and exact solution as number of iterations increase is calculated (Figure 12 and Figure 13). The error of the iteration at which its solution is like the exact one will be the minimum error permitted to continue iterating.

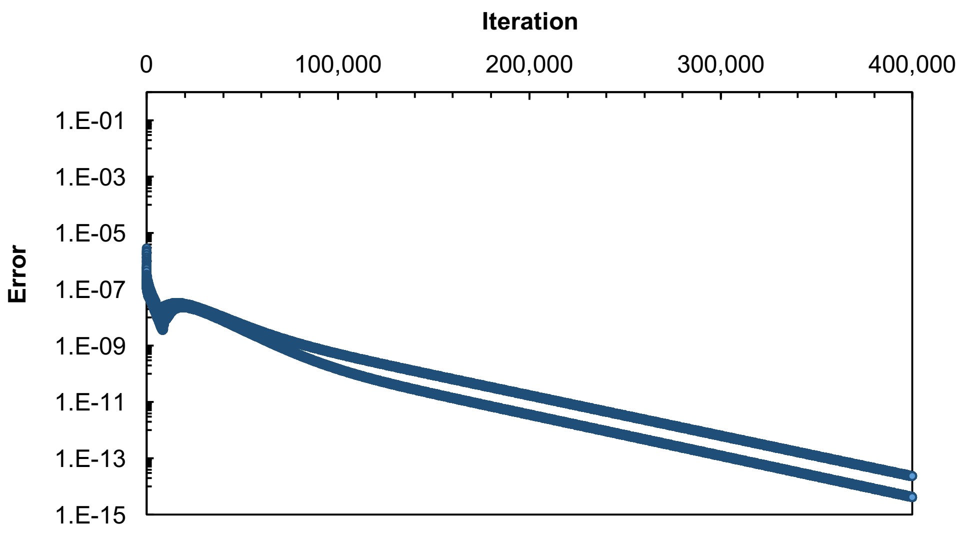


Figure 11 Error between consecutive iterations

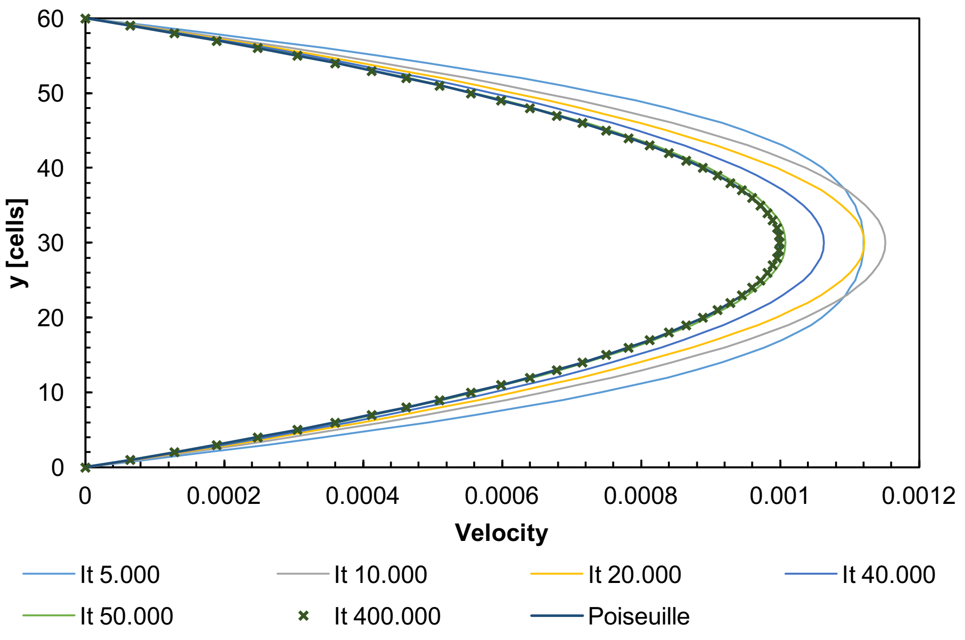


Figure 12 Velocity profile per iteration

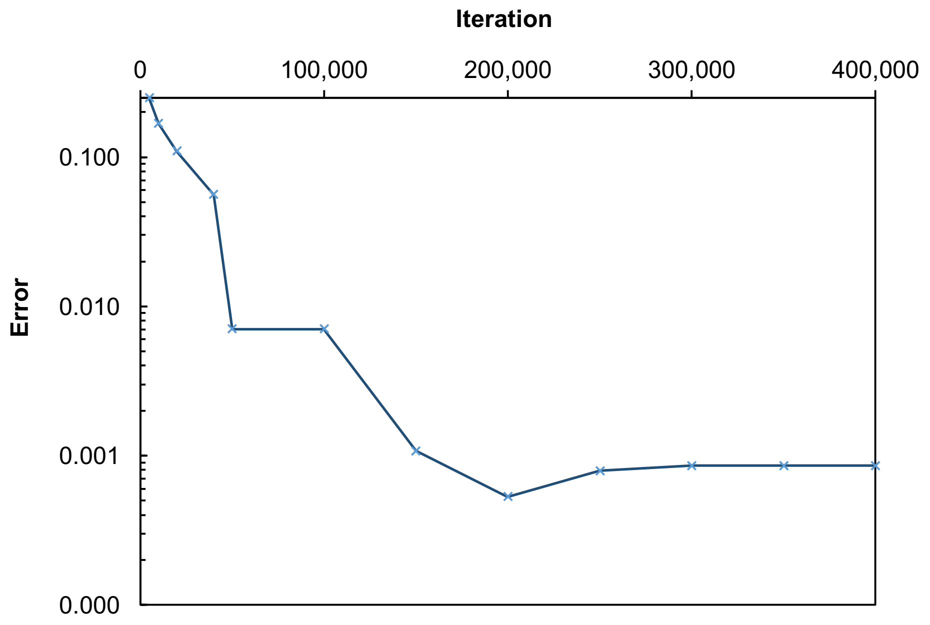


Figure 13 Error between exact and LBM solution per iteration

Error decreases notoriously until iteration 50,000 (error of 0.70%). However, it is not until iteration 200,000 that the error remains almost constant at a value of 0.05%. At iteration 200,000 error between consecutive iterations is approximately 10^{-10}. Therefore, the code needs to stop iterating when the error is smaller than 10^{-10}. The fact that minimum error permitted is smaller for the Poiseuille flow than for the Blasius flat plate is consistent with the fact that Poiseuille flow is run under double precision, while Blasius flat plate under single precision.

4.1.2.2 Mesh convergence

Iteration convergence was run with a lattice resolution of 60 cells per meter, which means that the domain is divided into 60 cells in vertical axis and into 180 cells in horizontal axis (Δx remains constant both in vertical and horizontal, since the lattice is represented by square cells). However, accuracy of the results provided may be also determined by the resolution employed. To see how resolution influences the results that LBM provides, several resolutions are tested.

Resolutions tested are 30, 60, 120, 240 and they are run until the average variation of velocity between iterations is smaller than 10^{-10}. Figure 14 shows the error between the solution provided and the exact solution. The resolution and the error are plotted in logarithms to find the exponent that correlate both parameters.

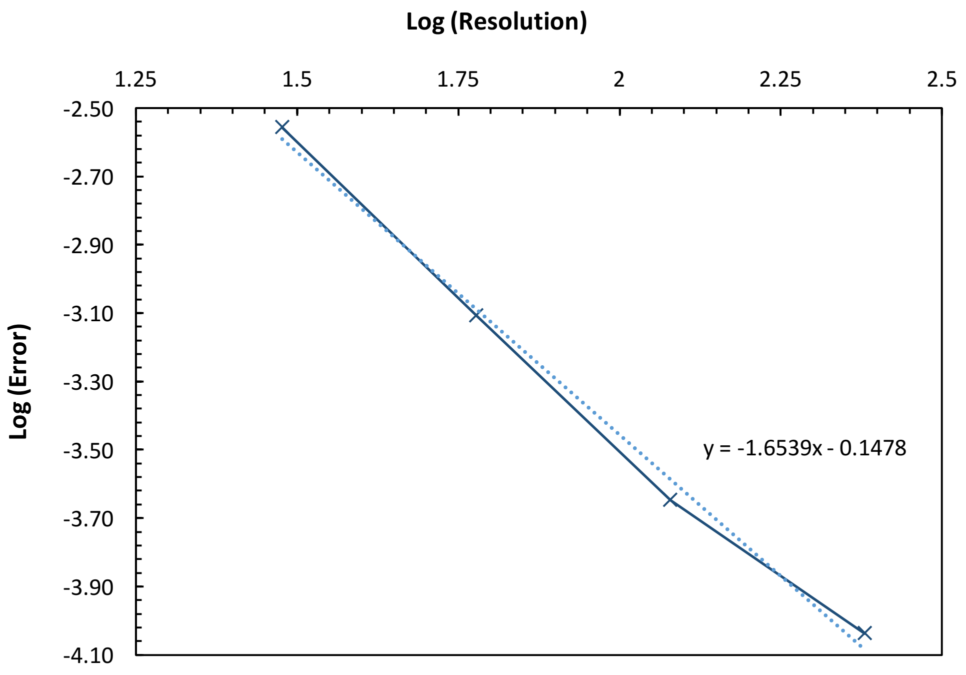


Figure 14 Error between exact and LBM solution per resolution

Figure 14 shows that correlation between the error and the resolution is

$$Error = 0.712 \cdot (Resolution)^{-1.6539} \quad . \tag{24}$$

Therefore, error decreases when resolution increases by an exponent of 1.65. For the Blasius flat plate case, this exponent is 1.64 (see equation (22)). Thus, for both cases, a modification in the resolution affects in the same way to the error. An error of only 0.01% is achieved when using a resolution of 240.

