UNIVERSITY OF CALIFORNIA,
IRVINE

Lattice Boltzmann Method for Fluid Flow

THESIS

Submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Mechanical and Aerospace Engineering

by

Amine Abdellah-El-Hadj

Thesis Committee:
Professor Yun Wang
Professor Feng Liu
Professor Ahmed M.Eltawil

2014

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF PRINCIPAL SYMBOLS

Latin symbols

c_k Lattice velocity

c_s Speed of sound

D Dimension of the space

D_0 Particle number of freedom degree

$e_{\sigma i}$ Velocity unit vector

e_σ Module of $e_{\sigma i}$

f Particle distribution function

f_k^{eq} Equilibrium distribution function

g Boltzmann-Maxwellian distribution function.

k_B Boltzmann constant

L Characteristic length

N Number of lattices

N_A Avogadro's number

Re Reynolds number

R Ideal gas constant

T Temperature

t Time

U Characteristic velocity

u Axial velocity

v Vertical velocity

W_α Weight coefficient

w Relative velocity (shock tube application)

Greek symbols

$\delta_{\alpha\beta}$ Kronecker delta

ξ_α Discrete velocity

ξ Macroscopic velocity

λ Collision relaxation time

ν Lattice kinematic viscosity

ρ Macroscopic density

∂_t Time step

τ Dimensionless relaxation time

$\psi(\xi)$ Polynomial of ξ

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my committee chair, Professor Yun Wang for giving me the chance to be part of his Laboratory and to explore previously unknown territories to me, either in electrochemistry and Fuel Cells or in the lattice Boltzmann Method. He continually and convincingly conveys a spirit of adventure in the way the work is done in the Lab.

I would like to thank my committee members, Professor Ahmed M. Eltawil for his academic and research advice and Professor Feng Liu for his indispensible support and advice. He is truly a source of inspiration. They showed me that complicated problems can be made simple and easy to tackle. I would like to thank Professor Yaohong Qian, for his help in understanding the basics and the theory behind the Lattice Boltzmann Method.

I thank the department of Mechanical and Aerospace Engineering at the University of California Irvine, and especially its chair Professor Derek Dunn-Rankin and all the Professors for giving me the chance to push my limits and explore new territories. It was an amazing experience that helped me grow both at the technical and personal levels.

ABSTRACT OF THE THESIS

Lattice Boltzmann Method for Fluid Flow

By
Amine Abdellah-El-Hadj

Master of Science in Mechanical and Aerospace Engineering

University of California, Irvine, 2014

Professor Yun Wang, Chair

In the last few years, a rapid development in the method known as the Lattice Boltzmann Method (LBM) has been achieved. It demonstrated its ability to simulate hydrodynamic systems, multiphase and multicomponent fluids. The main advantages of the LBM are the parallelism of the method, the simplicity of programming and the capability of incorporating model interactions. The use of the LBM to understand the flow structure inside the Gas Diffusion Layer (GDL) of a fuel cell is a particular active topic, motivated by the need of finding alternative energy conversion devices.

In the present work we developed a rigorous initial base of a flow solver based on the LBM, the BGK model is used to approximate the collision term in the Boltzmann equation. We used the bounce back scheme to simulate the boundary conditions and the flow solver is validated against three benchmarking cases. The process of applying the boundary conditions was automated to handle complicated flow structures. We simulated the flow in a 2D structure surface extracted from a 3D reconstructed GDL, using both non-parallel and parallel code. Also the parallelism of the code is easier comparing to the parallelism of the traditional Navier-Stokes solver. The results for a single phase flow show the flow structure expected and the convergence of the parallel code is faster compared to the nonparallel code.

Introduction

One of the major concerns of the gas diffusion layer (GDL) inside a proton exchange membrane fuel cell (PEMFC) is water management. Treatment of the pores of the GDL can affect the PEMFC performance due to the degree of water flooding inside the GDL.

The main purpose of the GDL in a PEMFC is to distribute the reactants along the active surface of the electrodes. In addition, the GDL has to ensure proper transport of reactants, product water, electrons, and heat of the reaction. Porous carbon materials are most often used to accomplish this complex task. The GDL is typically comprised of carbon for electrical conductivity and PTFE for hydrophobicity.

In this work, we are aiming to develop a rigorous framework of the Lattice Boltzmann method (LBM). We are primarily interested in the single phase flow through a porous media and its application to predict the flow inside a PEMFC GDL.

In the present report we start with a bibliographic review where we introduce the Lattice Boltzmann Method (LBM) and its main features, such as the approximation of the collision term, the type of boundary conditions used and the main strategies. In the second part, we develop the lattice Boltzmann equation (LBE) starting from the Boltzmann equation in its continuous form, using the BGK model to approximate the collision term. In the computation and results part, we present the main strategy of the code and the results for various test cases. We finish this report by a general conclusion and the direction for future work.

CHAPTER 1: Bibliographic review

The Lattice Boltzmann method (LBM) is particularly successful in fluid flow applications involving interfacial dynamics and complex boundaries [1]. Unlike conventional numerical schemes based on discretization of macroscopic models and mesoscopic kinetic equations, the fundamental idea of the LBM is to construct simplified kinetic models that incorporate the essential physics of microscopic processes so that the macroscopic averaged properties obey the desired macroscopic equations. The macroscopic dynamics of a fluid is the result of the collective behavior of many microscopic particles in the system which is not sensitive to the underlying details in microscopic physics. This is why the simplified kinetic-type methods are suitable to represent the macroscopic fluid flows.

The full Boltzmann equations require huge computational capabilities to follow each fluid particle like in the molecular dynamics simulations. By developing a simplified version of the kinetic equation, one avoids solving complicated kinetic equations such as the full Boltzmann equation.

In the following paragraphs we present the different developments that lead to the Lattice Boltzmann method form.

The origin of the LBM is the lattice gas automata (LGA). The LGA is a discrete particle kinetics that uses a discrete lattice and discrete time. The LBM can also be viewed as a finite difference scheme for the kinetic equation of the discrete velocity distribution function. The idea of using the simplified kinetic equation with a single particle speed to simulate fluid flows was first proposed by Boradwell in 1964 for studying shock structures. Multispeed discrete particle velocities models have also been used for studying shock-wave

structures by Inamuro and Sturtevant in 1990 [2]. In all these models, although the particle velocity in the distribution function was discretized, space and time were continuous. The full discrete particle velocity model, where space and time are also discretized on a square lattice, was proposed by Hardy and all in 1976 [3] for studying transport properties of fluids. Frisch & all in 1986 [4], recognized the importance of the symmetry of the lattice for the recovery of the Navier-Stokes equation. It was the first time they obtained the correct Navier-Stokes equation starting from the lattice gas automata on a hexagonal lattice. The central ideas in the papers, contemporary with the FHP paper, include the cellular automaton model (Wolfram 1986) [5] and the 3D model using the four dimensional face centered hyper cubic (FCHC) lattice (d'Humieres & all 1986)[6]. The main feature of the LBM is to replace the particle occupation variables (Boolean variables used in the LGA to describe the position of the particles) by a single particle distribution function f_i (real variable), and neglect individual particle motion and particle-particle correlations in the kinetic equations (McNamara & Zanetti 1988)[7]. This procedure eliminates the statistical noise in the LBM. The kinetic form of the LBM is still the same as the LG automata form; this gives the LBM the advantage of retaining the locality in the kinetic approach which is essential for parallelism.

An important simplification of the LBM was made by Higuera & Jmenez (1989) [8], who linearized the collision operator by assuming that the distribution is close to the local equilibrium state. An enhanced collision operator approach which is linearly stable was proposed by the same author [9]. A particular simple linearized version of the collision operator makes use of a relaxation time towards the local equilibrium using a single time relaxation. The relaxation term is known as the Bhatnagar-Gross-Krook (BGK) collision

operator (Bhatnagar & all 1954)[10] and has been independently suggested by other authors (Qian & all 1990 [11], Chen & all 1991[12]). In this Lattice BGK (LBGK) model, the local equilibrium distribution is chosen to recover the Navier-Stokes macroscopic equations (Qian & all 1992[13], Chen & all 1992[14]). Use of the lattice BGK model makes the computations more efficient and allows flexibility of the transport coefficient.

We can understand from the section above that the principal focus of the LBM is the average macroscopic behavior. The kinetic equation provides many of the advantages of molecular dynamics, including clear physical pictures, easy implementation of boundary conditions, and fully parallel algorithms. Therefore, the LBM method has gained popularity in recent years for simulating single-fluid and multiple-fluid phase flow, especially through porous media [15].

For the treatment of the boundary conditions different schemes and models with different precisions have been developed for different types of boundaries. The strategy to model the wall boundary conditions in the LBM is adopted from the LGA method. The particle distribution bounce back scheme, for example, was applied initially by Wolfram in 1986 [16] and Lavallee & all (1991) [17] to obtain no-slip velocity conditions at the wall. The bounce back scheme is one of the most popular schemes used in the LBM, it is based on the idea that when a velocity distribution streams to a wall node, the particle distribution scatters back to the node it came from. The bounce back scheme satisfies the no-slip velocity condition and it is very easy to implement, which makes the LBM very suitable to use in complicated geometries, such as flow through porous media.

The way we define the grid in the LBM (calculation flow domain) imposes on the nodes near the boundaries to have some neighboring nodes lying outside the flow domain. Therefore the distribution functions at these no-slip nodes are not uniquely defined. The bounce back scheme is a simple way to fix these unknown distributions on the wall node.

On the other hand, Cornubert & al in 1991 [18], Ziegler in 1993 [19] and Ginzbourg & Adler in 1994 [20] found that the bounce back scheme is only first order accuracy. This lowers the numerical accuracy of the LBM method which was initially of the second order.

Other boundary schemes have been proposed in order to improve the numerical accuracy of the LBM. Skordos (1993)[21] suggested including velocity gradients in the equilibrium distribution function at the wall nodes. Noble & al (1995)[22] proposed using hydrodynamic boundary conditions on no-slip walls by enforcing a pressure constraint. Maier & al (1996) [23] modified the bounce-back condition to nullify net momentum tangent to the wall and to preserve momentum normal to the wall. Zou and He (1997)[24] extended the bounce back condition for the nonequilibrium portion of the distribution. Ziegler (1993) noticed that if the boundary was shifted into fluid by one half mesh unit by placing the nonslip condition between nodes, then the bounce back scheme will give second order accuracy. Simulations demonstrated that these schemes yield good results for fluid flows around simple wall boundaries. However, it appears that the extension of these simple assumptions to arbitrary boundary conditions is difficult. Chen & al (1996) [25] viewed the LBM as a special finite difference scheme of the kinetic equation and they adopted staggered grid mesh discretization from traditional finite difference methods and proposed using a second order extrapolation scheme of distributions in the flow to obtain

the unknown particle distribution functions. The extrapolation scheme that they used was simple and it can be extended to use velocity, temperature and pressure boundary conditions and their derivatives (Maier& al 1996, Zou & He 1997). The simulations of this method have shown good agreements with the analytical results using benchmark cases.

Different studies of the stability of the method have been performed. The discrete velocity equation of the LBM is a hyperbolic equation that approximates the Navier-Stokes equation in the nearly incompressible limit (this is demonstrated in Appendix A), the numerical accuracy depends on the mach number.

From a numerical point of view the LBM, like other kinetic equations, is a relaxation method (Cao & al 1997)[26] for the macroscopic equations, which has much in common with the explicit "penalty" or "pseudo-compressibility method" (sterling & Chen 1996)[27]. This view was used by Ancona (1994)[28] to generalize the LBM to include fully Lagrangian methods for the solution of partial differential equations. The kinetic relaxation method for solving a hyperbolic conservation system was proposed by Jin and Xin (1995)[29]. This approach uses the relaxation method to model the nonlinear flux terms in the macroscopic equations and thus it does not require nonlinear Riemann solvers. Using this relaxation method, Jin and Katsoulakis calculated the curvature-dependent front propagation. In principle, the LBM and the kinetic relaxation method are very much alike. The kinetic relaxation was developed mainly for shock capture in Euler systems, whereas the LBM is more focused on viscous complex flows in the nearly incompressible limit. Nadiga and Pullin (1994)[30] proposed a simulation scheme for kinetic discrete velocity gases based on local thermodynamic equilibrium. Their method seems more general and

able to obtain high order numerical accuracy. Their finite volume technique was further developed (Nadiga 1995)[31] to solve the compressible Euler equation by allowing the discrete velocities to adapt to the local hydrodynamic state. Elton & all (1995)[32] studied issues of convergence, consistency, stability, and numerical efficiency for lattice Boltzmann models for viscous Burgers' equation and advection-diffusion system.

As we mentioned before a large number of applications of the LBM is directed to the study of the porous medium flow. A quantitative evaluation of the capability and the accuracy of the LBM for modeling flow through porous media have been conducted by Pan and all (2004) [15]. They performed a comparative study of the single phase LBM using two different models for the collision term, the multiple relaxation time (MRT) and the Bhatnagar-Gross-Krook (BGK) single relaxation time (SRT) collision operators. They have also investigated different fluid-solid boundary conditions including: the standard bounce-back (SBB) scheme, the linearly interpolated bounce-back (LIBB) scheme, the quadratically interpolated bounce back (QIBB) scheme and the multi-reflection (MR) scheme. The results show that the MRT-LBM model is superior to the BGK-LBM model, and interpolation significantly improves the accuracy of the fluid-solid boundary conditions.

In another paper, Yang & al(2011) [33] used the LBM method with a BGK model and a bounce back scheme for the boundary condition, to capture the hydro-mechanical impacts on the solid skeleton imposed by the fluid flowing through porous media at the pore-scale. The friction coefficient for various pore throats, of well studied problems of fluid creeping through idealized 2D and 3D porous mediums, are calculated using the LBM simulation

results. They concluded that the simulation results agree well with the data reported in the literature.

One of the most recent and active research areas is the application of the LBM for the flow simulation inside the Fuel cell Gas Diffusion Layer (GDL). In fact, a key performance limitation in fuel cells, called the mass transport loss, originates from the flooding of the constituent components by the liquid water transport. Hence, water management plays an important role in maintaining cell performance, stability, and durability.

Bo Han & al (2012)[34], simulated the fluid flow inside a fuel cell polymer electrolyte membrane (PEM), they used the LBM method to examine the interfacial phenomena in liquid water transport in the porous materials. The authors used a BGK model for the collision term and the Shan-Chen multiphase approach to handle liquid water transport processes in the porous diffusion media.

Lee & al (2009) [35] performed a pore-network analysis of two-phase water transport in the GDL of a polymer electrolyte membrane fuel cells (PEMFCs). They provided a good theoretical understanding of the porous medium flow. They also developed a pore network model to study the water transport in the GDL. The numerical results showed that the saturation distribution in GDLs maintained a concave shape, indicating the water transport in GDLs was strongly influenced by capillary processes.

In another paper, Partha & al (2009) [36] have studied the flooding phenomena in polymer electrolyte fuel cells. The mesoscopic modeling approach is a two phase LBM, with an interaction-potential based approach. This modeling is widely used due to its simplicity in implementing boundary conditions in complex porous structures and remarkable

versatility in terms of handling fluid phases with different densities, viscosities and wettabilities, as well as the capability of incorporating different equations of state. Using this model, they could quantify the structure wettability performance interlinks in the CL and GDL components.

CHAPTER 2: Generalities of the Boltzmann equation

Our main interest in this part is to develop the discretized form of the Boltzmann equation, also referred to as the lattice Boltzmann equation. The collision term found in the continuous Boltzmann equation is approximated by the BGK model with a single relaxation time. Most of the theoretical work developed in this section is based on references [37]-[41].

A thorough development of the Navier-Stokes equations from the Lattice Boltzmann Equations is presented in APPENDIX A.

The continuous Boltzmann equation with the BGK model takes the following form:

$$\frac{\partial f}{\partial t} + \xi \nabla f = -\frac{1}{\lambda}(f - g) \qquad (1)$$

Where $f \equiv f(x, \xi, t)$ is the single particle distribution function, ξ is the macroscopic velocity, λ is the relaxation time due to collision, and g is the Boltzmann-Maxwellian distribution function defined as:

$$g \equiv \frac{\rho}{(2\pi RT)^{D/2}} \exp\left(-\frac{(\xi - u)^2}{2RT}\right) \qquad (2)$$

Where R is the ideal gas constant, D is the dimension of the space, and ρ, u, and T are the macroscopic density of mass, velocity and temperature, respectively. They can be determined using the following relations:

$$\begin{aligned} \rho &= \int f d\xi = \int g d\xi \\ \rho u &= \int \xi f d\xi = \int \xi g d\xi \\ \rho \varepsilon &= \frac{1}{2}\int (\xi - u)^2 f d\xi = \frac{1}{2}\int (\xi - u)^2 g d\xi \end{aligned} \qquad (3)$$

The energy ε could be also written in terms of temperature T:

$$\varepsilon = \frac{D_0}{2} RT = \frac{D_0}{2} N_A k_B T \qquad (4)$$

Where D_0, N_A and k_B are the number of degrees of freedom of a particle, Avogadro's number and the Boltzmann constant, respectively.