However, not all resolutions are run the same number of iterations, since the error being smaller than 10^{-10} can occur at different iteration for different resolutions (Figure 15).

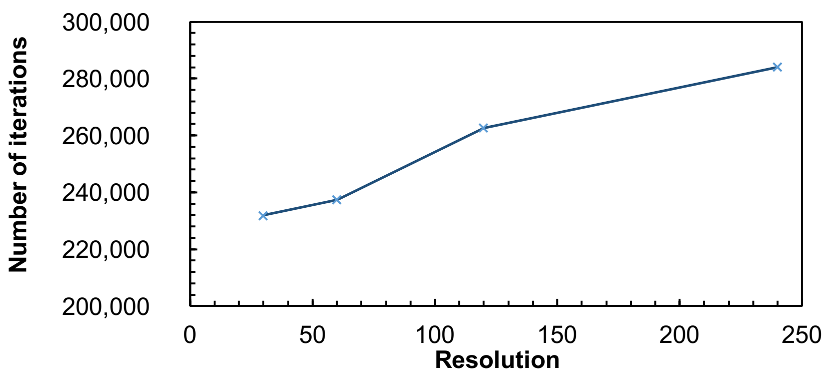


Figure 15 Number of iterations needed to achieve steady state

Figure 15 shows that the number of iterations needed increase as the resolution increases. Computational cost not only will be higher because at each iteration more cells need to be checked, but also because more iterations are run.

4.1.2.3 Influence of the relaxation parameter

For the Poiseuille flow case, a new analysis has been performed. In theory section, the relaxation parameter was described. The collision operator, also described in theory, depends significantly on the value of the relaxation parameter. To check how this parameter affects the accuracy of LBM, the Poiseuille flow case is tested for several values for that parameter.

The value for the relaxation parameter can only between 0.5 and 2, as described in theory. The cases selected include all possible values that the relaxation parameter can have. The Reynolds number is constant for all cases (Re=10), since Poiseuille flow is characterised by the Reynolds number. Therefore, same real example is tested even if relaxation parameter is different. Per each case, error between solution provided and analytical solution is calculated (Figure 16). To check if the relaxation parameter has an influence also in computational cost, the number of iterations and the time spent during computational is also calculated per each case (Figure 17).

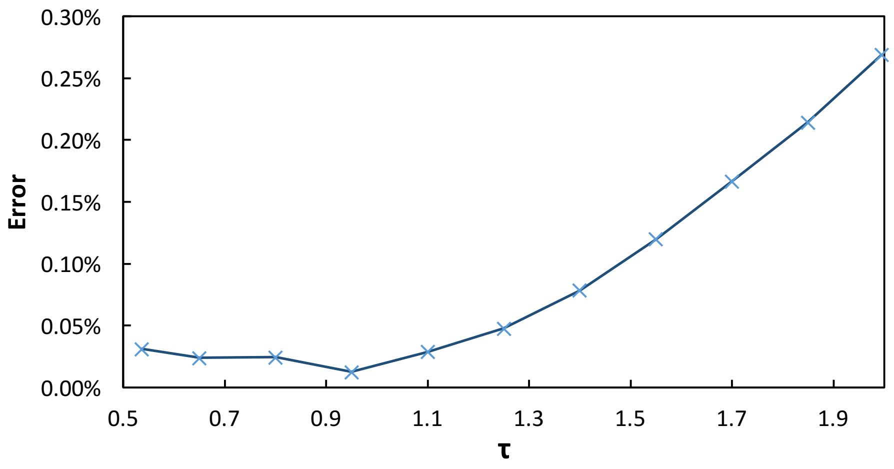


Figure 16 Error between exact and LBM solution as a function of the relaxation parameter

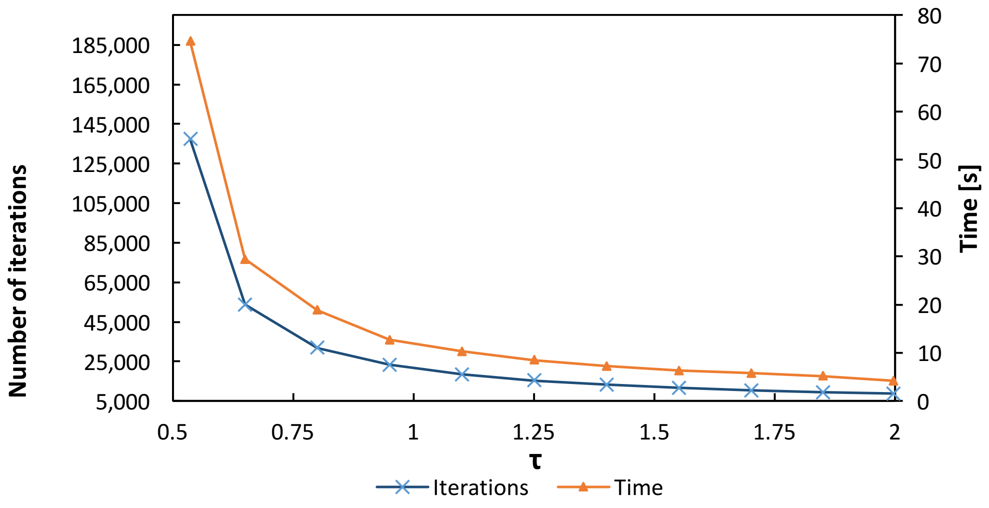


Figure 17 Computational cost as a function of the relaxation parameter

Accuracy of LBM depends on the value of the relaxation parameter, as shown in Figure 16. Best values in terms of precision are obtained with τ between 0,55 and 1,1. Moreover, not only is accuracy affected when varying relaxation parameter, but also is computational cost. Computational time decreases as relaxation time increases (Figure 17). This is because less number of iterations are needed to achieve steady state.

4.1.3 Backward facing step

The geometry of this case consists of a rectangular channel that at certain horizontal distance increases its height, creating a backward facing step. The fluid has a Reynolds number that varies from 50 to 300, a kinematic viscosity of 0.2 m^2/s and a density of 1 kg/m^3. The flow is initialised with a parabolic velocity profile in the inlet and it moves due to a decrease in pressure along the channel (pressure-driven flow).

Pressure gradient is 144 Pa/m. The domain has a length of 30m and a maximum height of 1m, while the resolution is fixed at 40 cells per meter.

This case has been well analysed and studied by scientists. However, there is not an exact analytical solution to compare it with. It can only be compared with experimental results or with other computational results. This report bases the study of the backward facing step on an extended article (Zarghami & Ahmadi, 2012) that analyses this case. The objective consists on reproducing the results obtained by this article, but using *Palabos* implementation of LBM.

The backward facing step is characterised by generating two recirculations: primary recirculation on the bottom wall after the step and secondary recirculation on the top wall beyond primary recirculation, but and only for certain Reynolds numbers (see Figure 18).

The object of study for this case is the point where primary recirculation ends, also known as reattachment point (X_1 in Figure 18).

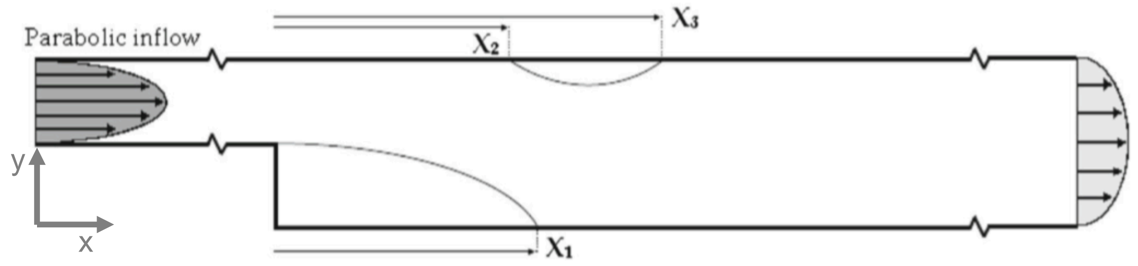


Figure 18 Geometry and recirculations of the backward facing step (Zarghami & Ahmadi, 2012)

Point X_1 changes its position as Reynolds number of the flow changes. The aim of this case is to validate that point X_1 is allocated nearby the results obtained by the article (Zarghami & Ahmadi, 2012), for different values of Reynolds number.

Point X_1 is calculated as it follows. X_1 corresponds exactly to the point where shear stress is null. For a Newtonian fluid, shear stress can be calculated as

$$\tau = \mu \cdot \frac{\partial u}{\partial y} \quad , \tag{25}$$

where u is the horizontal velocity, y is the vertical distance and μ is the dynamic viscosity of the fluid. Shear stress will be zero when the derivative is null. A linear approximation of the derivate is done to solve previous equation. Since u=0 when y=0 (boundary condition at the wall), the derivative being null is guaranteed when u=0 and y=1. Therefore, point X_1 is allocated at the same horizontal distance as that one of the cell that is just above the wall (y=1) and that has null horizontal velocity.