We used the assumption of Chapman Enskog to develop equations (3) where:

$$\int h(\xi) f(x, \xi, t) d\xi = \int h(\xi) g(x, \xi, t) d\xi \qquad (5)$$

Where $h(\xi)$ is a linear combination of coalitional invariants (Conserved quantities):

$$h(\xi) = A + B.\xi + C.\xi.\xi \qquad (6)$$

Where A and C are arbitrary constants and B is an arbitrary constant vector.

1. Time discretization

The Boltzmann equation with the BGK model (1) can be written in the form of an ordinary differential equation:

$$\frac{df}{dt} + \frac{1}{\lambda} f = \frac{1}{\lambda} g \qquad (7)$$

Where $\frac{d}{dt} \equiv \frac{\partial}{\partial t} + \xi.\nabla$ is the time derivative along the characteristic line ξ.

Equation (7) is a linear, first order, non homogeneous, ordinary differential equation, we can integrate using the Integrating Factor Method [41].

The integration of (7) over a time step ∂_t results in the following expression:

$$f(x + \xi \partial_t, \xi, t + \partial_t) = \frac{1}{\lambda} e^{-\frac{\partial_t}{\lambda}} \int_0^{\partial_t} e^{\frac{t'}{\lambda}} g\left(x + \xi t', \xi, t + t'\right) dt' + e^{-\frac{\partial_t}{\lambda}} f(x, \xi, t) \qquad (8)$$

Assuming that ∂_t is small enough and g is smooth enough locally, using the Taylor expansion series the following approximation could be made:

$$g\left(x+\xi t^{'},\xi,t+t^{'}\right)=\left(1-\frac{t^{'}}{\partial_t}\right)g(x,\xi,t)+\frac{t^{'}}{\partial_t}g(x+\xi\,\partial_t,\xi,t+\partial_t)+O\left(\partial_t{}^2\right) \qquad (9)$$

$$\text{Where} \quad 0\le t^{'}\le\partial_t$$

The leading terms neglected in the approximation of the function g are in the order of $O\left(\partial_t{}^2\right)$.

Using this approximation in expression (8) and the integration per part we obtain the following expression:

$$\begin{aligned} &f(x+\xi\,\partial_t,\xi,t+\partial_t)-f(x,\xi,t)\\ &\quad=\left(e^{-\frac{\partial_t}{\lambda}}-1\right)[\,f(x,\xi,t)-g(x,\xi,t)]+\left(1+\frac{\lambda}{\partial_t}\left(e^{-\frac{\partial_t}{\lambda}}-1\right)\right)\\ &\quad\times[\,g(x+\xi\,\partial_t,\xi,t+\partial_t)-g(x,\xi,t)] \qquad (10)\end{aligned}$$

If we go further and expand $e^{-\frac{\partial_t}{\lambda}}$ using Taylor expansion and neglect the terms of order $O\left(\partial_t{}^2\right)$ or smaller in the right hand side of equation (10), we develop the following expression:

$$f(x+\xi\,\partial_t,\xi,t+\partial_t)-f(x,\xi,t)=-\frac{1}{\tau}[f(x,\xi,t)-g(x,\xi,t)] \qquad (11)$$

Where $\tau\equiv\frac{\lambda}{\partial_t}$ is the dimensionless relaxation time.

Thus, equation (11) is accurate to the first order in ∂_t .It is the evolution equation of the distribution function f with discrete time.

2. Calculation of the hydrodynamic moments

Discretization in momentum space ξ is needed in order to evaluate numerically the hydrodynamic momentum equation (3), using the following discretization:

$$\int \phi(\xi) g(x, \xi, t) d\xi = \sum_{\alpha} W_{\alpha} \phi(\xi_{\alpha}) g(x, \xi_{\alpha}, t) \qquad (12)$$

Where $\phi(\xi)$ a polynomial of ξ, W_{α} is the weight coefficient of the quadrature, and ξ_{α} is the discrete velocity set or the abscissas of the quadrature. Accordingly, the hydrodynamic moments of Eqs.(3) can be computed by :

$$\begin{aligned} \rho &= \sum_{\alpha} f_{\alpha} = \sum_{\alpha} g_{\alpha} \\ \rho u &= \sum_{\alpha} \xi_{\alpha} f_{\alpha} = \sum_{\alpha} \xi_{\alpha} g_{\alpha} \\ \rho \varepsilon &= \frac{1}{2} \sum_{\alpha} (\xi_{\alpha} - u)^2 f_{\alpha} = \frac{1}{2} \sum_{\alpha} (\xi_{\alpha} - u)^2 g_{\alpha} \end{aligned} \qquad (13)$$

Where

$$\begin{cases} f_{\alpha} \equiv f_{\alpha}(x,t) \equiv W_{\alpha} f(x, \xi, t) \\ g_{\alpha} \equiv g_{\alpha}(x,t) \equiv W_{\alpha} g(x, \xi, t) \end{cases} \qquad (14)$$

3. Low mach number approximation

In the lattice Boltzmann equation, the equilibrium distribution function is obtained by a truncated small velocity expansion (Low Mach Number approximation).

That same technique will be used in what follows:

$$g = \frac{\rho}{(2\,\pi\,RT)^{\frac{D}{2}}} \exp\left(-\frac{(\xi - u)^2}{2RT}\right)$$

$$= \frac{\rho}{(2\,\pi\,RT)^{D/2}} \exp\left(-\frac{\xi^{\,2}}{2RT}\right) \exp\left(\frac{\xi \,.\, u}{RT} - \frac{\xi^{\,2}}{2RT}\right)$$

$$= \frac{\rho}{(2\,\pi\,RT)^{D/2}} \exp\left(-\frac{\xi^{\,2}}{2RT}\right) \times \left\{1 + \frac{\xi \,.\, u}{RT} + \frac{(\xi \,.\, u)^2}{(RT)^2} - \frac{u^2}{2RT}\right\} + O(u^3) \qquad (15)$$

For convenience we use the notation f^{eq} to describe the equilibrium distribution function with truncated small velocity expansion:

$$f^{eq} = \frac{\rho}{(2\,\pi\,RT)^{D/2}} \exp\left(-\frac{\xi^{\,2}}{2RT}\right) \times \left\{1 + \frac{\xi \,.\, u}{RT} + \frac{(\xi \,.\, u)^2}{(RT)^2} - \frac{u^2}{2RT}\right\} \qquad (16)$$

In the above expression of f^{eq} the terms of order $O(u^3)$ and higher has been neglected but we can maintain some of these terms if necessary.

For the discretization of phase space, different configurations of the lattice structure exist for 2D and 3D flows. We are adopting the D2Q9 model for the purpose of our application. The main features of this model are presented in the following chapter.

CHAPTER 3: Computation and results

In this chapter we present the main characteristics of the LBM code that we developed. After that, we show results for some fundamental problems for the purpose of validation.

1. LBM code characteristics

As we have seen before, the Lattice Boltzmann equation is linear (using the BGK model to approximate the collision term). In comparison with the Navier-Stocks, the non linear advection term in the macroscopic approach is replaced by the linear streaming process in the LBM. The LBM can be considered an explicit method, where there is no need to solve simultaneous equations at every time step, since the collision and streaming processes are local. Hence the LBM can be easily implemented in parallel computation techniques.

- **BGK Approximation**

One of the difficulties in solving the Boltzmann equation is the complicated nature of the collision integral. The main method used to overcome this difficulty is to utilize the BGK (Bhatnagar-Gross-Krook) approximation which will constraint the application of the LBM to incompressible flow (low Mach number). Therefore, in simulation, the lattice velocity must be set to small values (order of 0.1) in order to reduce error of simulation. However we can allow weak variation in density.

- **Lattice model D2Q9**

It is common to use the D2Q9 model to solve diffusive-convective problems. Figure 1 illustrates the streaming direction and values of the weight coefficients w_k, c_k are unit vectors along the lattice streaming direction.

The domain of interest should be divided into lattices, using the D2Q9 structure.

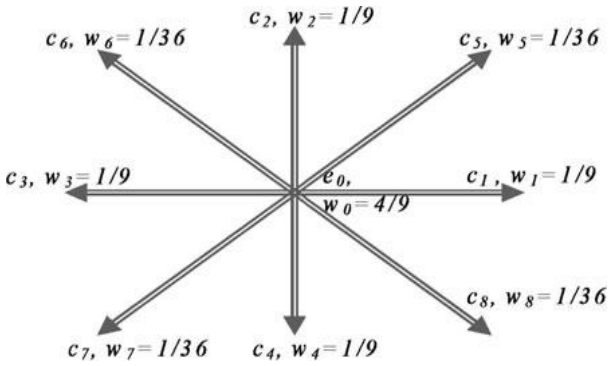


Fig. 1: D2Q9 lattice configuration

The description of the D2Q9 model is lengthy and only practical results will be presented here. A detailed presentation could be found in [37].

The lattice Boltzmann equation can be written as:

$$f_k(x + \Delta x, t + \Delta t) = f_k(x, t)[1 - \omega] + \omega f_k^{eq}(x, t) \tag{17}$$

Using the BGK model, the equilibrium distribution function can be written as:

$$f_k^{eq} = w_k \rho(x, t)\left[1 + \frac{c_k . u}{c_s^2} + \frac{1}{2}\frac{(c_k . u)^2}{c_s^4} - \frac{1}{2}\frac{u^2}{c_s^2}\right] \tag{18}$$

Where:

$$\begin{cases} c_s = \frac{c_k}{\sqrt{3}} \\ c_k = \frac{\Delta x}{\Delta t} i + \frac{\Delta y}{\Delta t} j \\ u = ui + vj \end{cases} \tag{19}$$

Detailed analysis shows that LBM resembles Navier-Stokes equations for incompressible flow at low Mach numbers. The fluid viscosity is related to the relaxation frequency as:

$$\nu = \frac{\Delta x^2}{3\Delta t}(\omega - 0.5) \qquad (20)$$

Reynolds number, $\mathrm{Re} = \frac{UL}{\nu}$ where U and L are characteristic velocity and characteristic length in macro-scale, respectively.

The value of $\frac{L}{\Delta x} = N$ is the number of lattices in the direction of the characteristic length. The usual practice in LBM is to consider Δx unity, hence $L = N$ and $\mathrm{Re}_{lattice} = \frac{U_{lattice} \cdot N}{\nu_{lattice}}$, which is commonly called, Lattice Reynolds Number.

The real and the lattice Reynolds number must be equal. $U_{lattice}$ and $\nu_{lattice}$ can be arbitrary selected within the range that insures the stability of the solution and reduces the error of calculation. In general, the value of $U_{lattice}$ should be in the order of 0.1 or 0.2, which is not related to the macroscopic velocity. Care should be taken in using very small values of $\nu_{lattice}$,which may cause stability problems.

Figure 2 illustrates the LBM calculation sequences; it is an iterative method. We start by calculating the equilibrium distribution function (Equation (10)) at each lattice site. Then we calculate the collision term (the right hand side of equation (9)). After that we stream the values of the distribution function and we calculate the flow variables. Finally we enforce the boundary conditions. The same process is repeated until we satisfy the convergence criteria.

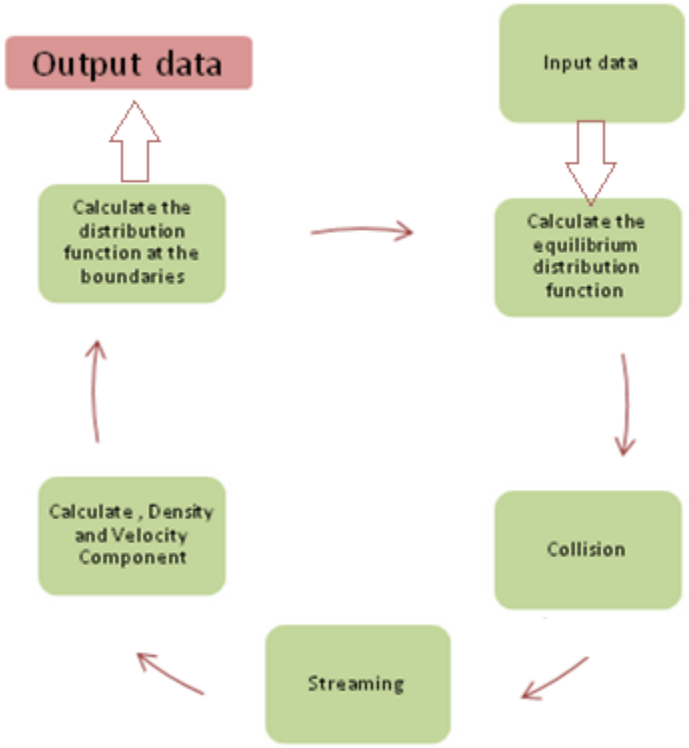


Fig. 2: LBM calculation sequences

- **Boundary conditions**

Specifying the boundary conditions for the Navier-Stokes equations is somehow straightforward, comparing to the LBM method. In LBM, the inward distribution functions need to be determined at the boundaries. Therefore, appropriate equations for calculating the distribution functions at the boundaries need to be determined.

In the literature different methods are used, the choice depends on the precision and the stability of the calculations. In what follows we are mainly using the bounce back scheme.

The bounce back scheme is used to model solid stationary or moving boundary conditions, non-slip conditions, or flow-over obstacles. The method is quite simple and mainly implies that an incoming particle towards the solid boundary bounces back into the flow domain.

A few versions of bounce back scheme have been suggested. The bounce back ensures conservation of mass and momentum at the boundary.

The simplest bounce back scheme is shown in Figure (3); the lattices are located directly at the solid surface. We simply let $f_5 = f_7$, $f_6 = f_8$ and $f_2 = f_4$, where f_7, f_4 and f_8 are known distribution functions from the streaming process. This scheme is first order accurate; it is rather simple and easily implemented comparing to other schemes, and it will be adopted in the coding.

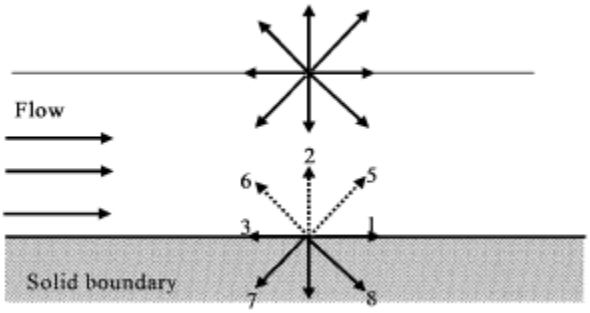


Fig. 3: Lattice position at the solid boundary

2. Simulation results

In this segment, we present results for different applications. We start by discussing three bench mark applications, the lid driven cavity flow, the channel flow and the shock tube. Then we discuss the flow inside a channel with a presence of obstacles to simulate a simple porous medium.

- **Lid Cavity Flow**

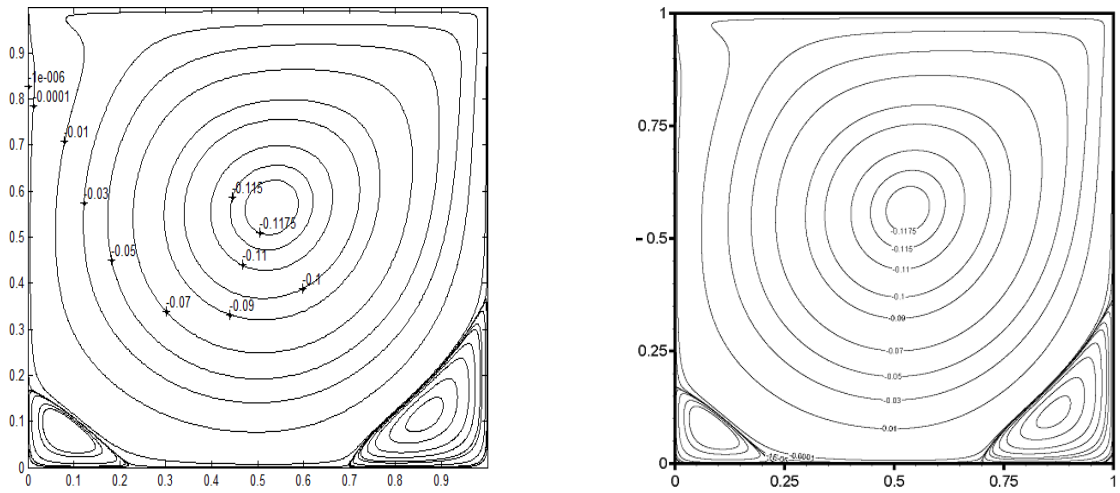
Lid driven cavity is a benchmark problem that is usually used to test CFD codes. A square cavity of 0.20m side is filled with a fluid, its kinematic viscosity is $1.2 \times 10^{-3} \mathrm{m}^2/\mathrm{s}$. The lid is set to motion with a speed of 6 m/s.

As mentioned before, in LBM we are free to use any values for U_{lid} and the viscosity provided that the Reynolds number and the geometrical aspect ratio are the same between the two forms. For our test case $\mathrm{Re} = 1000$. In order to reduce compressibility effects of LBM, we set $U_{lattice}$ to 0.1 and viscosity to 0.01, then we use 100 lattices to match the Reynolds number $1000 = \frac{U_{lattice} \cdot N}{\nu_{lattice}}$.

As mean of comparison we used the data from 223A class (Prof.Elgobashi), where we used a Navier-Stockes solver for the same cavity configuration. We also compared our results with Erturk's paper [44].

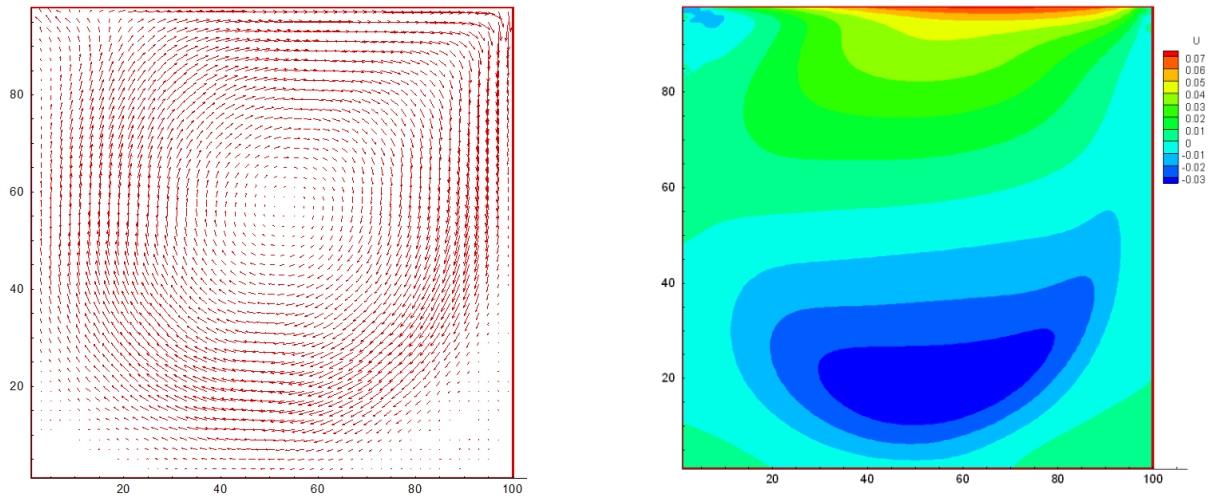
The results in Figure 4 and 5 show the streamlines plots and the velocity profiles for Re=1000, Figure 4 (a) and (b) are the results obtained by our LBM code and Ertuk's [44], respectively. Using LBM simulation, it is clear that we could detect the three vortices at the bottom left, bottom right and top left. Also the position and values of the stream lines are identical to Ertuk's results.

Figure 6 shows the plots for the horizontal component of the velocity along a vertical line passing through the geometric center of the cavity at Re=1000. And again we see a perfect match between the Navier-Stokes solution, the LBM simulation and the experimental data.



a) LBM 100x100 lattice b) NS solver 600x600 cells

Fig. 4: Cavity Flow streamlines, Re=1000



a) Velocity vector field b) Velocity Magnitude Field

Fig. 5: LBM 100x100 Lattice, Re=1000

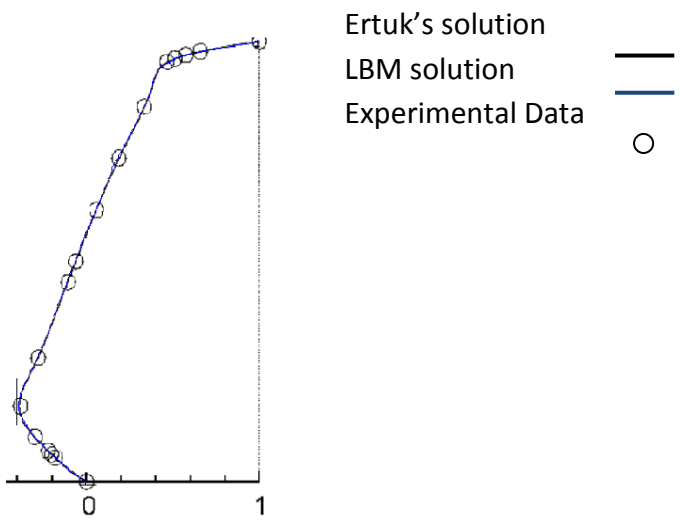


Fig. 6: Horizontal velocity profiles at the center of the cavity.

- **Two-Dimensional Channel Flow**

In another bench mark test, we simulated the water flow between two parallel plates (Figure 7) with a uniform velocity of 0.02 m/s, a distance betwee plates of 2.0 cm, and a plate length of 50 cm.

The Reynolds number is 400, using 1000 lattice in the x direction and 40 lattices in the y direction, with a lattice velocity of 0.2 and lattice viscosity of 0.02.

The Reynolds number and the channel dimensions ratio are kept the same between the real and lattice case.

The velocity profiles in Figure 8 show that we could reproduce a parabolic profile for the developed flow as the analytical solution predicts.

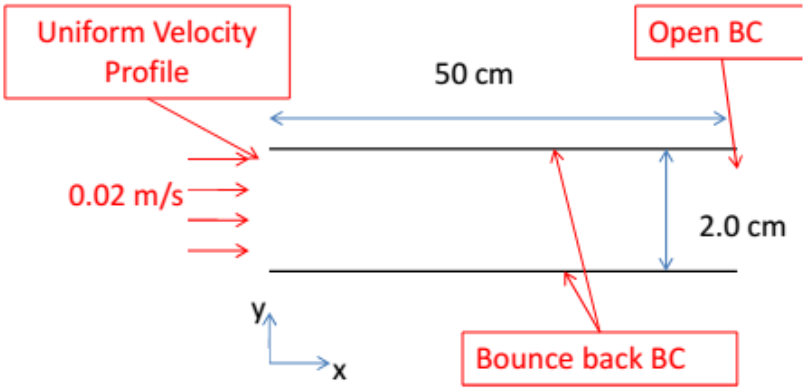


Fig. 7: Channel Flow configuration

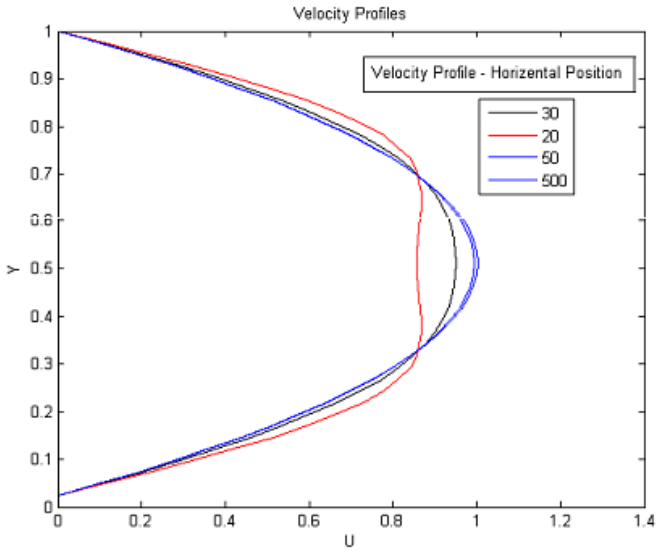


Fig. 8: Axial velocity profile at different lattice positions

- **Porous medium**

In our first test for the porous medium, five obstacles are positioned in the developed flow of the 2D channel, presented above. Figure 9 illustrates the configuration and the position of the obstacles inside the channel. Note that all the numbers on Figure 9 represent the number of lattices in the x and y directions.

We can see in Figure 10 the velocity vectors and fluid behavior at the obstacles region.

Figures 11 show the axial velocity profiles at different positions inside the 2D channel. We can see that the boundary condition at the solid surface is properly enforced. At the axial lattice position x=400, the flow is fully developed.

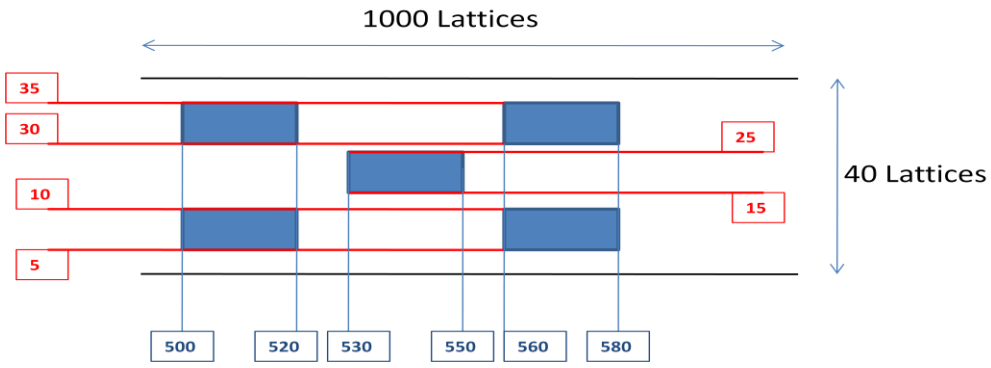


Fig. 9: Position and configuration of obstacles inside a channel flow.

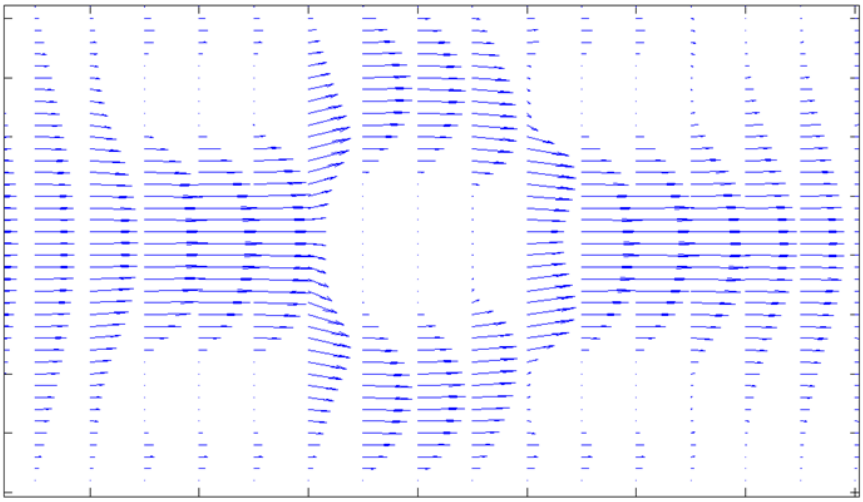


Fig. 10: velocity field flow around the obstacles.

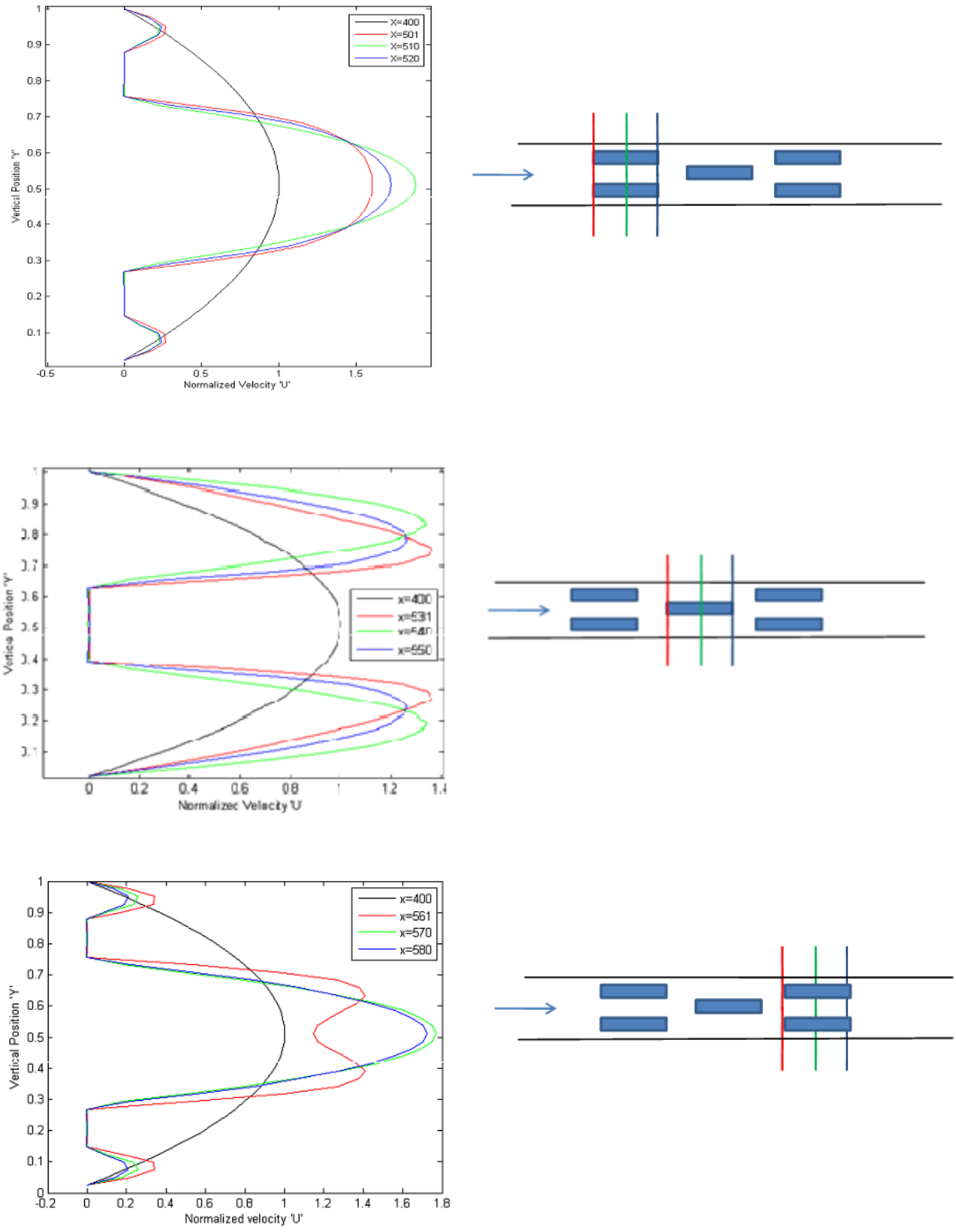


Fig. 11: Velocity profiles at different lattice positions

- **D1Q3 LBM Shock Wave validation case**

We use the classic case of a shock tube to benchmark the LBM model [46] we are using. In what follows, we present the analytical solutions for speed and density and compare it with the Lattice Boltzmann Method solution.

We present in the following the steps to obtain the analytical solutions for the expansion fan and the moving shock wave:

- Finite non linear waves-Isothermal inviscid expansion fan:

The compressible continuity equation and the Euler equations are the two differential equations governing the propagation of an isothermal inviscid expansion fan (Figure 12) [27].

Using the characteristic method, we can reduce the governing equations to:

$$\begin{cases} du + \frac{c}{\rho} d\rho = 0 \text{ along the characteristic } \frac{dx}{dt} = u + c & (21) \\ du - \frac{c}{\rho} d\rho = 0 \text{ along the characteristic } \frac{dx}{dt} = u - c & (22) \end{cases}$$

Where $\rho = \rho(x,t)$ and $u = u(x,t)$ are the density and the velocity of the fluid and c is the speed of sound constant everywhere in the domain.

Using the Riemann Invariant and some analytical developments we can get:

$$\rho = \rho_3 e^{-\left(\frac{x}{ct}+1\right)} \quad (23) \qquad \text{and } u = c \ln \frac{\rho_3}{\rho} \quad (24)$$

This represents the values of the density and the velocity anywhere in the region of the expansion fan (Figure 12) as a function of time. The density in region 3 and the sound speed is constant since we are considering an Isothermal case.

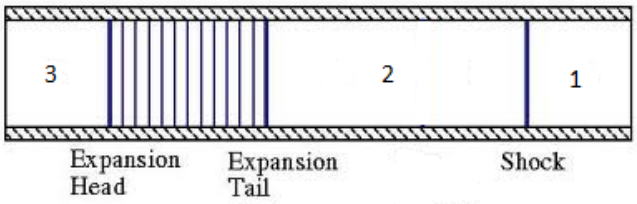


Fig. 12: Shock tube

- Moving shock wave (Isothermal inviscid conditions):

The equations governing the moving normal shock wave (Figure 13) are the continuity and the momentum equations, they take the following forms [45]:

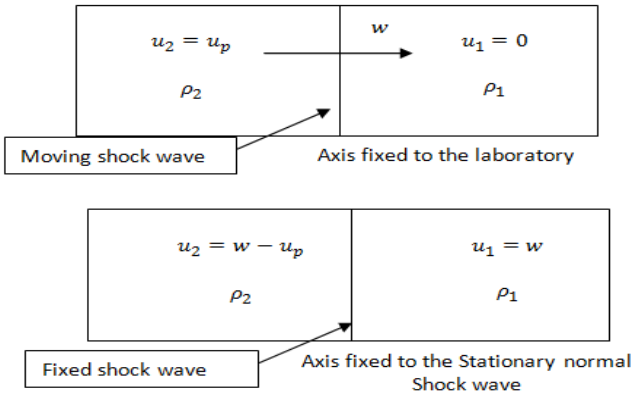


Fig. 13 : Normal Shock wave propagation

$$\begin{cases} \rho_1 u_1 = \rho_2 u_2 & (25) \\ p_1 + \rho_1 u_1^2 = p_2 + \rho_2 u_2^2 & (26) \end{cases}$$

By considering the relative velocity w, defined as $u_2 = w - u_p$, where $w = u_1$, equations (25) and (26) take the following form:

$$\begin{cases} \rho_1 w = \rho_2 (w - u_p) & (27) \\ p_1 + \rho_1 w^2 = p_2 + \rho_2 (w - u_p)^2 & (28) \end{cases}$$

Combining equations (27) and (28) and considering the equation of state $p = \rho c^2$ we have:

$$\begin{cases} w^2 = \frac{\rho_2}{\rho_1} c^2 & (29) \\ u_p = c\sqrt{\frac{\rho_2}{\rho_1}} . \left(1 - \frac{\rho_1}{\rho_2}\right) & (30) \end{cases}$$

- LBM simulation:

In this simulation we are using the D1Q3 (Figure 14) model with 2000 lattices in the x direction. We impose the bounce back scheme at the two ends of our 1D model. We constrain the initial velocity to null and we impose a lattice density of 4 in the first half domain (0 to lattice 999) and 2 in the other half of the domain.