For the backward facing step case, Reynolds number reads as

$$Re = \frac{4 \cdot U_{max} \cdot (H-h)}{3 \cdot v} \quad , \tag{26}$$

where H is the height at the outlet, h is the height of the step, U_{max} is the maximum velocity at the inlet and v is the kinematic viscosity.

4.1.3.1 Iteration convergence

The backward facing step is run under double precision, just as the Poiseuille flow. Poiseuille flow stopped iterating when error between consecutive iterations was smaller than 10^{-10}. The same criterion may be applied for the backward facing step. To check if this criterion is accurate for that case, the error between consecutive iterations for a fixed amount of iterations is studied (Figure 11). Error is calculated as the average variation in velocity between two consecutive iterations for those cells allocated in the vertical line at a horizontal distance of the height of the step, just after the step.

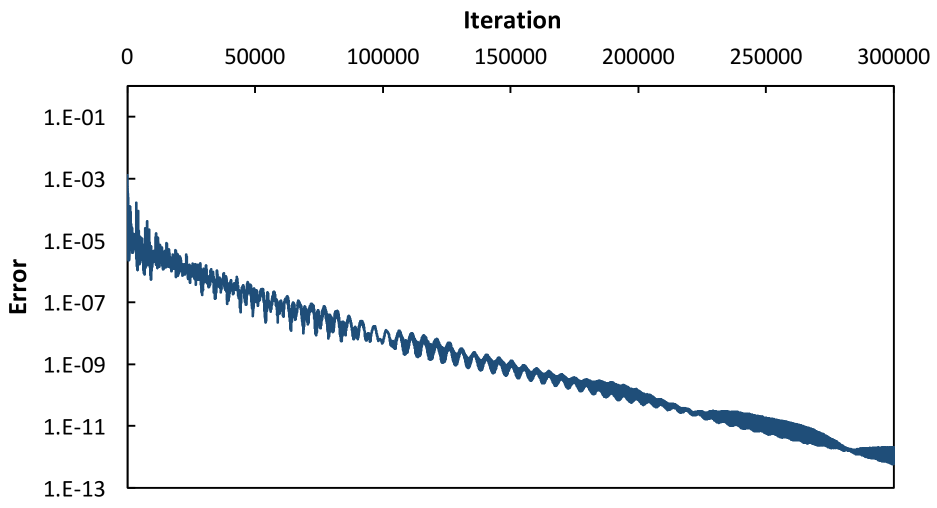


Figure 19 Error between consecutive iterations

Both Poiseuille flow curve (Figure 11) and backward facing step curve (Figure 19) for the error between consecutive iterations have the same aspect. Therefore, the same criterion of convergence as that one of the Poiseuille flow can be established. Steady state can be considered when variation in velocities between two consecutive iterations is smaller than 10^{-10}.

4.1.3.2 Mesh convergence

As in previous cases, a study regarding size of the mesh is done to evaluate the influence resolution has in results. This will tell us which resolution should be implemented. The case is tested for different number of resolutions.

Resolutions studied are 20, 40, 80 and 120 cells per meter. Reynolds number is fixed at 135. Resolutions are of the same order of those ones of the Poiseuille flow because Reynolds number employed is also of the same order. Resolution needed and Reynolds number are strictly related due to stability problems, as shown in theory. Since the object of study is point X_1, the parameter studied per each resolution case is precisely this point. Per each resolution case, position of point X_1 is calculated (Figure 20).

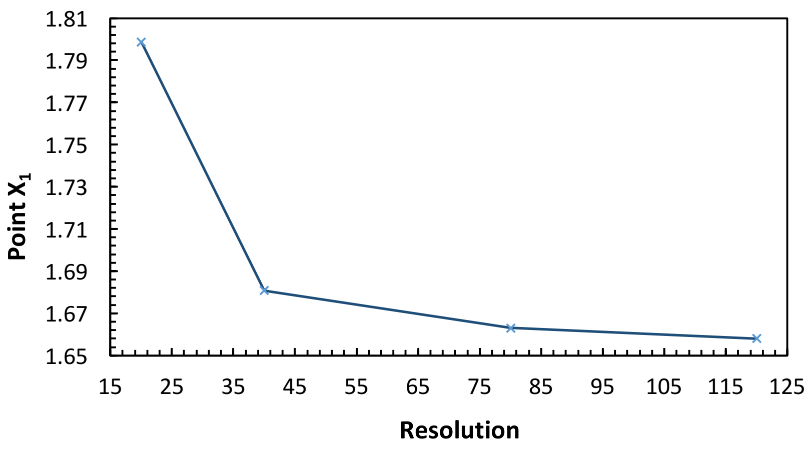


Figure 20 Position of point X_1 as a function of resolution

Figure 20 shows that for resolutions higher than 40 cells per meter, point X_1 does not change significantly its position. Therefore, a resolution of 40 cells per meter is high enough to represent accurately the position of point X_1. Using higher resolutions will lead to an increase in computational cost, as shown in Figure 21. Moreover, as for the Blasius flat plate case and the Poiseuille flow case, correlation between accuracy and resolution is quadratic.

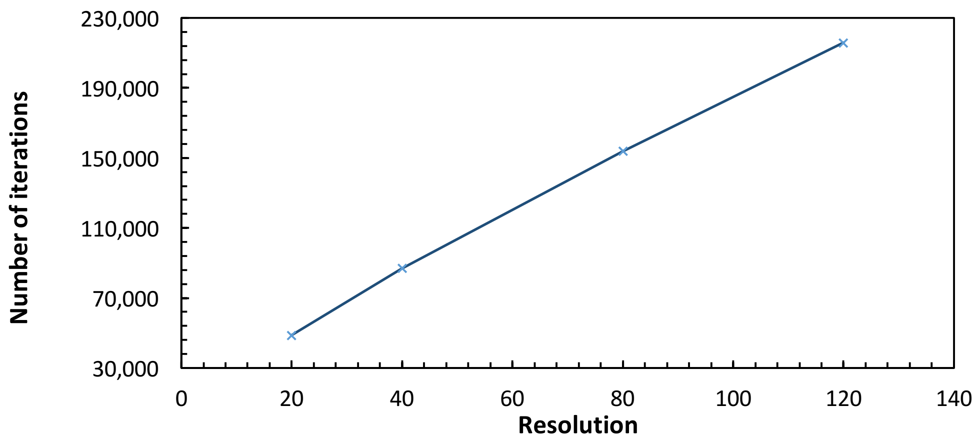


Figure 21 Number of iterations until steady state

4.1.3.3 Verification analysis

Once the criterion of convergence and number of cells per meter are established, the code is ready to be verified. Verification analysis consists on a comparison between a reference (Zarghami & Ahmadi, 2012) and the solution given by *Palabos* code. This comparison is based on studying the reattachment point (X_1) as a function of the Reynolds number. Figure 22 shows that *Palabos* LBM provides similar results as the ones obtained by the reference (Zarghami & Ahmadi, 2012). This means that both calculation of the reattachment point and the method employed to simulate the flow are accurate.

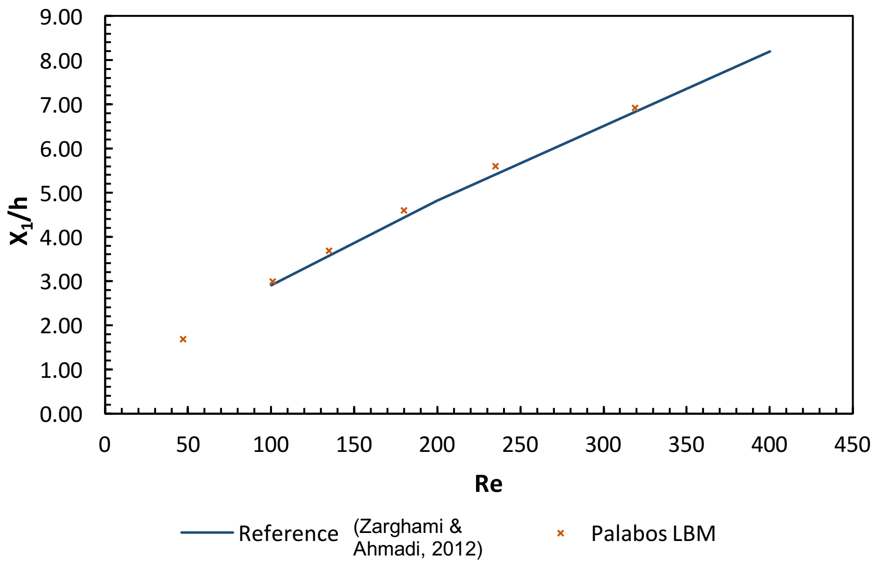


Figure 22 Comparison between reference (Zarghami & Ahmadi, 2012) and Palabos LBM for the backward facing step

4.2 Curved geometries

The previous section has shown that LBM is able to simulate accurately flows on Cartesian geometries, such as the Blasius flat plate, the Poiseuille flow and the backward facing step. The next step is to introduce curved geometries that do not necessarily match with the cartesian lattice. This will introduce a complexity to the model and I want to verify whether Lattice Boltzmann Method will accurately simulate flows around curved geometries: a cylinder and a NACA 0012 airfoil.