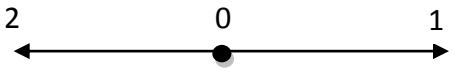


Fig. 14: D1Q3 model lattice.

Figure 15 shows the results of the LBM simulation. The shock wave propagates to the right at a lattice velocity of 0.69 and the flow behind the shock is set to a velocity of 0.2.

Figure 16 shows the comparison between the LBM simulation and the analytical solution, at the lattice time 900. We can see that the LBM simulations and the analytical solutions show a good agreement to predict the position and the shape of both the shock wave and the expansion fan. Notice that sound speed is kept low to satisfy the approximation of the lattice Boltzmann method.

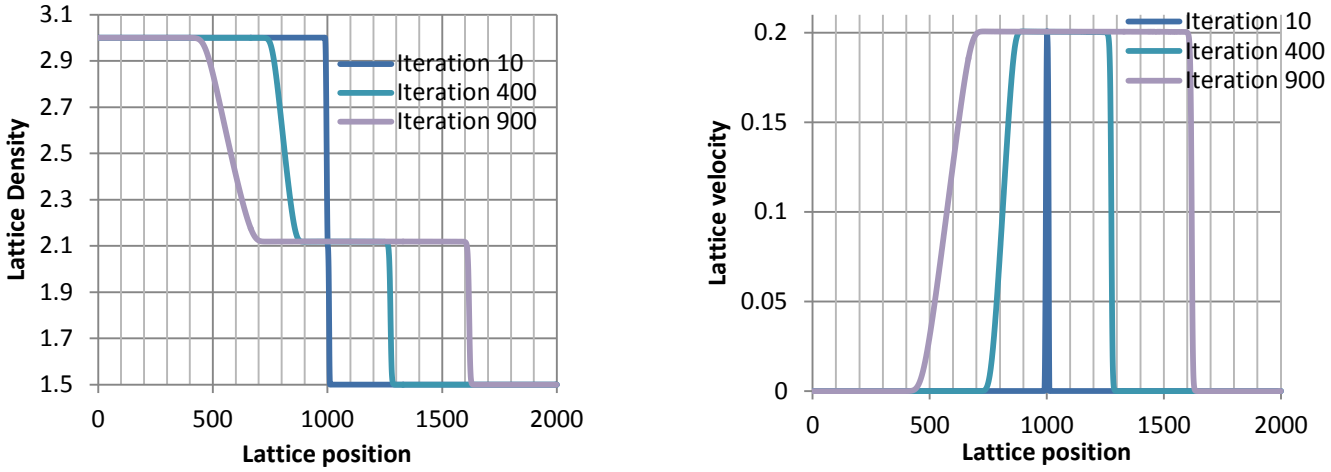


Fig. 15: LBM Simulations, Lattice velocity and density at different time lapses

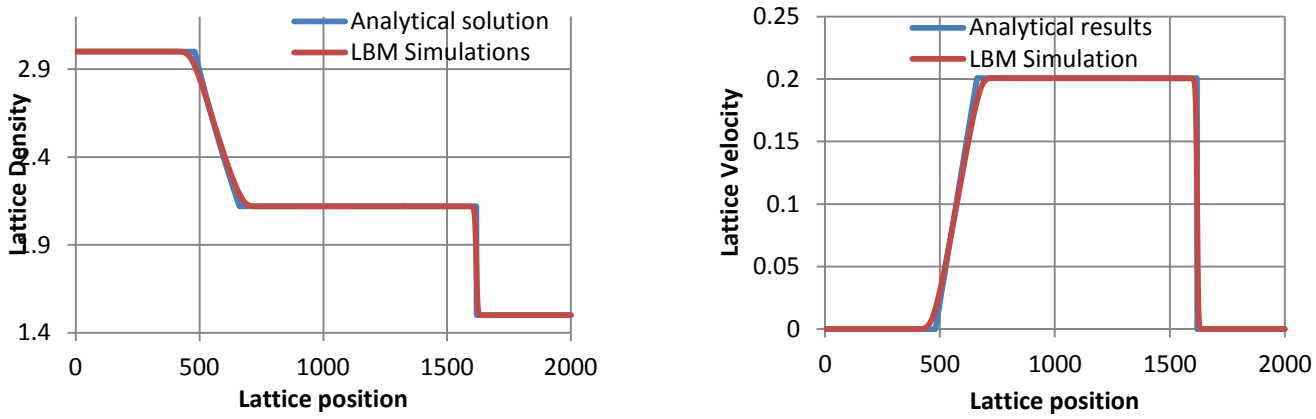


Fig. 16: Density Profile_ Lattice Time 900

- **2D Gas Diffusion Layer**

In this section we present the simulation results of a 2D Gas Diffusion Layer structure (GDL) inside a developed channel flow. Figure 17 illustrates a 3D reconstructed Gas diffusion Layer that consists of carbon fiber. Figure 18 illustrates the configuration and the position of the GDL inside the channel.

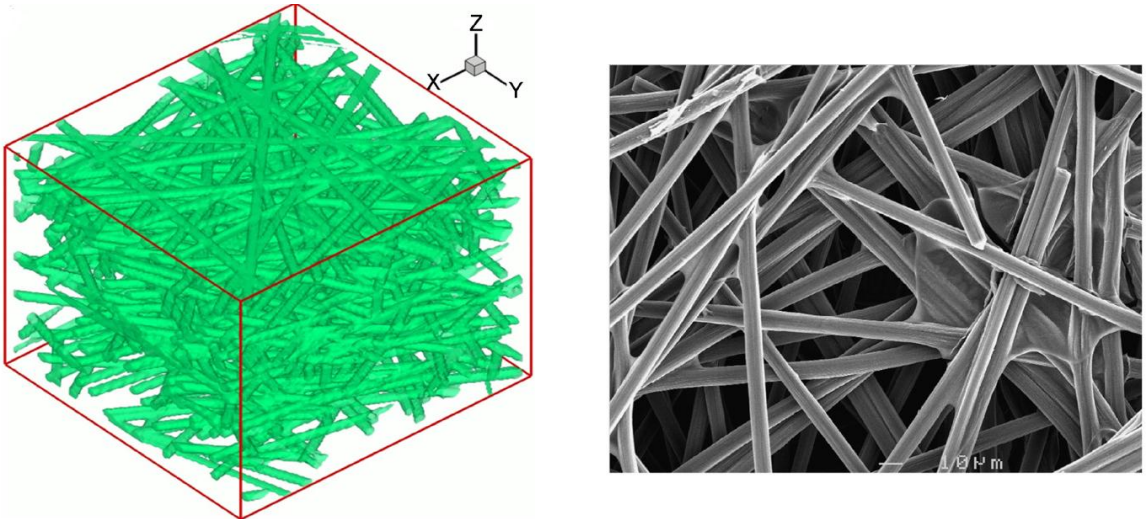


Fig. 17 : Reconstructed Gas diffusion Layer

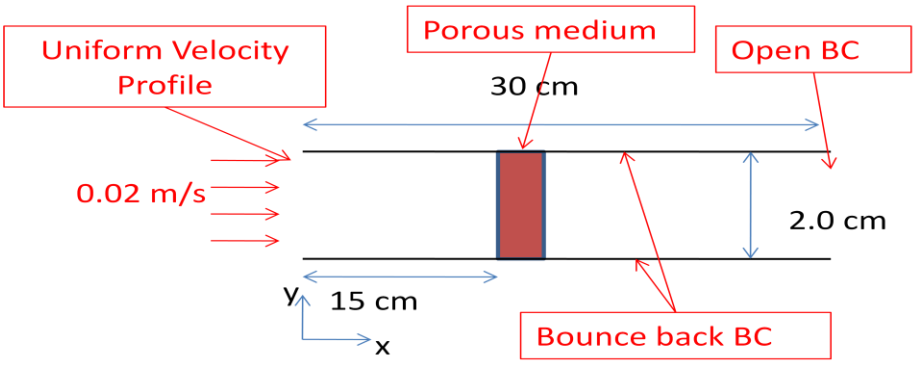


Fig. 18: Simulation setup

The simulation setup properties are as follow:

- We are considering water as the working fluid.
- Reynolds Number = 140.
- LBM grid dimensions:
    - 600 lattice in the x direction
    - 140 lattice in the y direction

Lattice velocity = 0.02.

The Reynolds number and the channel dimensions ratio are kept the same, with respect to the real dimensions.

The code scans over the porous medium shown is Figure 19 and build the boundary conditions matrices automatically as follows:

- Read the porous medium structure file.
- Scan through the structure and build the structure Matrix composed of 1 (material) and 0(void).
- Loop and select the positions where the material is present (1).
- Check the position of the node Up, Down, Right or Left.
- Store the coordinates node in the appropriate boundary vectors, with respect to their position.

Then the matrices are passed to the main LBM code.

We run the code for 120, 000 iterations.

The error on the axial velocity is of the order of 10^{-5}.

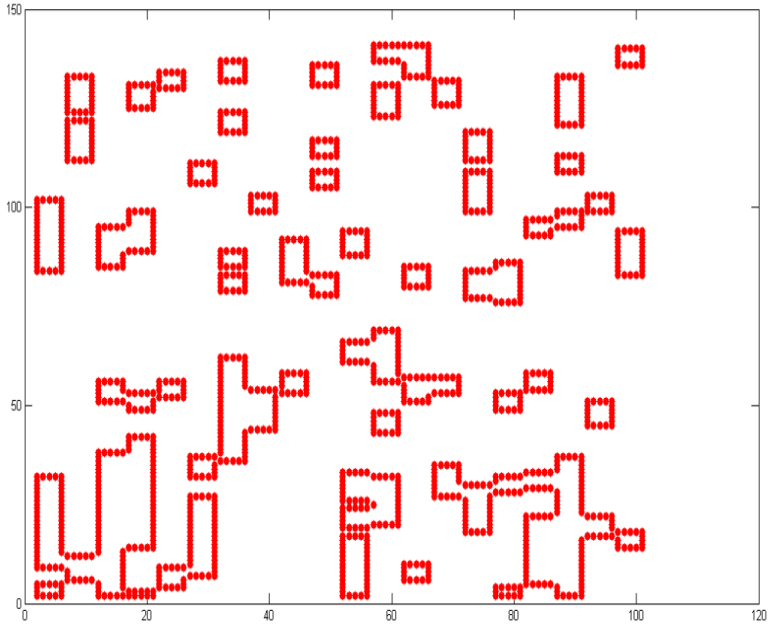


Fig. 19 : 2D GDL porous structure

Figure 20 shows the velocity convergence rate of simulation. We can see that the convergence of the velocity to the final solution is smooth and that a convergence of the order of 10^{-5} is reached at around 12,000 iterations. We can see in Figure 21 the axial velocity profiles in the whole domain. Figure 22 and 23 show more details about the fluid behavior and structure at the porous medium level.

It is clear that the axial velocity reaches its highest level at the center line. This was expected for two reasons, the flow is fully developed and the structure is highly porous in that region. Overall the flow shows the expected structure.

We can already see from the 2D simulations that the size and the distribution of the pores (structure obstacle) are two of the main parameters that will control the flow inside the GDL.

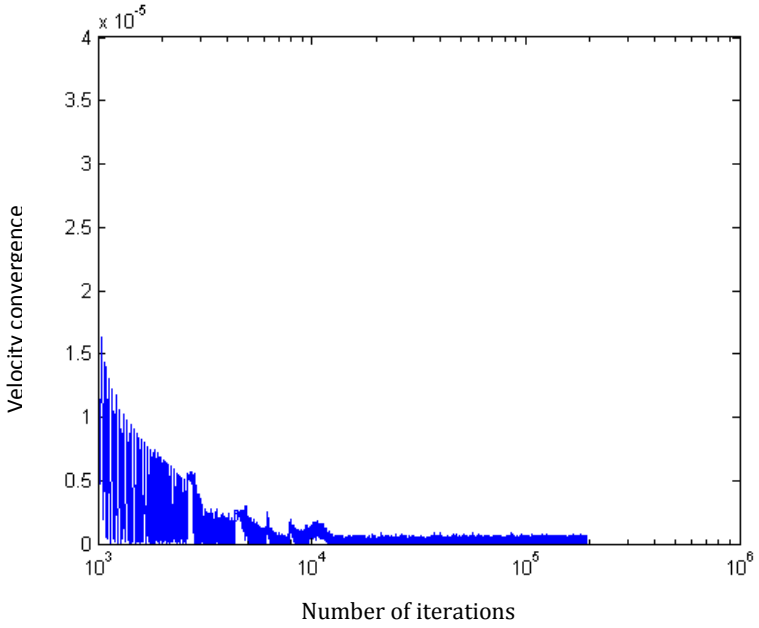


Fig. 20 : Velocity Convergence rate

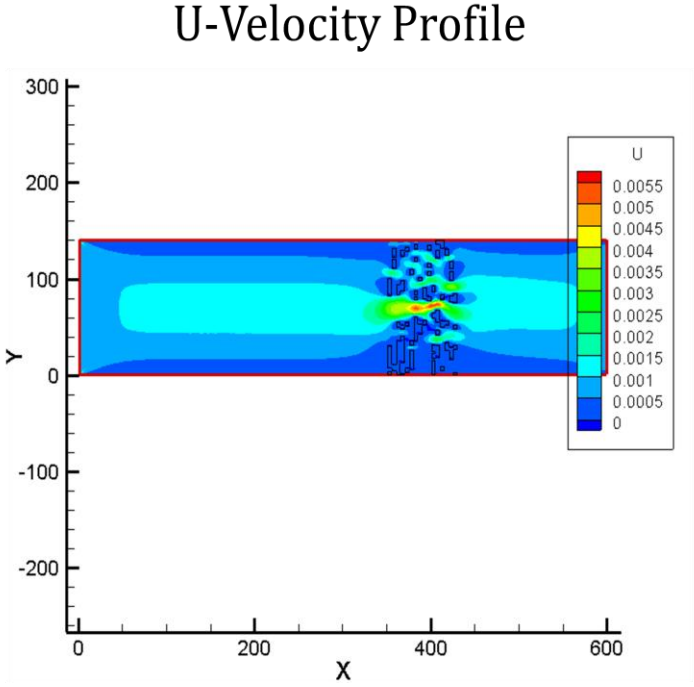


Fig. 21 : Axial velocity profile

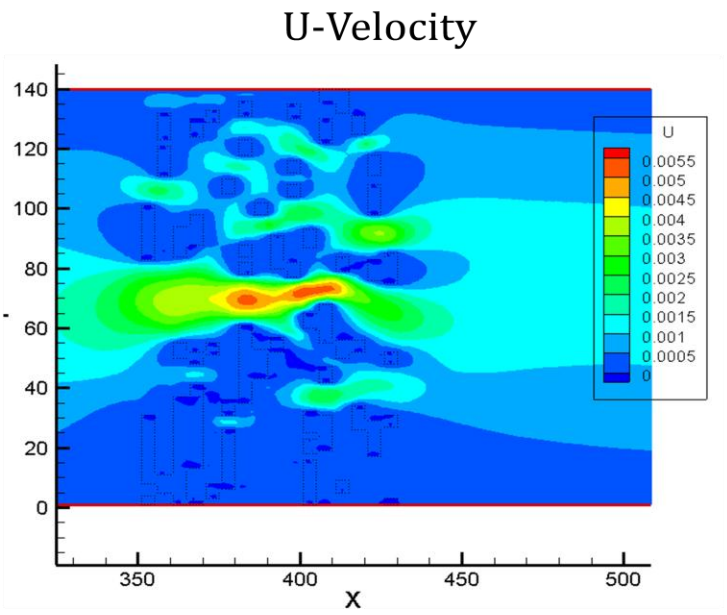


Fig. 22 : Axial velocity profile at the GDL level

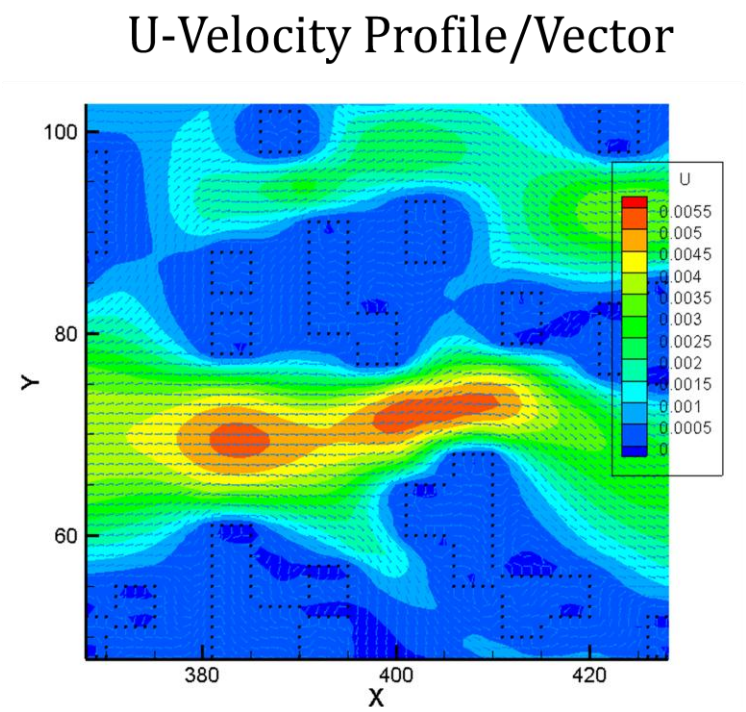


Fig. 23 : Axial velocity vectors and profile at the GDL level

CHAPTER 4: Conclusions and Future Work directions

We have demonstrated in this work that the single phase LBM with the BGK model approximation and a bounce back scheme for the boundary condition, could predict the flow with a high precision at low Reynolds number for the two benchmark tests. We have also presented the results of a 2D GDL carbon structure extracted from an original 3D GDL.