4.2.1 Bounce-back boundary condition

Boundary conditions are normally established in LBM by imposing certain value to a parameter in a node of the lattice. However, when geometries do not fit in the lattice the so-called bounce-back boundary condition, (Bill Bao & Meskas, 2011) and (Succi et al., 2010), permits to account for the curvature of the boundary.

The main idea of this technique is to reverse those distribution functions that when moving from one node to the final node they cross a wall on the way (see Figure 23). By reversing it is understood to change to the opposite direction. That way of implementing boundary conditions is called mid-grid bounce-back, as the wall is placed beyond two nodes. bounce-back can also be implemented directly on one node, when no-slip condition is established. Using the technique illustrated in Figure 23, it is possible to implement boundary conditions for curved geometries that do not fit in the lattice.

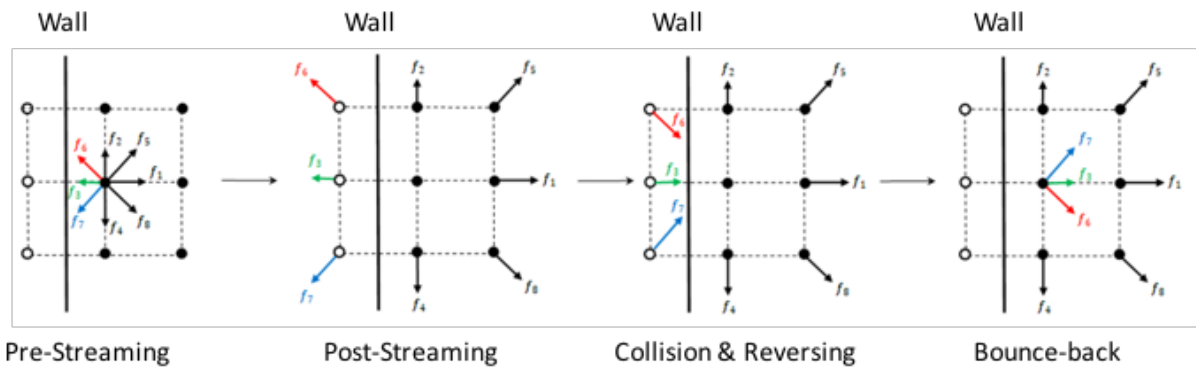


Figure 23 Illustration of mid-grid bounce-back (Bill Bao & Meskas, 2011)

4.2.2 Drag coefficient calculation

To analyse the accuracy of the results, a parameter to compare with other experimental or CFD solutions is needed. The parameter selected is the drag coefficient, which can be defined as the resistance that the cylinder causes to the flow. In practice, it is a function of the sum of all the forces acting in the object by the fluid, fluid density, inlet velocity and a characteristic length of the object in the direction of the incoming flow. Drag coefficient (C_d) is

$$C_d = \frac{2 \cdot F_x}{\rho \cdot U_{inlet}^2 \cdot d} \ . \tag{27}$$

All parameters that appear in equation (25) are already known except from F_x, which refers to the horizontal force acting in the object caused by the flow. Its value should be obtained from LBM simulation and there are several ways to achieve it. Article (Mei & Yu, 2002) points out three methods: *second-order accurate no-slip boundary condition for curved geometries*, *force evaluation based on stress integration* and *method based on the momentum exchange*. In this project, I use the method related to momentum exchange because it is the one that best fits with *Palabos* code. It calculates the force as

$$F_x = \sum_{all\ x_b} \sum_{\alpha \neq 0} e_\alpha \left[\widetilde{f_\alpha}(x_b, t) + \widetilde{f_{\bar{\alpha}}}(x_b + e_{\bar{\alpha}} \cdot \delta_t, t) \right] \cdot \left[1 - w(x_b + e_{\bar{\alpha}} \cdot \delta_t) \right] \ , \tag{28}$$

where $w(\vec{x})$ is a new function that takes the value of 0 if a lattice site $\vec{x}$ is occupied by the fluid and takes the value of 1 if a lattice site $\vec{x}$ is inside an object. X_b refers to all boundary nodes and α is used to identify the discrete velocity selected ($\bar{\alpha}$ denotes opposite direction of α).

With equation (26), this method is calculating the force as a function of the momentum exchange between boundary nodes and their adjacent fluid nodes. That is calculated by analysing population moving from a boundary node to the fluid node and vice versa, computed for all boundary nodes. It takes advantage of one of the main properties of LBM, which enables it to calculate macroscopic parameters from the distribution function as was pointed out in theory (equation (16)). It is subsequently calculated during the streaming process of LBM (Mei & Yu, 2002).

4.2.3 Flow around a cylinder

This case consists of a flow around a cylinder without external walls. Since the case is studied in 2 dimensions, the cylinder is represented as a circle. Inlet velocity is imposed on the left wall and an outflow condition on the right wall. The fluid has a Reynolds number that varies from 0.38 to 50, an inlet velocity of 0.01 m/s, a kinematic viscosity that varies from 0.00156 to 0.000012 m^2/s and a density of 1 kg/m^3. The domain has a length of 1m and a height also of 1m, while the circle has a diameter of 0.06m and it is allocated at 0.2m from the left wall at a medium height.

Resolution is fixed at 200 cells per meter. For that case, mesh convergence is not done anymore. Previous cases have shown that resolution is strictly related to Reynolds number. This case uses a Reynolds number of the same order as that one of the Poiseuille flow and the backward facing step, thus resolution is similar as that one of those cases.

Since the geometry is symmetric in horizontal axis, same results will be obtained on both sides of the symmetry. Therefore, by studying only half of the domain, one can obtain the results for all the domain. Making use of this technique, computational cost is significantly reduced. Figure 24 shows the case analysed and its symmetry.

Horizontal walls are stablished as outflows, meaning that the fluid may go out of the model. To reproduce the case properly, horizontal walls need to be far enough from the object to guarantee that almost no fluid escapes from the domain. By locating the walls far from the object, vertical velocity tends to zero and no fluid leaves the model. Regarding the symmetry case, mid wall is established as a free-slip wall that imposes zero vertical velocity, which is perfectly consistent with the case.

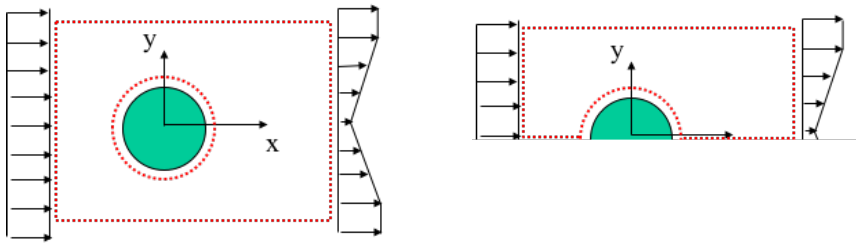


Figure 24 Geometry for the cylinder case and its symmetry

The results for that case will mainly depend on the value for the Reynolds number, which is calculated as

$$Re = \frac{U_{inlet} \cdot d}{v}, \tag{29}$$

where U_{inlet} is the inlet velocity, d is the diameter of the cylinder and v is the kinematic viscosity.

4.2.3.1 Iteration convergence

Since the case Flow around a cylinder contains curved geometries, the criterion established in cartesian geometries of when to stop iterating may be different. That is the reason why a study of the evolution of results among iterations is performed again for this case.

The procedure to determine the minimum error to continue iterating is similar to the one applied in previous cases. Firstly, the error in velocity between consecutive iterations is calculated for a fixed amount of iterations (Figure 25), to see whether this error is of the same order as that one of previous cases. Secondly, the evolution among iterations of the value for the drag coefficient is calculated. The code will stop iterating when the error in the drag coefficient between two consecutive iterations is smaller than a certain value that needs to be found. To find it, the value of the drag coefficient and its error among iterations is calculated, also for a fixed amount of iterations (Figure 26).

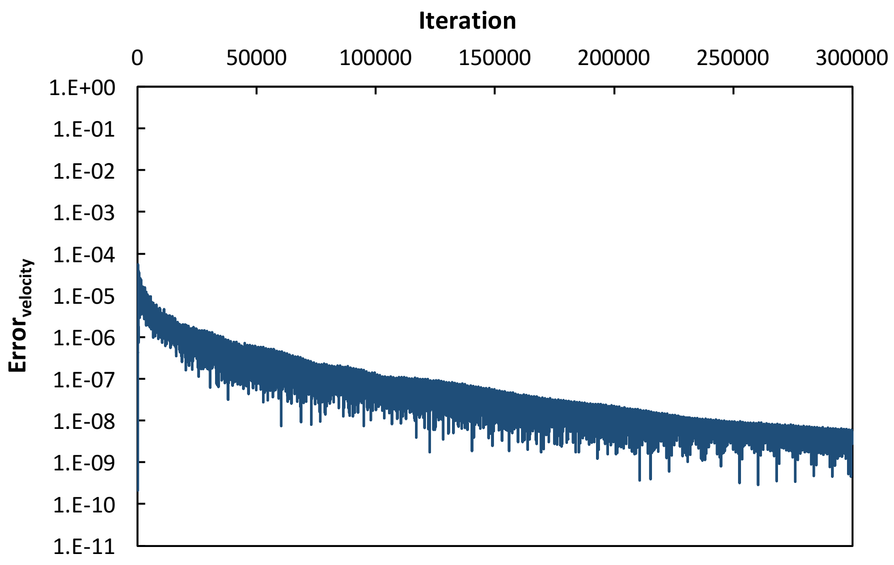


Figure 25 Error in velocity between consecutive iterations

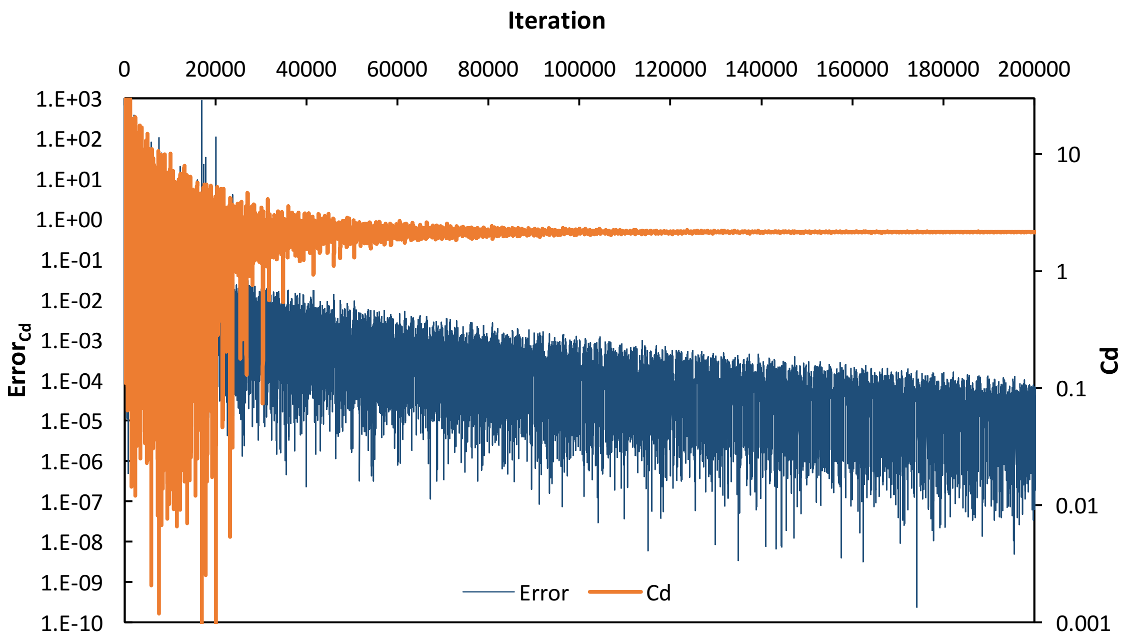


Figure 26 Evolution of the drag coefficient among iterations

Figure 25 shows that the error in velocity among iterations is higher for this case than for cartesian geometries cases (10^{-8} in the cylinder case, while 10^{-11} in cartesian geometries cases). That is because curved geometries introduce more variations between iterations to the model, due to the complexity of the geometry.

The criterion of when to stop iterating is established from Figure 26. The drag coefficient changes significantly until iteration 60,000, while it remains almost stable beyond iteration 150,000. At this iteration, error in C_d is always higher than 10^{-9}. Therefore, the code needs to stop iterating when the error in C_d between consecutive iterations is smaller than 10^{-9}.

4.2.3.2 Verification analysis

To validate that the results provided by *Palabos* in this case are accurate, the drag coefficient obtained by *Palabos* is compared to the one obtained in an experiment (Tritton, 1959). If the reader wants to know more about cylinder simulation using LBM, reference (He & Doolen, 1997) provides an in-depth analysis. Comparison is performed for different low Reynolds numbers (see Figure 27). This simulation is only performed for low Reynolds number. It could be done with higher Re, but this will imply using higher resolution which will notoriously increase computational cost. To be able to be performed in a reasonable amount of time, it should be executed on a faster machine.

Drag coefficient provided by *Palabos* is calculated using the balance momentum exchange explained above. The code stops iterating at the criterion established in the previous section (when the error in drag coefficient between two consecutive iterations is smaller than 10^{-9}).

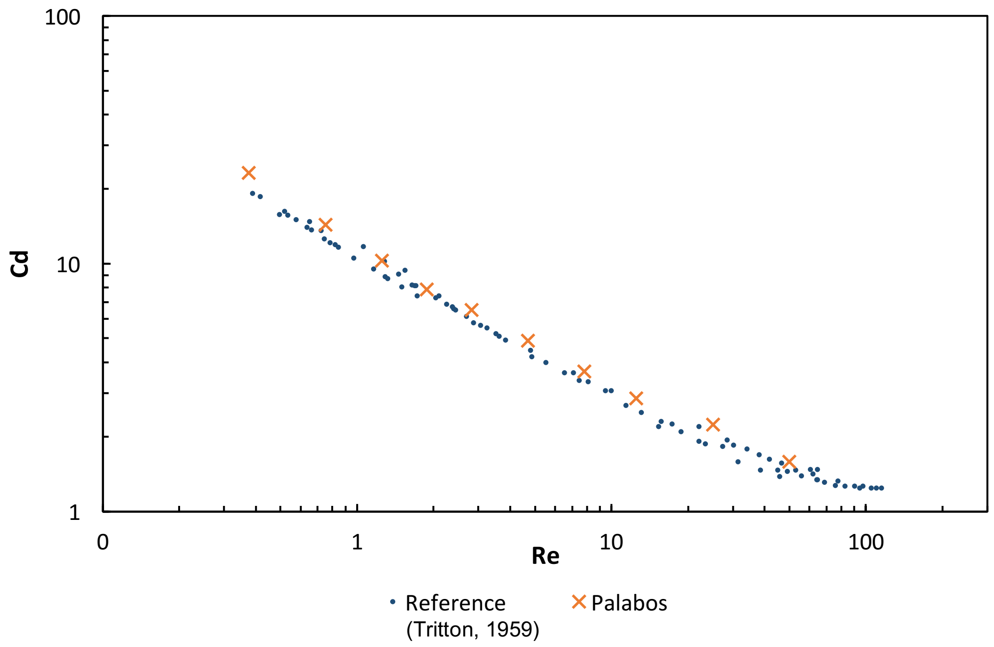


Figure 27 Comparison between the drag coefficient obtained with Palabos LBM and the one obtained in the reference (Tritton, 1959)

Figure 27 illustrates that drag coefficient obtained in *Palabos* tends to be slightly higher than the one obtained in the experiment. However, differences are small and in some cases values are very similar. Both the way of calculating drag coefficient and the implementation of bounce-back are techniques that are proved to be accurate. Therefore, *Palabos* LBM can properly reproduce the physics of the cylinder under a flow case.

4.2.4 Flow around a NACA 0012 airfoil

This case consists of a flow around an airfoil without external walls. The airfoil is a NACA 0012. The inlet velocity is imposed on the left wall and an outflow condition on the right wall. The fluid has a Reynolds number that varies from 15 to 160, an inlet velocity of 0.01 m/s, a kinematic viscosity that varies from 0.96 to 0.09 m^2/s and a density of 1 kg/m^3. The domain has a length of 1m and a height of also 1m, while the airfoil has a chord of 0.15625m and it is allocated at 0.2m from the left wall at a medium height. Resolution of the lattice is fixed at 320 cells per meter, slightly higher to the one of the cylinder, because the airfoil is a more complex geometry.

The airfoil is a NACA 0012, which belongs to the category of symmetric 4-digit NACA. First 00 indicates that it has no chamber, while 12 refers to the percentage of maximum thickness to chord. Its geometry can be represented with a specific formula, which reads as

$$y_t = 5 \cdot t \cdot c \cdot \left[0.2969 \cdot \sqrt{\frac{x}{c}} - 0.1260 \cdot \left(\frac{x}{c}\right) - 0.3516 \cdot \left(\frac{x}{c}\right)^2 + 0.2843 \cdot \left(\frac{x}{c}\right)^3 - 0.1015 \cdot \left(\frac{x}{c}\right)^4\right], \quad (30)$$

where c is the chord length, x the position along the chord, y_t half of the thickness at a given value of x and t the maximum thickness as a fraction of the chord (12/100). That equation generates the geometry shown in Figure 28.

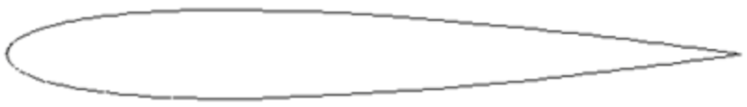


Figure 28 NACA 0012 geometry

The results for that case will mainly depend on the value for the Reynolds number, which is calculated as

$$Re = \frac{U_{inlet} \cdot c}{v}, \quad (31)$$

where U_{inlet} is the inlet velocity, c is the chord of the NACA and v is the kinematic viscosity.

4.2.4.1 Iteration convergence

For the flow around a cylinder, the model stopped iterating when error in the drag coefficient between consecutive iterations was smaller than 10^{-9}. The same criterion may be applied to the flow around a NACA, since both cases are quite similar. However, this criterion has only been tested in one case and the fact that the airfoil has a more complex geometry may affect as well the evolution of the drag coefficient among iterations. Therefore, an analysis studying how the drag coefficient evolves among iterations is performed, so as to validate that the criterion established in the flow around a cylinder may be applied to other cases.