The adopted strategy to handle the solid boundary conditions in the porous media was successful, and it is easily expanded to the 3D case.

One of the main advantages of the LBM is its straightforward parallelism. We used a very efficient tool (plugin) called OpenMP to parallel our code. The calculation of the equilibrium distribution function and the fluid properties at each node of the domain are straight forward. We just need to mention to the compiler that multithreading is allowed before we start looping through the domain lattices, and the compiler will automatically take care of distributing the tasks on different processors. On the other hand, the streaming part can be also parallelized but special care has to be considered. In this case, we cannot let the compiler take care of distributing the job on the compilers automatically but we need to do it manually in such a way that streaming on each direction and on each lattice line must be taken care of by one processor at a time. Overall, the results of the parallelism have shown a decrease of 43% in the convergence time on a quad-core machine. But the gain should be considerable when running a big grid on a cluster machine (example: 72 nodes x 8 processors).

Improvements in the accuracy of the results could be made by considering an extrapolation scheme for the boundary conditions [43]. An extrapolation scheme with a dynamical evolution of the LBGK-type on boundary nodes can be used for solving cases with a variety of boundary conditions. This scheme is proven to be of a second order approximation, which is of the same order of approximation of the lattice Boltzmann equation.

For future work, the simulation results of the porous medium in the fuel cell GDL should be compared and validated to the experimental data available. For this purpose the code should be extended to handle 3D structure and multiphase flow. Multiple LBM schemes to model the structure-wettability-transport interactions as well as the underlying two phase dynamics in porous mediums of a GDL type [35] could be found in the literature. Special care should be given in selecting and ameliorating these schemes.

REFERENCES

1. Chen S and Doolen GD (1998), Lattice Boltzmann method for fluid flows, Annu. Rev. Fluid Mech. 30, 329–364.
2. 1990. Inamuro, Takaji and Sturtevant, Bradford (1990) Numerical study of discrete-velocity gases. Physics of Fluids A, 2 (12). pp. 2196-2203. ISSN 0699-8213.
3. J. Hardy, O. de Pazzis, and Y. Pomeau. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions,Phys. Rev. A 13, 1949 – Published, May 1976.
4. U. Frisch, B. Hasslacher, and Y. Pomeau, Lattice-Gas Automata for the Navier-Stokes Equation, Phys. Rev. Lett. 56, 1505 – Published 7 April 1986.
5. d'Humieres,D. and P.Lallemand. Lattice Gas Automata for Fluid Mechanics. Physica, 140A:326–335, 198,
6. d'Humieres et al., Lattice Gas Models for 3D Hydrodynamics, Europhys. Lett., 2 (4), pp. 291 297 (1986).
7. Guy R. McNamara and Gianluigi Zanetti, Use of the Boltzmann Equation to Simulate Lattice-Gas Automata, Phys. Rev. Lett. 61, 2332 – Published 14 November 1988.
8. Higuera FJ, Jimenez J. 1989. Boltzmann approach to lattice gas simulations. Europhys.Lett. 9:663–68.
9. Higuera FJ, Succi S, Benzi R. 1989. Lattice gas dynamics with enhanced collisions. Europhys., Lett. 9:345–49.
10. Bhatnagar PL, Gross EP, Krook M. 1954. A model for collision processes in gases. I: small amplitude processes in charged and neutral one-component system. Phys. Rev. 94:511–525.
11. Qian YH. 1990. Lattice gas and lattice kinetic theory applied to the Navier-Stokes equations. PhD thesis. Universit´e Pierre et Marie Curie, Paris.
12. Chen S, Chen HD, Martinez D, Matthaeus W.1991. Lattice Boltzmann model for simulation of magnetohydrodynamics. Phys Rev. Lett. 67:3776–79.
13. Qian YH, d'Humi`eres D, Lallemand P. 1992.Lattice BGK models for Navier-Stokes equation. Europhys. Lett. 17:479–84.
14. Chen H, Chen S, Matthaeus WH. 1992. Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. Phys. Rev. A. 45:R5339–42.
15. Pan C, Luo L-S, Miller CT. An evaluation of lattice Boltzmann equation methods for simulating flow through porous media. In: Proceedings of the XVth International Conference on Computational Methods in Water Resources, Chapel Hill, 2004. Elsevier, p. 95–106.
16. Wolfram S. 1986. Cellular automaton fluids. 1: Basic theory. J. Stat. Phys. 45:471–526
17. Lavallee P, Boon JP, Noullez A. 1991. Boundaries in lattice gas flows. Physica D 47:233–40 Luo L

18. Cornubert R, d'Humieres D, Levermore D.1991. A Knudsen layer theory for lattice gases. Physica D 47:241–59
19. Ziegler DP. 1993. Boundary conditions for lattice Boltzmann simulations. J. Stat. Phys. 71:1171–77
20. Ginzbourg I, Adler PM. 1994. Boundary flow condition analysis for the three-dimensional lattice Boltzmann model. J. Phys. II 4:191–214
21. Skordos PA. 1993. Initial and boundary conditions for the lattice Boltzmann method. Phys. Rev. E 48:4823–42
22. Noble DR, Chen S, Georgiadis JG, Buckius RO.1995. A consistent hydrodynamic boundary condition for the lattice Boltzmann method. Phys. Fluids 7:203–9
23. Maier RS, Bernard RS, Grunau DW. 1996. Boundary conditions for the lattice Boltzmann method. Phys. Fluids 8:1788–1801
24. Zou Q, He X. 1997. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. Phys. Fluids. 9:1591–98
25. Chen S, Martinez D, Mei R. 1996. On boundary conditions in lattice Boltzmann methods. Phys. Fluids 8:2527–36
26. Cao N, Chen S, Jin S, Martinez D. 1997. Physical symmetry and lattice symmetry in lattice Boltzmann method. Phys. Rev. E 55:R21–24
27. Hou S, Sterling J, Chen S, Doolen GD. 1996. A lattice Boltzmann subgrid model for high Reynolds number flows. Fields Inst. Comm. 6:151–66
28. Ancona MG. 1994. Fully-Lagrangian and lattice-Boltzmann methods for solving systems of conservation equations. J. Comp.Phys. 115:107–20
29. Jin S, Xin ZP. 1995. The relaxation schemes for systems of conservation laws in arbitrary space dimensions. Comm. Pure Appl. Math.48:235–76.
30. Nadiga BT, Pullin DI. 1994. A method for near-equilibrium discrete-velocity gas flows. J. Comp. Phys. 112:162–72
31. Nadiga BT. 1995. An Euler solver based on locally adaptive discrete velocities. J. Stat. Phys. 81:129–46
32. Elton BH, Levermore CD, Rodrigue H. 1995. Covergence of convective-diffusive lattice Boltzmann methods. *SIAM J. Sci. Comp.*32:1327–54
33. Jianhui Yang, Edo S.Boek Paore Scale Simulation of Flow in Porous Media using the lattice-Boltzmann method. (2011), IFP Energies nouvelles (France).
34. Bo Han, Hua Meng.Numerical studies of interfacial phenomena in liquid water transport in polymer electrolyte membrane fuel cells using the lattice Boltzmann method. (2012).
35. Kyu-Jin Lee, Jin Hyun Nam, Charn-Jung Kim. Pore-Network analysis of two-phase water transport in gas diffusion layers of polymer electrolyte membrane fuel cells. Electrochimica Acta 54(2009), 1166-1176.

36. Mukherjee, Partha P., Chao-Yang Wang, and Qinjun Kang. "Mesoscopic modeling of two-phase behavior and flooding phenomena in polymer electrolyte fuel cells." Electrochimica Acta 54.27 (2009): 6861-6875.
37. Xiaoyi He, and Li-Shi Luo. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation.Physical Review E, Volume 56, Number 6.
38. Xiaowen shan, xiaoyi He. Discretization of the Velocity Space in the Solution of the Boltzmann Equation. Physical review letters, Volume 80, number 1.
39. Shiyi Chen, Gary D. Doolen. Lattice Boltzmann Method for Fluid Flows. Annu.Rev. Fluid Mech, 1998.
40. Lattice Gas Hydrodynamics in two and three Dimensions. Complex systems (1987) 649-707.
41. Advanced Engineering Mathematics, Greenberg (second edition)
42. E.Erturk, T.C.Corke. Numerical Solutions of 2-D steady Incompressible Driven Cavity Flow at High Reynolds Numbers. Int. J. Numer. Meth. Fluids 2005; Vol 48: pp747-774.
43. Shiyi chen , Daniel Martinez, Renwei Mei. On boundary conditions in lattice Boltzmann methods. Phys.Fluid 8 (9), 1996, 2527-2536.
44. John D.Anderson. Modern Compressible Flow: With Historical Perspective. 3rd Edition, 2002.
45. Michel A.Saad. Compressible Fluid Flow. 2nd edition, by Prentice hall.
46. Yan Guangwu, Chen Yaosong. And Hu Shouxin, 1999. Simple lattice Boltzmann model for simulating flows with shock wave. Physical Review E, Volume 59, Number 1.
47. Sauro Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Clarendon Press, Oxford. 2001.
48. Hou, Shuling, Lattice Boltzmann method for incompressible, viscous flow, PhD dissertation, Kansas State University.

APPENDIX A: Derivation of the Navier-Stokes equations from the Lattice Boltzmann BGK equation

1. Lattice geometry

Let's construct our developments on a grid with a square lattice and unit spacing, in which each node is connected to its immediate neighbors with 8 links (see Figure.1). Particles can only reside on the nodes and move to their nearest neighbors along these links at each unit time step.

From the schematic of Figure.1 we can distinguish three types of particles [48]:

- Rest particles with speed zero on each node.
- Particles of type 1 moving along the vertical and horizontal axes with speed $|e_{1i}|=1$.
- Particles of type 2 moving along the diagonal directions with speed $|e_{2i}| = \sqrt{2}$

The velocity vectors $e_{1,i}$ and $e_{2,i}$ for the moving particles are defined as:

$$e_{1,i} = \left(\cos\frac{i-1}{2}\pi, \sin\frac{i-1}{2}\pi\right) \qquad i = 1,..,4 \qquad \text{(A1.1)}$$

$$e_{2,i} = \sqrt{2}\left(\cos\left(\frac{i-1}{2}\pi + \frac{\pi}{4}\right), \sin\left(\frac{i-1}{2}\pi + \frac{\pi}{4}\right)\right) \qquad i = 1,..,4 \qquad \text{(A1.2)}$$

Further down in the derivations, the tensors $\sum_i(e_{\sigma i\alpha}e_{\sigma i\beta}\ldots.)$ (where $\alpha, \beta \ldots = 1$ or 2 denote the components of $e_{\sigma i}$) are needed. They are developed below.

The odd orders of tensors are equal to zero:

$$\sum_i e_{\sigma i\alpha} = 0 \qquad \text{(A1.3)}$$

$$\sum_i e_{\sigma i\alpha}e_{\sigma i\beta}e_{\sigma i\gamma} = 0 \qquad \text{(A1.4)}$$

$$\sum_i e_{\sigma i\alpha}e_{\sigma i\beta}e_{\sigma i\gamma}e_{\sigma i\theta}e_{\sigma i\epsilon} = 0 \qquad \text{(A1.5)}$$

With $\begin{cases} \alpha, \beta, \gamma, \theta, \epsilon = 1 \text{ or } 2 \\ i = 1,..,4 \\ \sigma = 1,2 \end{cases}$

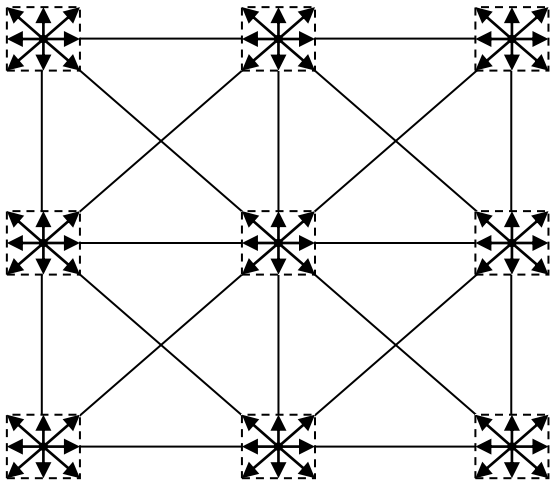


Fig A. 1: Schematic representation of the D2Q 9 Velocity Model on a square lattice

The second order Tensor:

$$\sum_i e_{\sigma i\alpha} e_{\sigma i\beta} = 2\, e_\sigma{}^2 \delta_{\alpha\beta} \qquad \text{(A1.6)} \quad (\alpha, \beta, \gamma, \delta, \epsilon = 1 \text{ or } 2)$$

Where $\delta_{\alpha\beta}$ is the kronecker delta: $\delta_{\alpha\beta} = \begin{cases} 1, & \alpha = \beta \\ 0, & \alpha \neq \beta \end{cases}$ (A1.7)

And e_σ is the length of $e_{\sigma i}$: $e_\sigma = \begin{cases} 1, & \sigma = 1 \\ \sqrt{2}, & \sigma = 2 \end{cases}$ (A1.8)

The third order Tensor:

$$\sum_i e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} e_{\sigma i\theta} = \begin{cases} 2\delta_{\alpha\beta\gamma\theta} & \sigma = 1 \\ 4\Delta_{\alpha\beta\gamma\theta} - 8\delta_{\alpha\beta\gamma\theta} & \sigma = 2 \end{cases} \qquad \text{(A1.9)}$$

where $\delta_{\alpha\beta\gamma\theta} = \begin{cases} 1, & \alpha = \beta = \gamma = \theta \\ 0, & \text{otherwise} \end{cases}$ (A1.10)

and $\Delta_{\alpha\beta\gamma\theta} = (\delta_{\alpha\beta}\delta_{\gamma\theta} + \delta_{\alpha\gamma}\delta_{\beta\theta} + \delta_{\alpha\theta}\delta_{\beta\gamma})$ (A1.11)

2. Derivation of the Navier-Stokes Equations

The particle distribution function satisfies the following lattice Boltzmann equation with the collision operator simplified using the single time relaxation approximation and the BGK model [47]:

$$f_{\sigma i}(x + e_{\sigma i}, t + 1) = f_{\sigma i}(x, t) + \omega\left(f_{\sigma i}^{eq}(x, t) - f_{\sigma i}(x, t)\right) \qquad (A1.12)$$

Where $\omega = \frac{1}{\tau}$ and τ is the the relaxation time that control the rate of approach to the equilibrium. Also σ represent the type of velocity and i its direction (defined later on in the development).

The following constraints are imposed on the distribution function:

$$\sum_\sigma \sum_i f_{\sigma i}(x, t) = \rho \qquad (A1.13)$$

$$\sum_\sigma \sum_i e_{\sigma i\alpha} f_{\sigma i}(x, t) = \rho u_\alpha \qquad (A1.14)$$

Where $f_{\sigma i}$ is the distribution function, $f_{\sigma i}^{eq}$ is its equilibrium.

We Taylor expand the first term of equation (A1.12):

$$f_{\sigma i}(x + e_i, t + 1) = f_{\sigma i}(x, t) + e_{\sigma i\alpha}\frac{\partial f_{\sigma i}(x,t)}{\partial x_\alpha} + \frac{\partial f_{\sigma i}(x,t)}{\partial t} + \frac{1}{2!}\left(e_{\sigma i\alpha}\frac{\partial}{\partial x_\alpha}\left(e_{\sigma i\beta}\frac{\partial f_{\sigma i}(x,t)}{\partial x_\beta}\right) + \frac{\partial^2 f_{\sigma i}(x,t)}{\partial t^2} + 2e\sigma i\alpha\partial\partial x\alpha\partial f\sigma ix,t\partial t\right) \qquad (A1.15)$$

Where α and β represent the Cartesian coordinate (The summation is implied for repeated indices).