The same procedure as the one of the cylinder is studied. Firstly, the error in velocities between two consecutive iterations (Figure 29) and, secondly, the error of the drag coefficient among iterations (Figure 30).

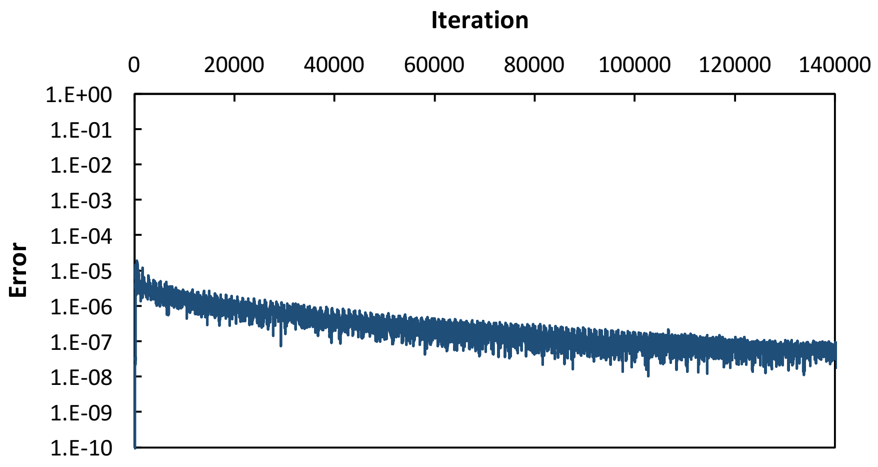


Figure 29 Error in velocity between consecutive iterations

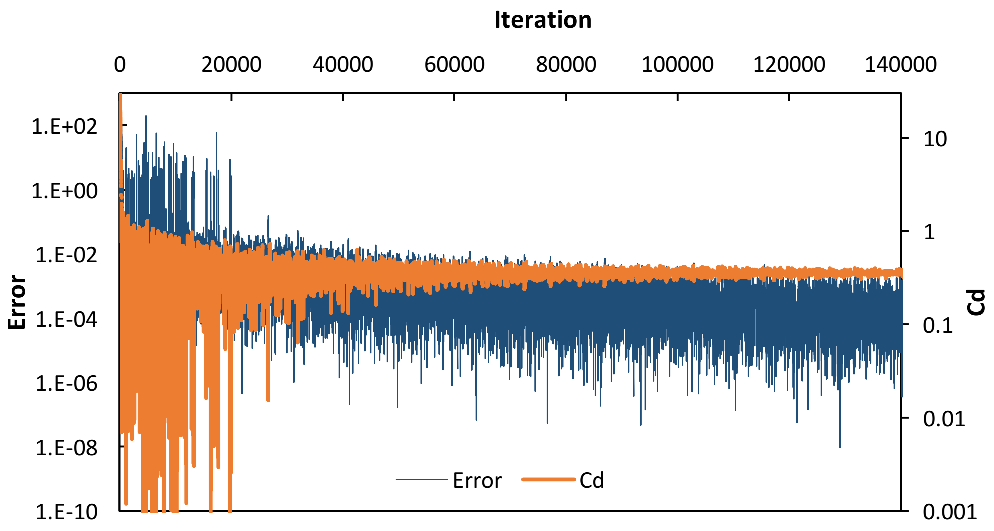


Figure 30 Evolution of the drag coefficient among iterations

The curve of the error in velocity among iterations (Figure 29) is quite similar to that of the cylinder (Figure 25). However, the curve of the error in the drag coefficient among iterations is more unstable for the airfoil (Figure 30) than for the cylinder (Figure 26). This fact is probably because the NACA is a more complex geometry that introduces complexity and variations to the model.

Even if the curves regarding the evolution of the drag coefficient among iterations do not look similar between both cases, the criterion of when to stop iterating is actually the same for both cases. Figure 30 shows that the drag coefficient does not change significantly beyond iteration 100,000. It is precisely at that iteration that the error in the drag coefficient is always smaller than 10^{-9}. Therefore, the same criterion of convergence as that of the cylinder case is established for the NACA airfoil: when variation in C_d value is smaller than 10^{-9}, results are considered steady.

4.2.4.2 Verification analysis

To validate the accuracy of the results provided by LBM, the drag coefficient obtained with *Palabos* is compared with the one obtained using another computational method (see Figure 31). XFoil (MIT, 2007) is the other computational method employed. It consists of an interactive program provided by MIT students for the design and analysis of subsonic isolated airfoils. More information about that program can be found on its website.

The cases tested are implemented with low Reynolds number flows. Higher Reynolds number may be performed as well, but this will cause incrementing resolution to avoid stability problems. By increasing resolution, computational cost increases quadratically. That is the reason why just low Reynolds number are implemented, so as to run the cases in a considerable amount of time.

The drag coefficient from *Palabos* is calculated using the momentum exchange technique, explained in the section of drag coefficient calculation. Boundary conditions are implemented with bounce-back.

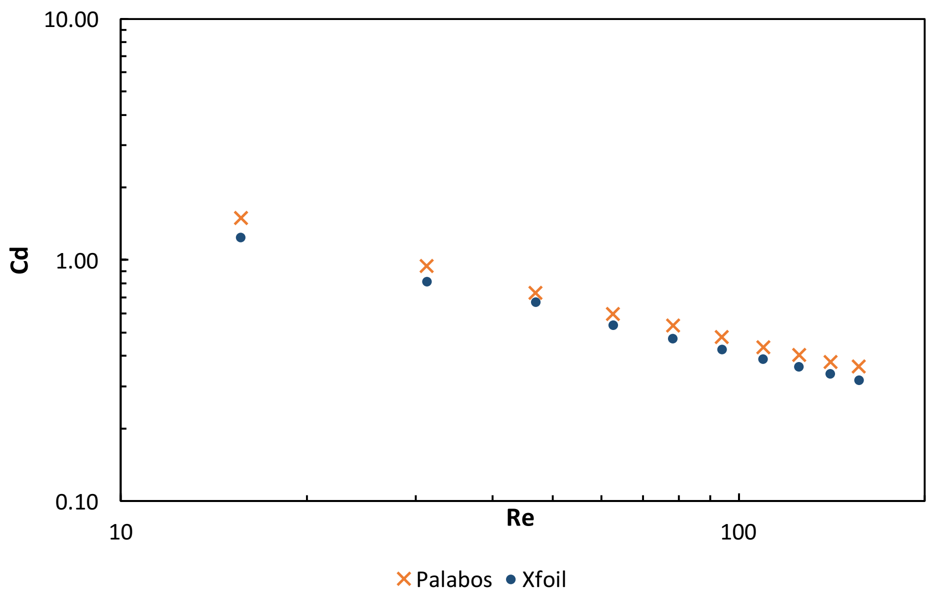


Figure 31 The drag coefficient for Palabos LBM and XFoil

Both *Palabos* and XFoil obtain similar results for the drag coefficient of the NACA 0012 for different Reynolds number. However, as in the cylinder, *Palabos* provides slightly higher values for C_d. This fact just confirms again that the way the drag coefficient is calculated and bounce-back boundary conditions are accurate. *Palabos* LBM can properly reproduce the physics of the NACA 0012 airfoil case.

5. MULTI-RELAXATION-TIME LBM

All examples tested in the previous chapter were using the Bhatnagar-Gross-Krook Boltzmann Method (LBGK or Single-Relaxation-Time LBM), which is the most popular implementation of Lattice Boltzmann Method. It has proven to be an accurate model to simulate a variety of flows, since its numerical results agree well with analytical solutions or other existing solutions. However, it has also been observed that it may suffer from numerical instability when relaxation parameter approaches to its stability limit (0.5). This section presents a new implementation of Lattice Boltzmann Method that shows better numerical stability: the Multi-Relaxation-Time Lattice Boltzmann Method (MRT-LBM). Here a theoretical introduction to this method is developed and a computational case is performed to explore in depth the results provided by this method.

5.1 Theory of MRT-LBM

This section describes theoretically the main idea of Multi-Relaxation-Time LBM, if reader wants to know more details about this model, papers (Du a et al., 2006) and (d’Humières et al., 2002) provide a complete analysis of MRT-LBM.

Multi-Relaxation-Time LBM evolves from LBGK. LBGK is a Lattice Boltzmann Method that represents the collision operator (right-hand side of the equation of the model) as a relaxation towards an equilibrium. In the first section of the project, I explained that this equilibrium is based on the Maxwellian distribution equilibrium function. In the theory section, I saw that the LBGK or Single-Relaxation-Time-LBM equation reads as

$$f_\alpha\,(\vec{x}+\vec{v_\alpha}\cdot dt, t+dt) - f_\alpha\,(\vec{x_\iota}, t) = \frac{1}{\tau}\cdot\left(f_\alpha\,(\vec{x},t) - {f^{eq}}_\alpha\,(\vec{x},t)\right), \tag{32}$$

for all discrete direction velocities α. The α equations can be written in a single equation by using matrixes. Equation (30) then evolves to

$$\boldsymbol{f}\;(\vec{x}+\vec{v_\alpha}\cdot dt, t+dt) - \boldsymbol{f}\;(\vec{x},t) = S\cdot\left(\boldsymbol{f}\;(\vec{x},t) - \boldsymbol{f^{eq}}\;(\vec{x},t)\right), \tag{33}$$

where $\boldsymbol{f}\;(\vec{x},t) = (f_0\,(\vec{x},t),\, f_1\,(\vec{x},t), \dots, f_N\,(\vec{x},t))^T$, being N the number of discrete velocities minus 1 (8 for D2Q9). S is a diagonal matrix such that S=$\frac{1}{\tau}\cdot I$, where I is the identity matrix.