If we substitute the left hand side of equation (A1.15) using equation (A1.12) we get:

$$\Omega\left(f_{\sigma i}^{eq}(x,t)-f_{\sigma i}(x,t)\right)=e_{\sigma i\alpha}\frac{\partial f_{\sigma i}(x,t)}{\partial x_\alpha}+\frac{\partial f_{\sigma i}(x,t)}{\partial t}+\frac{1}{2}\left(e_{\sigma i\alpha}\frac{\partial}{\partial x_\alpha}\left(e_{\sigma i\beta}\frac{\partial f_{\sigma i}(x,t)}{\partial x_\beta}\right)+\frac{\partial^2 f_{\sigma i}(x,t)}{\partial t^2}+\right.$$

$$2e\sigma i\alpha\partial\partial x\alpha\partial f\sigma ix,t\partial t \qquad \text{A1.16}$$

We can now apply the Chapman-Enskog expansion as follows:

$$\begin{cases} f_{\sigma i}(x,t)=f_{\sigma i}^{0}(x,t)+\varepsilon f_{\sigma i}^{1}(x,t)+O(\varepsilon^2) \\ \frac{\partial}{\partial t}=\varepsilon\frac{\partial}{\partial t_1}+\varepsilon^2\frac{\partial}{\partial t_2} \\ \frac{\partial}{\partial x}=\varepsilon\frac{\partial}{\partial x_1} \end{cases} \qquad \text{(A1.17)}$$

Equation (A1.16) becomes: $\omega\left(f_{\sigma i}^{eq}(x,t)-f_{\sigma i}(x,t)\right)=e_{\sigma i\alpha}\varepsilon\frac{\partial f_{\sigma i}(x,t)}{\partial x_{1\alpha}}+\left(\varepsilon\frac{\partial f_{\sigma i}(x,t)}{\partial t_1}+\right.$

$\varepsilon 2\partial f\sigma ix,t\partial t2+12e\sigma i\alpha\varepsilon\partial\partial x1\alpha e\sigma i\beta\varepsilon\partial f\sigma ix,t\partial x1\beta+\varepsilon\partial\partial t1\varepsilon\partial f\sigma ix,t\partial t1+\varepsilon 2\partial f\sigma ix,t\partial t2+\varepsilon 2\partial\partial t2$

$\varepsilon\partial f\sigma ix,t\partial t1+\varepsilon 2\partial f\sigma ix,t\partial t2+2e\sigma i\alpha\varepsilon\partial\partial x1\alpha\varepsilon\partial f\sigma ix,t\partial t1+\varepsilon 2\partial f\sigma ix,t\partial t2$ (A1.18)

With: $f_i(x,t)=f_i^{(0)}(x,t)+\varepsilon f_i^{(1)}(x,t)+\varepsilon^2 f_i^{(2)}(x,t)$ (A1.19)

Now we proceed to consider different levels of approximations to (A1.18):

Zeroth approximation of (A1.18) in ε : $\omega\left(f_{\sigma i}^{eq}(x,t)-f_{\sigma i}^{0}(x,t)\right)=0$ implies:

$$f_{\sigma i}^{eq}(x,t)=f_{\sigma i}^{0}(x,t) \qquad \text{(A1.20)}$$

First order approximation of (A1.18) in ε combined with the result of equation (A1.20) gives us:

$$-\omega f_{\sigma i}^{(1)}(x,t) = \frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial t_1} + e_{\sigma i\alpha}\frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\alpha}} \qquad \text{(A1.21)}$$

Summation of equation (A1.21) results in:

$$\sum_\sigma \sum_i \left(-\omega f_{\sigma i}^{(1)}(x,t) = \frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial t_1} + e_{\sigma i\alpha}\frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\alpha}}\right) \qquad \text{(A1.22)}$$

The first term is zero to satisfy the conservation of mass:

$$\frac{\partial \rho}{\partial t_1} + \frac{\partial \rho u_\alpha}{\partial x_{1\alpha}} = 0 \qquad \text{(A1.23)}$$

Let's multiply (A1.21) by $e_{\sigma i\beta}$:

$$-\omega e_{\sigma i\beta} f_{\sigma i}^{(1)}(x,t) = \frac{\partial e_{\sigma i\beta} f_{\sigma i}^{(0)}(x,t)}{\partial t_1} + e_{\sigma i\beta} e_{\sigma i\alpha}\frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\alpha}} \qquad \text{(A1.24)}$$

Summation of (A1.24) results in:

$$\sum_\sigma \sum_i \left(\frac{\partial e_{\sigma i\beta} f_{\sigma i}^{(0)}(x,t)}{\partial t_1} + e_{\sigma i\beta} e_{\sigma i\alpha}\frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\alpha}}\right) = 0 \qquad \text{(A1.25)}$$

Second order approximation of (A1.18) in ε :

$$-\omega f_{\sigma i}^{(2)}(x,t) = \frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial t_2} + e_{\sigma i\alpha}\frac{\partial f_{\sigma i}^{(1)}(x,t)}{\partial x_{1\alpha}} + \frac{\partial f_{\sigma i}^{(1)}(x,t)}{\partial t_1} + \frac{1}{2}\left(e_{\sigma i\alpha}\frac{\partial}{\partial x_{1\alpha}}\left(e_{\sigma i\beta}\frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\beta}}\right) + \frac{\partial^2 f_{\sigma i}^{(0)}(x,t)}{\partial {t_1}^2} +$$

$$2e\sigma i\alpha\partial\partial x1\alpha\partial f\sigma i(0)x,t\partial t1 \qquad \text{(A1.26)}$$

Reorganizing (A1.26) results in:

$$\frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial t_2} + \frac{\partial f_{\sigma i}^{(1)}(x,t)}{\partial t_1} + e_{\sigma i\alpha}\frac{\partial f_{\sigma i}^{(1)}(x,t)}{\partial x_{1\alpha}} + \frac{1}{2}\frac{\partial}{\partial t_1}\left(\frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial t_1} + e_{\sigma i\alpha}\frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\alpha}}\right) + \frac{1}{2}\frac{\partial}{\partial x_{1\alpha}}\left(e_{\sigma i\alpha}\frac{\partial f_{\sigma i}^{(0)}(x,t)}{\partial t_1} +\right.$$

$$e\sigma i\alpha e\sigma i\beta\partial f\sigma i(0)x,t\partial x1\beta = -\omega f\sigma i(2)x,t \qquad \text{(A1.27)}$$

By summing (A1.27) over i and σ we find that the second, third term and the right hand side are zero. The fourth and fifth terms are zero by using equation (A1.23) and (A1.24) consecutively, hence:

$$\frac{\partial \rho}{\partial t_2} = 0 \qquad \text{(A1.28)}$$

By combining equations (A1.23) x ε and (A1.28) x ε^2 we get the mass conservation equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_\alpha}{\partial x_\alpha} = 0 \qquad \text{(A1.29)}$$

Multiplying (A1.27) by $e_{\sigma i\gamma}$:

$$\frac{\partial\, e_{\sigma i\gamma} f_{\sigma i}^{(0)}(x,t)}{\partial t_2} + \frac{\partial\, e_{\sigma i\gamma} f_{\sigma i}^{(1)}(x,t)}{\partial t_1} + \frac{\partial\, e_{\sigma i\gamma} e_{\sigma i\alpha} f_{\sigma i}^{(1)}(x,t)}{\partial x_{1\alpha}} + \frac{1}{2}\frac{\partial}{\partial t_1}\left(\frac{\partial\, e_{\sigma i\gamma} f_{\sigma i}^{(0)}(x,t)}{\partial t_1} + \frac{\partial\, e_{\sigma i\gamma} e_{\sigma i\alpha} f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\alpha}}\right) +$$

$$\frac{1}{2}\frac{\partial}{\partial x_{1\alpha}}\left(\frac{\partial\, e_{\sigma i\gamma} e_{\sigma i\alpha} f_{\sigma i}^{(0)}(x,t)}{\partial t_1} + \frac{\partial\, e_{\sigma i\gamma} e_{\sigma i\alpha}\, e_{\sigma i\beta} f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\beta}}\right) = -\omega\, e_{\sigma i\gamma} f_{\sigma i}^{(2)}(x,t) \qquad \text{(A1.30)}$$

By summing (A1.30) over i and σ , the second and fourth terms are zero by momentum conservation and equation (A1.24). The right hand side is also zero by momentum conservation.

Equation (A1.30) reduces to:

$$\frac{\partial \rho\, u_\gamma}{\partial t_2} + \sum_\sigma \sum_i \frac{\partial\, e_{\sigma i\gamma} e_{\sigma i\alpha} f_{\sigma i}^{(1)}(x,t)}{\partial x_{1\alpha}} +$$

$$\frac{1}{2}\sum_\sigma \sum_i \frac{\partial}{\partial x_{1\alpha}}\left(\frac{\partial\, e_{\sigma i\gamma} e_{\sigma i\alpha} f_{\sigma i}^{(0)}(x,t)}{\partial t_1} + \frac{\partial\, e_{\sigma i\gamma} e_{\sigma i\alpha}\, e_{\sigma i\beta} f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\beta}}\right) = 0 \qquad \text{(A1.31)}$$

The second term is obtained by multiplying equation (A1.21) by $e_{i\gamma} e_{i\alpha}$ and summing:

$$\sum_\sigma \sum_i -\omega\, e_{\sigma i\gamma} e_{\sigma i\alpha} f_{\sigma i}^{(1)}(x,t) =$$

$$\sum_\sigma \sum_i \frac{\partial\, e_{\sigma i\gamma} e_{\sigma i\alpha} f_{\sigma i}^{(0)}(x,t)}{\partial t_1} + \sum_\sigma \sum_i e_{\sigma i\beta} \frac{\partial\, e_{\sigma i\gamma} e_{\sigma i\alpha} f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\beta}} \qquad \text{(A1.32)}$$

We substitute (A1.32) into (A1.31) to get: $\frac{\partial \rho\, u_\gamma}{\partial t_2} + (\frac{1}{2} - \frac{1}{\omega}) \sum_\sigma \sum_i \frac{\partial^2 e_{\sigma i\alpha} e_{\sigma i\gamma} f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\alpha}\, \partial t_1} +$

$(\frac{1}{2} - \frac{1}{\omega}) \sum_\sigma \sum_i \frac{\partial^2 e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\alpha}\, \partial x_{1\beta}} = 0$ (A1.33)

We can rewrite (A1.33) in the following form: $\frac{\partial \rho\, u_\alpha}{\partial t_2} + \left(\frac{1}{2} - \frac{1}{\omega}\right) \sum_\sigma \sum_i \frac{\partial^2 e_{\sigma i\alpha} e_{\sigma i\beta} f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\beta}\, \partial t_1} +$

$\left(\frac{1}{2} - \frac{1}{\omega}\right) \sum_\sigma \sum_i \frac{\partial^2 e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\beta}\, \partial x_{1\gamma}} = 0$ (A1.34)

The summations in equation (A1.34) can be determined by determining the following sums:

$$\begin{cases} \prod_{\alpha\beta}^{(1)} = \sum_\sigma \sum_i e_{\sigma i\alpha} e_{\sigma i\beta} f_{\sigma i}^{(0)}(x,t) & \text{(A1.35)} \\ \prod_{\alpha\beta}^{(2)} = \sum_\sigma \sum_i \frac{\partial e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} f_{\sigma i}^{(0)}(x,t)}{\partial x_{1\gamma}} & \text{(A1.36)} \\ \prod_{\alpha\beta}^{(3)} = \sum_\sigma \sum_i \frac{\partial\, e_{\sigma i\beta} e_{\sigma i\alpha} f_{\sigma i}^{(0)}(x,t)}{\partial t_1} & \text{(A1.37)} \end{cases}$$

At this step we need to assume a form to the equilibrium function and then determine its parameters that lead us to retrieve the Euler and the NS equations.

Let's assume an equilibrium function that takes the following general form:

$$f_{i\sigma}^{(0)}(x,t) = A_\sigma + B_\sigma(e_{i\sigma}.u) + C_\sigma(e_{i\sigma}.u)^2 + D_\sigma u^2 \qquad \text{(A1.38)}$$

Where $A_\sigma, B_\sigma, C_\sigma$ and D_σ are coefficients to be determined that depend on ρ, but not on u. It can be thought of as a special type of small velocity expansion of $f_i^{(eq)}$ including terms up to u^2. We can see that $B_0 = C_0 = 0$ for the rest particles.

Recalling the constraints on the distribution function:

$$\begin{cases} \sum_\sigma \sum_i f_{\sigma i}^{(0)}(x,t) = \rho \\ \sum_\sigma \sum_i e_{\sigma i}.\, f_{\sigma i}^{(0)}(x,t) = \rho u_\alpha \end{cases} \quad \text{(A1.39)}$$

and

$$\begin{cases} \sum_\sigma \sum_i f_{\sigma i}^{(n)}(x,t) = 0 \\ \sum_\sigma \sum_i e_{\sigma i} f_{\sigma i}^{(n)}(x,t) = 0 \end{cases} \quad n \geq 1 \quad \text{(A1.40)}$$

These constraints imply that the non-equilibrium distributions do not contribute to the density and momentum values. Applying the first set of constraints on (A1.38) results in the following formulas:

$$A_0 + A_1 + A_2 = \rho \quad \text{(A1.41)}$$

$$D_0 + 4D_1 + 4D_2 + 2C_1 + 4C_2 = 0 \quad \text{(A1.42)}$$

$$2B_1 + 4B_2 = \rho \quad \text{(A1.43)}$$

Let's now work on the development of the term:

$$\prod_{\alpha\beta}^{(1)} = \sum_\sigma \sum_i e_{\sigma i\alpha} e_{\sigma i\beta} f_{\sigma i}^{(0)}(x,t) \quad \text{(A1.44)}$$

$$\prod_{\alpha\beta}^{(1)} = \sum_\sigma \sum_i e_{\sigma i\alpha} e_{\sigma i\beta} (A_\sigma + B_\sigma (e_{i\sigma}.u) + C_\sigma (e_{i\sigma}.u)^2 + D_\sigma u^2)$$

$$\prod_{\alpha\beta}^{(1)} = 2\, {e_\sigma}^2 \delta_{\alpha\beta} \sum_\sigma A_\sigma + \sum_\sigma C_\sigma . u_\gamma u_\theta \sum_i e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} e_{\sigma i\theta} + 2\, {e_\sigma}^2 \delta_{\alpha\beta} \sum_\sigma D_\sigma u^2$$

$$\prod_{\alpha\beta}^{(1)} = 2\, \delta_{\alpha\beta} (A_1 + 2A_2) + \left(2C_1 \delta_{\alpha\beta\gamma\theta} + 4C_2 \Delta_{\alpha\beta\gamma\theta} - 8C_2 \delta_{\alpha\beta\gamma\theta}\right) u_\gamma u_\theta + 2\, \delta_{\alpha\beta} (D_1 + 2D_2) u^2$$

And

$$\begin{cases} \delta_{\alpha\beta\gamma\theta} u_\gamma u_\theta = \delta_{\alpha\beta} u_\alpha u_\beta \\ \Delta_{\alpha\beta\gamma\theta} u_\gamma u_\theta = (\delta_{\alpha\beta}\delta_{\gamma\theta} + \delta_{\alpha\gamma}\delta_{\beta\theta} + \delta_{\alpha\theta}\delta_{\beta\gamma}) u_\gamma u_\theta = \delta_{\alpha\beta} u^2 + 2u_\alpha u_\beta \end{cases}$$

Hence using these two last equations:

$$\prod_{\alpha\beta}^{(1)} = 2\,\delta_{\alpha\beta}\,(A_1 + 2A_2) + \left(2C_1\delta_{\alpha\beta}u_\alpha u_\beta + 4C_2\delta_{\alpha\beta}u^2 + 8C_2u_\alpha u_\beta - 8C_2\delta_{\alpha\beta}u_\alpha u_\beta\right) + 2\,\delta_{\alpha\beta}\,(D_1 + 2D_2)u^2 \qquad \text{(A1.45)}$$

$$\prod_{\alpha\beta}^{(1)} = \delta_{\alpha\beta}[2\ (A_1 + 2A_2) + (4C_2 + 2D_1 + 4D_2)u^2] + 8C_2u_\alpha u_\beta + (2C_1 - 8C_2)\delta_{\alpha\beta}u_\alpha u_\beta \qquad \text{(A1.46)}$$

If we use (A1.46) in (A1.24) and try to identify it with the Euler equation we can see that the first term is the pressure term and the two others are the non linear terms. In order to obtain a velocity independent pressure (otherwise non physical solution), the coefficient of u^2 in the first term should be set to zero hence:

$$4C_2 + 2D_1 + 4D_2 = 0 \qquad \text{(A1.47)}$$

To have a Galilean invariance, the non-isotropic term is eliminated by choosing:

$$2C_1 - 8C_2 = 0 \qquad \text{(A1.48)}$$

Equation (A1.46) then becomes:

$$\prod_{\alpha\beta}^{(1)} = \delta_{\alpha\beta}[2\ (A_1 + 2A_2)] + 8C_2u_\alpha u_\beta \qquad \text{(A1.49)}$$

Assuming that:

$$8C_2 = \rho \qquad \text{(A1.50)}$$

And

$$2\ (A_1 + 2A_2) = c_s^2\rho \qquad \text{(A1.51)}$$

Where c_s^2 is the speed of sound, hence the final expression of (A1.49) is as follows:

$$\prod_{\alpha\beta}^{(1)} = \delta_{\alpha\beta}c_s^2\rho\ + \rho\,u_\alpha u_\beta \qquad \text{(A1.52)}$$

Substituting equation (A1.52) in (A1.46):

$$\frac{\partial\rho u_\alpha}{\partial t_1} + \frac{\partial\rho\,u_\alpha u_\beta}{\partial x_{1\beta}} = -\frac{\partial\rho c_s^2}{\partial x_{1\alpha}} \qquad \text{(A1.53)}$$

By summing equations (A1.53) × ε and (A1.28) × ε^2 we get:

$$\frac{\partial \rho u_\alpha}{\partial t} + \frac{\partial \rho\, u_\alpha u_\beta}{\partial x_\beta} = -\frac{\partial \rho c_s^2}{\partial x_\beta} \qquad (A1.54)$$

The third order Tensor:

$$\prod^{(2)}_{\alpha\beta} = \sum_i \frac{\partial e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} f^{(0)}_{\sigma i}(x,t)}{\partial x_{\sigma 1\gamma}}$$

Using the definition of the equilibrium function in equation (A1.38), we get:

$$\prod^{(2)}_{\alpha\beta} = \frac{\partial}{\partial x_{1\gamma}} \left(\sum_\sigma \sum_i e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} (A_\sigma + B_\sigma (e_{i\sigma}.u) + C_\sigma (e_{i\sigma}.u)^2 + D_\sigma u^2)\right)$$

$$\prod^{(2)}_{\alpha\beta} = \frac{\partial}{\partial x_{1\gamma}} \left(\sum_\sigma \sum_i \left[(e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} A_\sigma) + B_\sigma \left(e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} e_{\sigma i\theta} u_\theta \right) + C_\sigma\, e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma} (e_{i\sigma}.u)^2 + \right.\right.$$

$$\left.\left. e\sigma i\alpha e\sigma i\beta e\sigma i\gamma D\sigma u2 \right)\right)$$

The first, third and fourth terms are all equal to zero:

$$\prod^{(2)}_{\alpha\beta} = \frac{\partial}{\partial x_{1\gamma}} \left(\sum_\sigma B_\sigma u \sum_i \left(e_{\sigma i\alpha} e_{\sigma i\beta} e_{\sigma i\gamma}\, e_{\sigma i\theta} u_\theta \right)\right)$$

$$\prod^{(2)}_{\alpha\beta} = \frac{\partial}{\partial x_{1\gamma}} \left((2\delta_{\alpha\beta\gamma\theta} u_\theta B_1 + (4\Delta_{\alpha\beta\gamma\theta} u_\theta - 8\delta_{\alpha\beta\gamma\theta} u_\theta) B_2 \right)$$

$$\prod^{(2)}_{\alpha\beta} = \frac{\partial}{\partial x_{1\alpha}} (\delta_{\alpha\beta} u_\beta (2B_1 - 8B_2)) + 4 \frac{\partial}{\partial x_{1\gamma}} (\delta_{\alpha\beta} B_2 u_\gamma) + 4 \frac{\partial}{\partial x_{1\alpha}} \left(B_2 u_\beta\right) + 4 \frac{\partial}{\partial x_{1\beta}} (B_2 u_\alpha) \qquad (A1.55)$$

To maintain isotropy we impose:

$$2B_1 - 8B_2 = 0$$

Combining this last equation with equation (A1.43) we get:

$$B_1 = \frac{\rho}{3}, \qquad B_2 = \frac{\rho}{12},$$

Equation (A1.55) can be simplified to:

$$\prod^{(2)}_{\alpha\beta} = \frac{1}{3} \frac{\partial}{\partial x_{1\gamma}} (\delta_{\alpha\beta} \rho u_\gamma) + \frac{1}{3} \frac{\partial}{\partial x_{1\alpha}} \left(\rho u_\beta\right) + \frac{1}{3} \frac{\partial}{\partial x_{1\beta}} (\rho u_\alpha) \qquad (A1.56)$$

Let's work on the last remaining tensor:

$$\prod_{\alpha\beta}^{(3)} = \sum_i \sum_\sigma \frac{e_{\sigma i\alpha} e_{\sigma i\beta} f_{\sigma i}^{(0)}(x,t)}{\partial t_1} \qquad (A1.57)$$

Using equation (A1.52) we get:

$$\prod_{\alpha\beta}^{(3)} = \frac{\partial}{\partial t_1}\left(\delta_{\alpha\beta} c_s^2 \rho + \rho\, u_\alpha u_\beta\right) \qquad (A1.58)$$

By using equation (A1.23) we can rewrite the first term of equation (A1.58) as:

$$\prod_{\alpha\beta}^{(3)} = -\delta_{\alpha\beta} c_s^2 \frac{\partial \rho u_\gamma}{\partial x_{1\gamma}} + \frac{\partial}{\partial t_1}\left(\rho\, u_\alpha u_\beta\right) \qquad (A1.59)$$

By using equation (A1.53) and (A1.23) we can rewrite the second term of equation (A1.59) as: $\frac{\partial}{\partial t_1}\rho\, u_\alpha u_\beta = u_\beta \frac{\partial}{\partial t_1}\rho\, u_\alpha + \rho\, u_\alpha \frac{\partial}{\partial t_1} u_\beta$

$$= -u_\beta\left(\frac{\partial \rho c_s^2}{\partial x_{1\alpha}} + \frac{\partial \rho\, u_\alpha u_\gamma}{\partial x_{1\gamma}}\right) + u_\alpha \frac{\partial}{\partial t_1}\rho\, u_\beta - u_\alpha\, u_\beta \frac{\partial}{\partial t_1}\rho$$

$$= -u_\beta\left(\frac{\partial \rho c_s^2}{\partial x_{1\alpha}} + \frac{\partial \rho\, u_\alpha u_\gamma}{\partial x_{1\gamma}}\right) - u_\alpha\left(\frac{\partial \rho c_s^2}{\partial x_{1\beta}} + \frac{\partial \rho\, u_\gamma u_\beta}{\partial x_{1\gamma}}\right) + u_\alpha\, u_\beta \frac{\partial \rho u_\gamma}{\partial x_{1\gamma}}$$

$$= -u_\alpha \frac{\partial \rho c_s^2}{\partial x_{1\beta}} - u_\beta \frac{\partial \rho c_s^2}{\partial x_{1\alpha}} - u_\beta \frac{\partial \rho\, u_\alpha u_\gamma}{\partial x_{1\gamma}} - u_\alpha \frac{\partial \rho\, u_\gamma u_\beta}{\partial x_{1\gamma}} + u_\alpha\, u_\beta \frac{\partial \rho u_\gamma}{\partial x_{1\gamma}}$$

$$= -u_\alpha \frac{\partial \rho c_s^2}{\partial x_{1\beta}} - u_\beta \frac{\partial \rho c_s^2}{\partial x_{1\alpha}} - u_\beta u_\alpha \frac{\partial \rho u_\gamma}{\partial x_{1\gamma}} - \rho u_\beta u_\gamma \frac{\partial u_\alpha}{\partial x_{1\gamma}} - u_\alpha \frac{\partial \rho\, u_\gamma u_\beta}{\partial x_{1\gamma}} + u_\alpha\, u_\beta \frac{\partial \rho u_\gamma}{\partial x_{1\gamma}}$$

$$\frac{\partial}{\partial t_1}(\rho\, u_\alpha u_\beta) = -u_\alpha \frac{\partial \rho c_s^2}{\partial x_{1\beta}} - u_\beta \frac{\partial \rho c_s^2}{\partial x_{1\alpha}} - \frac{\partial \rho\, u_\alpha u_\beta u_\gamma}{\partial x_{1\gamma}} \qquad (A1.60)$$

By substituting equation (A1.60) into (A1.61):

$$\prod_{\alpha\beta}^{(3)} = -\delta_{\alpha\beta} c_s^2 \frac{\partial \rho u_\gamma}{\partial x_{1\gamma}} - u_\alpha \frac{\partial \rho c_s^2}{\partial x_{1\beta}} - u_\beta \frac{\partial \rho c_s^2}{\partial x_{1\alpha}} - \frac{\partial \rho\, u_\alpha u_\beta u_\gamma}{\partial x_{1\gamma}} \qquad (A1.61)$$

Now let's combine equations (A1.34), (A1.56) and (A1.61):

$$\frac{\partial \rho\, u_\alpha}{\partial t_2} + \left(\frac{1}{2} - \frac{1}{\omega}\right)\frac{\partial}{\partial x_{1\beta}}\left(-\delta_{\alpha\beta} c_s^2 \frac{\partial \rho u_\gamma}{\partial x_{1\gamma}} - u_\alpha \frac{\partial \rho c_s^2}{\partial x_{1\beta}} - u_\beta \frac{\partial \rho c_s^2}{\partial x_{1\alpha}} - \frac{\partial \rho\, u_\alpha u_\beta u_\gamma}{\partial x_{1\gamma}}\right) + \frac{1}{3}\left(\frac{1}{2} - \frac{1}{\omega}\right)\frac{\partial}{\partial x_{1\beta}}\left(\frac{\partial}{\partial x_{1\gamma}}(\delta_{\alpha\beta}\rho u_\gamma) + \frac{\partial}{\partial x_{1\alpha}}(\rho u_\beta) + \frac{\partial}{\partial x_{1\beta}}(\rho u_\alpha)\right) = 0 \qquad (A1.62)$$

By combining equations (A1.53) × ε and (A1.64) × ε^2 we get the NS equation:

$$\frac{\partial \rho\, u_\alpha}{\partial t}+\frac{\partial \rho\, u_\alpha u_\beta}{\partial x_\beta}+\frac{\partial \rho c_s^2}{\partial x_\alpha}+\varepsilon\left(\frac{1}{2}-\frac{1}{\omega}\right)\frac{\partial}{\partial x_\beta}\left(-\delta_{\alpha\beta}c_s^2\frac{\partial \rho u_\gamma}{\partial x_\gamma}-u_\alpha\frac{\partial \rho c_s^2}{\partial x_\beta}-u_\beta\frac{\partial \rho c_s^2}{\partial x_\alpha}-\frac{\partial \rho\, u_\alpha u_\beta u_\gamma}{\partial x_\gamma}\right)+$$

$$\varepsilon\frac{1}{3}\left(\frac{1}{2}-\frac{1}{\omega}\right)\frac{\partial}{\partial x_\beta}\left(\frac{\partial}{\partial x_\gamma}(\delta_{\alpha\beta}\rho u_\gamma)+\frac{\partial}{\partial x_\alpha}(\rho u_\beta)+\frac{\partial}{\partial x_\beta}(\rho u_\alpha)\right)+O(\varepsilon^2)=0 \qquad \text{(A1.63)}$$

We can rewrite it to the form:

$$\frac{\partial \rho\, u_\alpha}{\partial t}+\frac{\partial \rho\, u_\alpha u_\beta}{\partial x_\beta}=-\frac{\partial \rho c_s^2}{\partial x_\alpha}+\varepsilon\left(\frac{1}{\omega}-\frac{1}{2}\right)\frac{\partial}{\partial x_\beta}\left\{\left(\frac{1}{3}-c_s^2\right)\frac{\partial}{\partial x_\gamma}(\rho u_\gamma)\delta_{\alpha\beta}+\frac{1}{3}\frac{\partial}{\partial x_\alpha}(\rho u_\beta)+\frac{1}{3}\frac{\partial}{\partial x_\beta}(\rho u_\alpha)-\right.$$

$$u\alpha\partial\rho cs2\partial x\beta-u\beta\partial\rho cs2\partial x\alpha-\partial\rho\, u\alpha u\beta u\gamma\partial x\gamma+O\varepsilon 2$$

(A1.64)

Equation (A.163) can also take the following form:

$$\frac{\partial \rho\, u_\alpha}{\partial t}+\frac{\partial \rho\, u_\alpha u_\beta}{\partial x_\beta}=-\frac{\partial \rho c_s^2}{\partial x_\alpha}+\varepsilon\left\{\left[\left(\frac{1}{3}-c_s^2\right)\left(\frac{1}{\omega}-\frac{1}{2}\right)\frac{\partial}{\partial x_\gamma}(\rho u_\gamma)\right]+\left(\frac{1}{\omega}-\frac{1}{2}\right)\frac{\partial}{\partial x_\beta}\left[\frac{1}{3}\rho\left(\frac{\partial u_\beta}{\partial x_\alpha}+\frac{\partial u_\alpha}{\partial x_\beta}\right)+\right.\right.$$

$$13-cs2u\alpha\partial\rho\partial x\beta+u\beta\partial\rho\partial x\alpha-\partial\rho\, u\alpha u\beta u\gamma\partial x\gamma+O\varepsilon 2 \qquad \text{(A1.65)}$$

Considering the constraints on A_σ in equations (A1.41) and (A1.51) and choosing:

$$A_0=\frac{4}{9}\rho, \qquad A_1=\frac{1}{9}\rho, \qquad A_2=\frac{1}{36}\rho.$$

We use equation (A1.41) to determine the speed of sound:

$$c_s^2=\frac{1}{3}$$

Then we can simplify equation (A1.65) to the form:

$$\frac{\partial \rho\, u_\alpha}{\partial t}+\frac{\partial \rho\, u_\alpha u_\beta}{\partial x_\beta}=-\frac{\partial \rho c_s^2}{\partial x_\alpha}+\varepsilon\left(\frac{1}{\omega}-\frac{1}{2}\right)\frac{\partial}{\partial x_\beta}\left[\frac{1}{3}\rho\left(\frac{\partial u_\beta}{\partial x_\alpha}+\frac{\partial u_\alpha}{\partial x_\beta}\right)-\frac{\partial \rho\, u_\alpha u_\beta u_\gamma}{\partial x_\gamma}\right]+O(\varepsilon^2) \qquad \text{(A1.66)}$$

We plug in for the relaxation time ($\omega=\frac{1}{\tau}$) and reorganize Equation (A1.66):

$$\frac{\partial \rho\, u_\alpha}{\partial t}+\frac{\partial \rho\, u_\alpha u_\beta}{\partial x_\beta}=-\frac{\partial \rho c_s^2}{\partial x_\alpha}+\frac{\partial}{\partial x_\beta}\left[2\upsilon\rho S_{\alpha\beta}\right]+\varepsilon\left(\tau-\frac{1}{2}\right)\frac{\partial}{\partial x_\beta}\left[\frac{\partial \rho\, u_\alpha u_\beta u_\gamma}{\partial x_\gamma}\right]+O(\varepsilon^2) \qquad \text{(A1.67)}$$

Where $\upsilon = \varepsilon\left(\frac{2\tau-1}{6}\right)$ represent the kinematic viscosity and $S_{\alpha\beta} = \frac{1}{2}\left(\frac{\partial u_\beta}{\partial x_\alpha} + \frac{\partial u_\alpha}{\partial x_\beta}\right)$ is the strain rate tensor.

Notice that by omitting the last term $O(\varepsilon u^3)$ and $O(\varepsilon^2)$, equation (A1.68) is exactly the incompressible Navier-Stokes equation.