This model can be modified to achieve better numerical stability. This can be done by introducing different relaxation parameters in the collision operator matrix (S), since the relaxation parameters correspond to the inverse of the eigenvalues of matrix S. The (N+1) eigenvalues of S are all between 0 and 2 to maintain linear stability and the separation of scales, which means that the relaxation times of non-conserved quantities are much faster than the hydrodynamic time-scales, which in this case are mass density and momentum (d’Humières et al., 2002).

This fact allows to introduce optional relaxation parameters without any kind of influence in the physics of the model, while in LBGK case all relaxation parameters are equal and related with the kinematic viscosity of the fluid. Introducing some optional relaxation parameter result in better numerical stability. In MRT-LBM, the only restriction is that

$$\tau_\alpha = \frac{v}{c_s^2} + {}^1/_2 \qquad (\alpha = 7{,}8)\ , \tag{34}$$

where v refers to kinematic viscosity and c_s to speed of sound (${}^1/_{\sqrt{3}}$ for D2Q9). This means that only two of the nine relaxation parameters are related to the kinematic viscosity, while in LBGK all of them are related to the kinematic viscosity. Also, the relaxation parameter for $\alpha = 0$ can have any value, since it has no influence in the MRT-LBM model. This allows to introduce a value that benefit the stability of the model.

When introducing high Reynolds number, kinematic viscosity decreases considerably, corresponding in a value for the relaxation parameter closer to 0.5 (stability limit). If not all relaxation parameters have this value, stability is notably improved. Therefore, MRT-LBM enables implementing better higher Reynolds number as will be tested in the following computational case.

5.2 MRT-LBM case

An analysis comparing Multi-Relaxation-Time LBM and Single-Relaxation-Time LBM is performed in this section of the project. That analysis consists of reproducing the same case with both models and studying differences in results. The case tested is the Poiseuille Flow, which consists of a flow through two parallel plates. The flow moves horizontally due to a pressure decrease between both extremes of the channel (pressure-driven flow). The fluid has Reynolds number that varies from 2 to 1200 and a density of 1 kg/m^3. The domain has a length of 3 metres and a height of 1m, while resolution is fixed in 32 cells per meter. Pressure gradient is fixed in 144 Pa/m. Code is run until variation in velocity profile between two consecutive iterations is smaller than 10^{-10}.

Both cases are tested for different Reynolds number, which is strictly related to the relaxation parameter by

$$Re = \frac{N \cdot u}{(\tau - 0.5)\cdot\frac{1}{3}}\ . \tag{35}$$

Therefore, when Re increases, relaxation parameter decreases because Resolution is fixed at 32 cells per meter. Thus, at some Reynolds number, the model will become unstable since the relaxation parameter will be too close to its stability limit.

To analyse to difference between both models, the error between exact solution and solution provided by the model is calculated, per each Re case (see Figure 32).

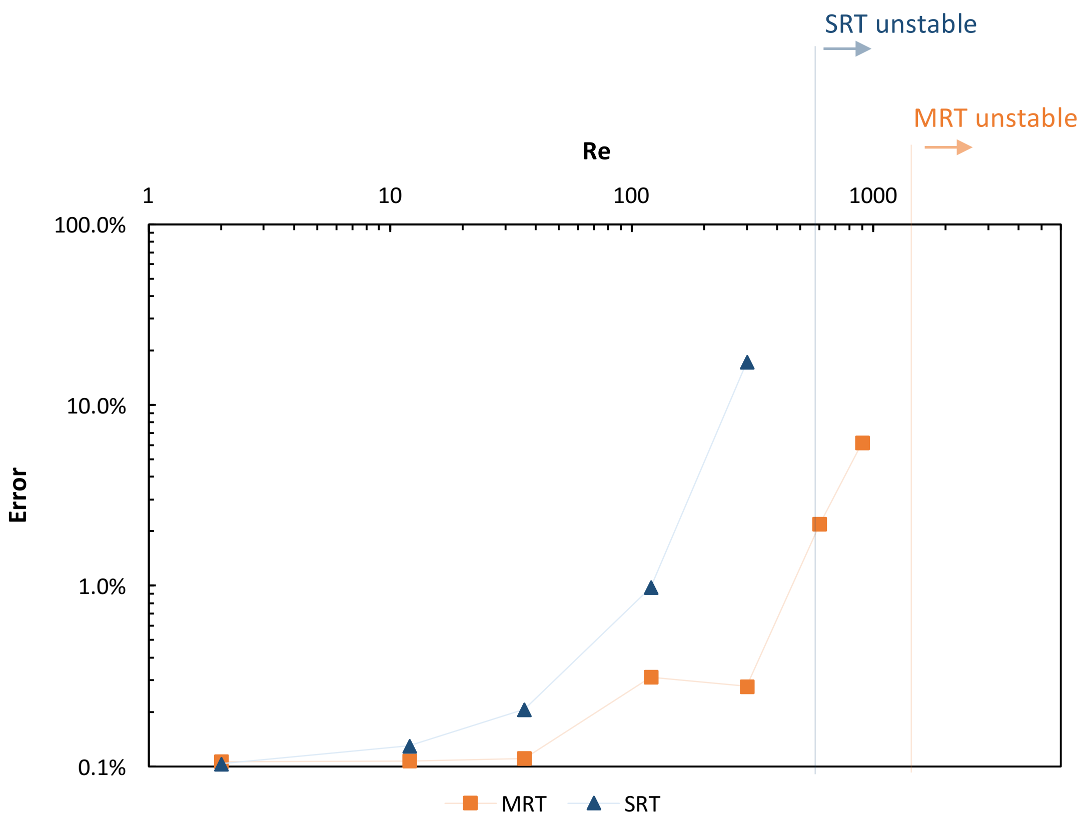


Figure 32 MRT vs SRT for Poiseuille Flow

Figure 32 shows that Multi-Relaxation-Time has lower stability problems than Single-Relaxation-Time. MRT becomes unstable when Re=1200, while SRT when Re=600 (τ = 0.50016). This means that MRT enables the implementation of higher Reynolds number than SRT, maintaining constant resolution. It should also be noted that Single-Relaxation-Time is not able to achieve stability for values of relaxation parameter smaller than 0.5002. However, for low Reynolds number, both models provide a similar level of accuracy.

Nevertheless, Multi-Relaxation-Time LBM has a disadvantage when compared to Single-Relaxation-Time LBM. MRT-LBM needs more iterations to achieve steady state and it also spend more time per iteration due to the complexity of the model. Therefore, computational cost is higher.

To avoid instabilities in the Single-Relaxation-Time LBM, one can increase the resolution to increase the relaxation parameter. This will lead also to an increase in computational cost. Therefore, to analyse which option is better in terms of computational cost a new analysis is done. It consists on calculating the computational cost for both models to achieve an error of 0.5% for several Reynolds number (see Figure 33). SRT will need more resolution as Re increases, while MRT will need a smaller resolution but it will spend more time per iteration and it will run more iterations.

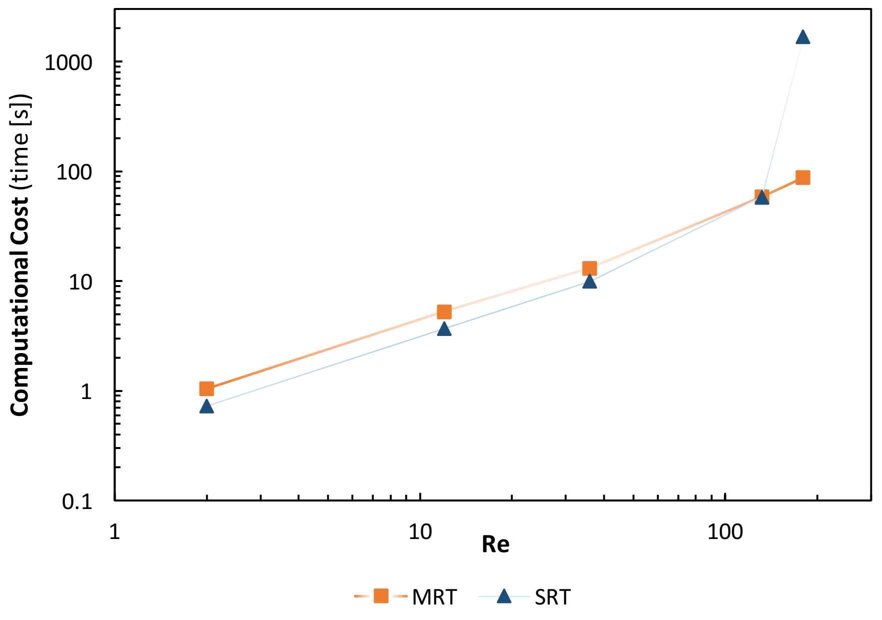


Figure 33 Computational cost for several Reynolds number for SRT and MRT to achieve an error of 0.5%

Figure 33 shows that SRT needs from less computational cost to achieve an error of 0.5% than MRT needs. For the case of Re=132, even if SRT needs a resolution of 37 and MRT only of 23, SRT stills has less computational cost. However, for higher Reynolds number, the increase in resolution needed by SRT is too high to maintain less computational cost than MRT. Therefore, for higher Re, Multi-Relaxation-Time LBM is a better option than Single-Relaxation-Time.

6. CONCLUSIONS

During the development of this research project, some remarkable conclusions regarding LBM have been obtained. The main conclusions of the project are the following.

Lattice Boltzmann Method can reproduce flow simulations for incompressible fluids accurately. For all cases tested, both staircase and curved geometries, it has been possible to match the solution provided by LBM with another analytical or computational solution. For the Blasius flat plate, the Poiseuille flow and the backward facing step the minimum error (average difference for all points between exact and solution provided by the model) is smaller than 0.5%. Regarding curved geometries, cylinder had an error of almost null, which also proved that bounce-back boundary conditions enables to properly represent curved geometries. For the NACA 0012 airfoil, the error was closer to 8% in comparison with XFoil results.

To avoid instabilities in LBM at high Reynolds numbers the resolution (number of cells per meter) should be increased. That is to elude values for the relaxation parameter closer to its stability limit of 0.5. Lattice velocity cannot be increased as it also has an upper limit. Therefore, stability can be achieved by selecting resolution properly, that fixes Δx, and lattice velocity, that fixes Δt since Δx is already fixed with resolution.

A different implementation with respect to the one of Bhatnagar-Gross-Krook LBM obtains better numerical stability. It is the Multi-Relaxation-Time LBM, which can implement flows with two times the upper Reynolds number limit for Single-Relaxation-Time LBM (SRT was unstable for Re=600, while MRT for Re=1200). Moreover, it is observed that SRT is unstable for values of relaxation parameter smaller than 0.5002. Computational cost is smaller in SRT even if it needs higher resolution to avoid stability problems, but only until a certain value for the Reynolds number.

Correlation between error and resolution has been proved to be quadratic, being the error the average percentage difference between the existing solution and the solution provided by the model. Both the Blasius flat plate and the Poiseuille flow have a correlation between the error and the resolution with an exponent of 1.6. Correlation between resolution and computational cost has been proved to be quadratic as well. In the Blasius flat plate case, when multiplying resolution per 3.5, computational cost multiplies per 5.4 and the average percentage error between the exact solution and the solution provided by the model divides per 15.5 (from 6.2% to 0,4%).

This project has established a condition for LBM to stop iterating that provides accurate results. A condition fixing a minimum error allowed between two consecutive iterations has been performed. For cartesian geometries, a minimum error of 10^{-10} is enough to achieve accurate results, while for curved geometries a minimum error of 10^{-9}.

The relaxation parameter (τ) plays an important role on the results provided by Lattice Boltzmann Method in Poiseuille flow. More accurate results are obtained for values of relaxation parameter between 0.55 and 1.1 (between those values error keeps almost constant at 0.025%, but beyond τ = 1.1 error increases until 0.27% when τ =1.95). Moreover, when this parameter is increased, less iterations are needed to achieve steady state and less stability problems occur. Therefore, the most accurate value for the relaxation parameter is 1.1.

The calculation of forces in the Lattice Boltzmann Method via the momentum exchange method has produced accurate estimates of the drag coefficient for the flow around a cylinder case and for the flow around a NACA 0012 case.

7. SUGGESTIONS FOR FURTHER WORK

There are also other aspects regarding LBM that can be analyzed. One of the main advantages of LBM is its ability to parallelize numerical calculations. Parallelization in this work has been limited to 2 processors, for convenience, an analysis of the efficiency and achievable sealing for large number of processors may be performed. Studying the implementation of multi-phase flows or the simulation of flows in 3 dimensions may also be of interest to estimate the abilities LBM can offer.

This project has focused on Lattice Boltzmann Method for incompressible flows. However, one of the main advantages of LBM relies on its ability to implement different kind of fluids just by modifying the collision operator. There are two promising models implementing compressible flows in LBM: KT-LBM (Kataoka & Tsutahara, 2004) and QU-LBM (Qu, 2009). Other references for LBM in compressible flows are (Yan et al., 1999) and (Shan et al., 2006).

As a first overview of these models, it should be noted that more discrete velocities are needed (KT-LBM uses a D2Q16 model and QU-LBM a D2Q13). However, the main problem when applying compressible flows concerns the streaming process. In incompressible flow models, the molecular velocity depends only on the cell spacing and the time step, while in compressible flow models the velocities are irregular and defining the velocity as in incompressible flow is not valid. To cope with this, these models define a new distribution equilibrium function, as well as new ways of relating macroscopic variables to the distribution function. It should be noted that those models cannot be implemented using *Palabos* code. *Palabos* only allows the execution of incompressible flows. To implement these new models, other LBM code needs to be programmed since *Palabos* structure is not able to support these models.

REFERENCES

1. Bewazeer, S., 2013. *Stability and accuracy of Lattice Boltzmann Method,* Thesis: University of Calgary (Alberta).
2. Bhatnagar, P.L., Gross, E. P. & Krook, M., 1954. A model for collision process in gases. I. Small amplitude processes in charged and neutral one-component system.. *Phys. Rev.,* 94(3), pp. 511-525.
3. Bill Bao, Y. & Meskas, J., 2011. *Lattice Boltzmann Method for Fluid Simulations,* Report No. 930: New York University.
4. C.Sukop, M. & T.Thorne, D., 2006. *Lattice Boltzmann Modeling.* 2nd ed. : Springer.
5. Chapman, S. & Cowling, T., 1992. *The Mathematical Theory of Non-Uniform Gases.* 3rd edition ed. Cambridge: Cambridge University Press.
6. Chen, S. & D.Doolen, G., 1998. Lattice Boltzmann Method for Fluid Flows. *Fluid Mechanics,* Issue 30, pp. 329-364.
7. D. Sterling, J. & Chen, S., 1996. Stability Analysis of Lattice Boltzmann Methods. *Journal of Computational Physics,* 123(1), pp. 196-206.
8. d'Humières, D. et al., 2002. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *The Royal Society,* Volume 360, pp. 437-451.
9. Du, R., Shi, B. & Chen, X., 2006. Multi-relaxation-time lattice Boltzmann model for incompressible flow. *Physics Letters A,* 359(6), pp. 564-572.
10. Fang, T., Guo, F. & F. Lee, C.-f., 2006. A note on the extended Blasius equation. *Applied Mathematical Letters,* 19(7), pp. 613-617.
11. Flowkit, L., 2011. *Palabos User Guide - Release 1.0.* [Online] Available at: http://www.palabos.org/palabos/documentation.userguide/
12. He, X. & Doolen, G., 1997. Lattice Boltzmann method on curvilinear coordiantes systems: Flow around a circular cylinder. *Journal of computational physics,* Volume 134, pp. 306-315.
13. K. Aidun, C. & R.Clausen, J., 2010. Lattice-Boltzmann Method for Complex Flows. *Fluid Mechanics,* 42(1), pp. 439-472.
14. Kataoka, T. & Tsutahara, M., 2004. Lattice Boltzmann method for the compressible Euler equations. *Physical Review E,* 69(5), p. 056702.
15. Kataoka, T. & Tsutahara, M., 2004. Lattice Boltzmann model for the compressible Navier-Stokes equations with flexible specific-heat ratio. *Physical Review,* 69(3), p. 035701.
16. Mei, R. & Yu, D. S. W., 2002. *Force Evaluation in the Lattice Boltzmann Method Involving Curved Geometry,* Report No. 2002-22: NASA, ICASE.
17. MIT, 2007. *XFoil: Subsonic Airfoil development system.* [Online] Available at: http://web.mit.edu/drela/Public/web/xfoil/
18. Qian, Y., d'Humières, D. & Lallemand, P., 1992. Lattice BGK models for Navier-Stokes equation. *Europhys,* 17(6), pp. 479-484.

19. Qu, K., 2009. *Development of lattice Boltzmann method for compressible flows,* PhD thesis: Northwestern Polytechnical University.
20. Shan, X.-W., Yuan, X.-F. & Chen, H.-D., 2006. Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation. *Journal of Fluid Mechanics,* 550(1), pp. 413-441.
21. Succi, S., 2001. *The Lattice Boltzmann Equation.* Oxford: Oxford University Press.
22. Succi, S., Sbragaglia, M. & Ubertini, S., 2010. Lattice Boltzmann Method. *Scholarpedia,* 5(5), p. 9507.
23. Tritton, D., 1959. Experiments on the flow past a circular cylinder at low Reynolds numbers. *Fluid Mechanics,* 6(4), pp. 547-567.
24. Worthing, R. A., Mozer, J. & Seeley, G., 1997. Stability of lattice Boltzmann methods in hydrodynamic regimes. *Physical Review,* 56(2), pp. 2243-2263.
25. Yan, G.-W., Chen, Y.-S. & Hu, S.-X., 1999. Simple lattice Boltzmann model for simulating flows with shock wave. *Physical Review E,* Volume 59, pp. 454-459.
26. Zarghami, A. & Ahmadi, N., 2012. A stable Lattice Boltmann Method for Steady Backward-Facing Step Flow. *Arabian Journal for Science and Engineering,* 39(8), pp. 6375-6384.