Collecting all the calculated coefficients we have:

$$A_0 = \frac{4}{9}\rho, \qquad A_1 = \frac{1}{9}\rho, \qquad A_2 = \frac{1}{36}\rho.$$

$$B_1 = \frac{\rho}{3}, \qquad B_2 = \frac{\rho}{12},$$

$$C_1 = \frac{\rho}{2}, \qquad C_2 = \frac{\rho}{8}.$$

The remaining three D_σ coefficients are related only by two equations (A1.42) and (A1.47), so we have one free parameter. We can follow the trend of the other parameters and assume that $B_2 = \frac{1}{4}B_1$, resulting in:

$$D_0 = -\frac{2}{3}\rho, \qquad D_1 = -\frac{1}{6}\rho, \qquad D_2 = -\frac{1}{24}\rho.$$

Finally we can write the equilibrium function as:

$$f_{i0}^{(0)}(x,t) = \frac{4}{9}\rho\left[1 - \frac{3}{2}\rho u^2\right] \qquad (A1.68)$$

$$f_{i1}^{(0)}(x,t) = \frac{1}{9}\rho\left[1 + 3(e_{i\sigma}.u) + \frac{9}{2}(e_{i\sigma}.u)^2 - \frac{3}{2}u^2\right] \qquad (A1.69)$$

$$f_{i2}^{(0)}(x,t) = \frac{1}{36}\rho\left[1 + 3(e_{i\sigma}.u) + \frac{9}{2}(e_{i\sigma}.u)^2 - \frac{3}{2}u^2\right] \qquad (A1.70)$$

The generalized equilibrium function for the D2Q9 model takes the following form:

$$f_i^{(0)}(x,t) = t_i\rho\left[1 + \frac{(e_i.u)}{c_s^2} + \frac{u_\alpha u_\beta}{2c_s^2}\left(\frac{e_{i\alpha} e_{i\beta}}{c_s^2} - \delta_{\alpha\beta}\right)\right] \qquad \text{(A1.71)}$$

Where:

$$t_i = \begin{cases} \frac{4}{9} & i = 0 \\ \frac{1}{9} & i = 1,4 \\ \frac{1}{36} & i = 5,8 \end{cases}$$

and the speed of sound is: $c_s^2 = \frac{1}{3}$

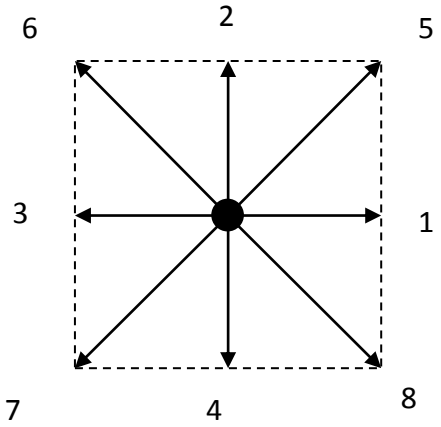


Fig A. 2: Lattice structure

APPENDIX B: Lattice Boltzmann Code

Code sample for the 2D GDL structure

```
! Computer code for 2D GDL layer
! Complete code that scans through the 3D GDL structure, create the 2D Layer
!   and the boundary condition coordinate vectors.Then run the LBM code and
!   create an ASCII file.

parameter (n=300,m=300,p=1)
real f(0:8,0:n,0:m)
real feq(0:8,0:n,0:m),rho(0:n,0:m)
real w(0:8), cx(0:8),cy(0:8)
real u(0:n,0:m), v(0:n,0:m)
real ue,uw,us,un,uo
real error,uemax,vemax
integer i,ii,jj
real be(2,n*3),bw(2,n*3),bs(2,n*3),bn(2,n*3)  !!!Porous Boundary Matrices.
real ke,kw,kn,ks                        !!!Increments for the porous boundary.

open(2,file='uvfield')
open(3,file='uvely')
open(4,file='vvelx')
open(6,file='Conv_Gen.dat')
open(7,file='Conv_Center.dat')
open(8,file='timeu')
!


uo=0.02
sumvelo=0.0
rhoo=7.00

!specify the velocities at the injection boundaries
ue=0.01
uw=ue
us=ue
un=ue

dx=1.0
dy=dx
dt=dx

alpha=0.02
Re=uo*m/alpha
print *, 'Re=', Re
```

```
omega=1.0/(3.*alpha+0.5)
print *, 'omega=', omega
pause

mstep=10000



w(0)=4./9.
do i=1,4
w(i)=1./9.
end do
do i=5,8
w(i)=1./36.
end do

cx(0)=0
cx(1)=1
cx(2)=0
cx(3)=-1
cx(4)=0
cx(5)=1
cx(6)=-1
cx(7)=-1
cx(8)=1
cy(0)=0
cy(1)=0
cy(2)=1
cy(3)=0
cy(4)=-1
cy(5)=1
cy(6)=1
cy(7)=-1
cy(8)=-1

do j=0,m
do i=0,n
do k=0,8
rho(i,j)=rhoo
f(k,i,j)=rhoo/9
end do
u(i,j)=0.0
v(i,j)=0.0
end do
end do

!!!Create the Porous Medium with the appropriate Boundary conditions
vector...
pp=100
!call BCs_Build(ii,jj,n,m,pp,be,bw,bs,bn)



!do i=1,n-1
!u(i,m)=uo
!v(i,m)=0.0
!end do
```

```
!!!!! files main comment
write (6,*),'Overal,kk,uemax,vemax,rhoemax'
write (7,*),'Center,kk,up(n/2,m/2),vp(n/2,m/2),rhop(n/2,m/2)'

!============================main loop==================================
1 do kk=1,mstep
call collesion(u,v,f,feq,rho,omega,w,cx,cy,n,m)
call streaming(f,n,m)
call sfbound(f,n,m,ue,uw,us,un,be,bw,bs,bn,ii,jj)
call rhouv(f,rho,u,v,cx,cy,n,m,error,kk,uemax,vemax,rhoemax)

END DO
! end of the main loop

call result(u,v,rho,uo,n,m)
stop
end
!================ end of the main program==============================

subroutine collesion(u,v,f,feq,rho,omega,w,cx,cy,n,m)
real f(0:8,0:n,0:m)
real feq(0:8,0:n,0:m),rho(0:n,0:m)
real w(0:8), cx(0:8),cy(0:8)
real u(0:n,0:m), v(0:n,0:m)

DO i=0,n
DO j=0,m
t1=u(i,j)*u(i,j)+v(i,j)*v(i,j)
DO k=0,8
t2=u(i,j)*cx(k)+v(i,j)*cy(k)
feq(k,i,j)=rho(i,j)*w(k)*(1.0+3.0*t2+4.50*t2*t2-1.50*t1)
f(k,i,j)=omega*feq(k,i,j)+(1.-omega)*f(k,i,j)
END DO
END DO
END DO
return
end


subroutine streaming(f,n,m)
real f(0:8,0:n,0:m)

! streaming
DO j=0,m
DO i=n,1,-1 !RIGHT TO LEFT
f(1,i,j)=f(1,i-1,j)
END DO

DO i=0,n-1 !LEFT TO RIGHT
```

```
f(3,i,j)=f(3,i+1,j)
END DO
END DO

DO j=m,1,-1 !TOP TO BOTTOM
DO i=0,n
f(2,i,j)=f(2,i,j-1)
END DO

DO i=n,1,-1
f(5,i,j)=f(5,i-1,j-1)
END DO

DO i=0,n-1
f(6,i,j)=f(6,i+1,j-1)
END DO
END DO

DO j=0,m-1 !BOTTOM TO TOP
DO i=0,n
f(4,i,j)=f(4,i,j+1)
END DO
DO i=0,n-1
f(7,i,j)=f(7,i+1,j+1)
END DO

DO i=n,1,-1
f(8,i,j)=f(8,i-1,j+1)
END DO
END DO
return
end

subroutine sfbound(f,n,m,ue,uw,us,un,be,bw,bs,bn,ii,jj)
real f(0:8,0:n,0:m)
real ue,uw,us,un
real be(2,n*3),bw(2,n*3),bs(2,n*3),bn(2,n*3)  !!!Porous Boundary Matrices.
real ke,kw,kn,ks                        !!!Increments for the porous boundary.

do j=0,m
! Imposed velocity on west boundary
rhow=(f(0,0,j)+f(2,0,j)+f(4,0,j)+2*(f(3,0,j)+f(6,0,j)+f(7,0,j)))/(1-uo)
f(1,0,j)=f(3,0,j)+(2/3)*rhow*uo
f(5,0,j)=f(7,0,j)+ rhow*uo/6
f(8,0,j)=f(6,0,j)+ rhow*uo/6
enddo
!do j=0,m
!! Bounce back on west boundary
!f(1,0,j)=f(3,0,j)
!f(5,0,j)=f(7,0,j)
!f(8,0,j)=f(6,0,j)
!enddo

do j=1,m
! Open Boundary condition on the east
f(6,n,j)= 2*f(6,n-1,j)-f(6,n-2,j)
f(3,n,j)= 2*f(3,n-1,j)-f(3,n-2,j)
```

```
f(7,n,j)= 2*f(7,n-1,j)-f(7,n-2,j)
enddo
!do j=0,m
!! Bounce back on the east
!f(6,n,j)= f(8,n,j)
!f(3,n,j)= f(1,n,j)
!f(7,n,j)= f(5,n,j)
!enddo
! bounce back on south boundary
do i=0,n
f(2,i,0)=f(4,i,0)
f(5,i,0)=f(7,i,0)
f(6,i,0)=f(8,i,0)
end do
! bounce back on north boundary
do i=0,n
f(4,i,m)=f(2,i,m)
f(8,i,m)=f(6,i,m)
f(7,i,m)=f(5,i,m)
end do
!! Open north boundary
!do i=0,n
!f(4,i,m)=2*f(4,i,m-1)-f(4,i,m-2)
!f(8,i,m)=2*f(8,i,m-1)-f(8,i,m-2)
!f(7,i,m)=2*f(7,i,m-1)-f(7,i,m-2)
!end do

!=========================================================
!========Automation of the Porous Medium Strategy ======
!=========================================================
do j=1,jj-1
! bounce back on west boundary

f(1,bw(1,j),bw(2,j))=f(3,bw(1,j),bw(2,j))
f(5,bw(1,j),bw(2,j))=f(7,bw(1,j),bw(2,j))
f(8,bw(1,j),bw(2,j))=f(6,bw(1,j),bw(2,j))

! bounce back on east boundary

f(3,be(1,j),be(2,j))=f(1,be(1,j),be(2,j))
f(7,be(1,j),be(2,j))=f(5,be(1,j),be(2,j))
f(6,be(1,j),be(2,j))=f(8,be(1,j),be(2,j))
enddo

! bounce back on south boundary
do j=1,ii-1
f(2,bs(1,j),bs(2,j))=f(4,bs(1,j),bs(2,j))
f(5,bs(1,j),bs(2,j))=f(7,bs(1,j),bs(2,j))
f(6,bs(1,j),bs(2,j))=f(8,bs(1,j),bs(2,j))

!bounce back on north boundary
f(4,bn(1,j),bn(2,j))=f(2,bn(1,j),bn(2,j))
f(8,bn(1,j),bn(2,j))=f(6,bn(1,j),bn(2,j))
f(7,bn(1,j),bn(2,j))=f(5,bn(1,j),bn(2,j))
enddo

!=============================================================
```

```
!========End of Automation of the Porous Medium Strategy======
!============================================================

return
end



subroutine rhouv(f,rho,u,v,cx,cy,n,m,error,kk,uemax,vemax,rhoemax)
real f(0:8,0:n,0:m),rho(0:n,0:m),u(0:n,0:m),v(0:n,0:m),cx(0:8),cy(0:8)
real rhop(0:n,0:m),up(0:n,0:m),vp(0:n,0:m),uemax,vemax,rhoemax,Error

!Save the previous velocities and density matrices
do j=0,m
    do i=0,n
    rhop(i,j)=rho(i,j)
    up(i,j)=u(i,j)
    vp(i,j)=v(i,j)
    enddo
enddo

do j=0,m
do i=0,n
ssum=0.0
do k=0,8
ssum=ssum+f(k,i,j)
end do
rho(i,j)=ssum
end do
end do

do i=1,n
rho(i,m)=f(0,i,m)+f(1,i,m)+f(3,i,m)+2.*(f(2,i,m)+f(6,i,m)+f(5,i,m))
end do

DO i=1,n
DO j=1,m-1
usum=0.0
vsum=0.0
DO k=0,8
usum=usum+f(k,i,j)*cx(k)
vsum=vsum+f(k,i,j)*cy(k)
END DO
u(i,j)=usum/rho(i,j)
v(i,j)=vsum/rho(i,j)
END DO
END DO

!Calculate the Errors

do j=0,m
    do i=0,n
    rhop(i,j)=(rhop(i,j)-rho(i,j))/rhop(i,j)
    up(i,j)=(u(i,j)-up(i,j))/up(i,j)
    vp(i,j)=(v(i,j)-vp(i,j))/vp(i,j)
    enddo
enddo
```

```
 rhoemax= maxval(abs(rhop))!, mask = numbers.lt.100)
 uemax = maxval(abs(up))!,    mask = numbers.lt.100)
 vemax = maxval(abs(vp))!,    mask = numbers.lt.100)
 Error= max(uemax,vemax)
!write (*,*),'Overal',kk,uemax,vemax,rhoemax
write (*,*),'Center',kk,u(n/2,m/2),v(n/2,m/2),rho(n/2,m/2)
write (*,*)
write (*,*)
!write (*,*),'Up',u(n/2,m),v(n/2,m),rhop(n/2,m)
!write (*,*),'Base',u(n/2,0),v(n/2,0),rhop(n/2,0)
!write (*,*),'Right',u(n,m/2),v(n,m/2),rhop(n,m/2)
!write (*,*),'Left',u(0,m/2),v(0,m/2),rhop(0,m/2)
write (6,*),kk,uemax,vemax,rhoemax
write (7,*),kk,up(n/2,m/2),vp(n/2,m/2),rhop(n/2,m/2)

Do j=1,m
 v(n,j)=0
enddo

return
end

subroutine result(u,v,rho,uo,n,m)
real u(0:n,0:m),v(0:n,0:m)
real rho(0:n,0:m),strf(0:n,0:m)

open(5, file='streamf')
! streamfunction calculations
strf(0,0)=0.
do i=0,n
    if(i.ne.0)then
    rhoav=0.5*(rho(i-1,0)+rho(i,0))
    strf(i,0)=strf(i-1,0)-rhoav*0.5*(v(i-1,0)+v(i,0))
    endif
do j=1,m
rhom=0.5*(rho(i,j)+rho(i,j-1))
strf(i,j)=strf(i,j-1)+rhom*0.5*(u(i,j-1)+u(i,j))
end do
end do
! ------------
write(2,*)'VARIABLES =X, Y, U, V, S'
write(2,*)'ZONE ','I=',n+1,'J=',m+1,',','F=BLOCK'

do j=0,m
write(2,*)(u(i,j),i=0,n)
end do

do j=0,m
write(2,*)(v(i,j),i=0,n)
end do

do j=0,m
write(5,*)(strf(i,j),i=0,n)
end do

do j=0,m
```

```
write(3,*)j/float(m),u(n/2,j)/uo,u(n/4,j)/uo,u(3*n/4,j)/uo
end do




do i=0,n
write(4,*) i/float(n),v(i,m/2)/uo
end do


!Create the ASCII file for Tecplot Use

open(30, file='ASCII.dat')
    write(30,*) 'VARIABLES = "X", "Y", "U", "V","Density"'
    write(30,*) 'ZONE I=',n, ', J=', m, ',DATAPACKING=POINT'

      do J=1,m
      do I=1,n
           write(30,*) I, J, u(I,J), v(I,J), rho(I,J)
      enddo
      enddo

!End of ASCII File creation.

return
end


! computer code for Building the Porous Medium
Subroutine  BCs_Build(ii,jj,n,m,pp,be,bw,bs,bn)

integer i,j,k,kk,linend,idim,jdim,ipos
real ke,kw,kn,ks                        !!!Increments for the porous boundary.
real be(2,n*3),bw(2,n*3),bs(2,n*3),bn(2,n*3)  !!!Porous Boundary Matrices.
CHARACTER(LEN=n) :: line
Character(LEN=1) :: LiEl
Integer PorStru(0:n+1,0:m+1,0:pp+1),var,np,mp

open(10,file='NoPormed.dat')
open(11,file='SoPormed.dat')
open(12,file='EaPormed.dat')
open(13,file='WePormed.dat')

open(21,file='3D-2M-DM.txt')
open(20,file='3D-2M-DM_topo.txt')
open(30,file='ASCII_Porous_structure.txt')

np=140
mp=140
pp=80
ipos=50

!!!!!! Read the numeric porous medium File and build the
do k=1,pp
```

```
   write(20,*)'Horizental slice number is :',k
    do j=1,mp
        do i=1,np
            read(21,'(A)',iostat=linend)line
            LiEl=line(48:48)
            read(LiEl,'(I1)')PorStru(i,j,k)

            var=PorStru(i,j,k)
            if (var.eq.2) then
            PorStru(i,j,k)=1
            endif
        enddo
!        write(*,*)(PorStru(i,j,k),i=1,np)
        write(20,*)(PorStru(i,j,k),i=1,np)
!        Pause
    enddo
enddo

kn=0
ks=0
ke=0
kw=0
ii=0
jj=0

j=70
do i=1,np
    do k=1,pp
        If (PorStru(i,j,k)==1)then

            if(PorStru(i,j,k-1)==0)then
            kn=kn+1
            bn(1,kn)=k+ipos
            bn(2,kn)=i
            ii=ii+1
            endif

            if(PorStru(i,j,k+1)==0)then
            ks=ks+1
            bs(1,ks)=k+ipos
            bs(2,ks)=i
            ii=ii+1
            endif

            if(PorStru(i+1,j,k)==0)then
            ke=ke+1
            be(1,ke)=k+ipos
            be(2,ke)=i
            jj=jj+1
            endif

            if(PorStru(i-1,j,k)==0)then
            kw=kw+1
            bw(1,kw)=k+ipos
            bw(2,kw)=i
            jj=jj+1
            endif
```

```
        endif
    enddo
enddo



write(*,*),'North'
do i=1,kn
    write(*,*),bn(1,i),bn(2,i)
    write(10,*),bn(1,i),bn(2,i)
    write(30,*)'Geometry X=',bn(1,i),',Y=',bn(2,i),',T=CIRCLE, CS=GRID,
C=BLACK, FC=BLACK, F=POINT, S=GLOBAL'
    write(30,*)'0.1'
enddo

write(*,*),'South'
do i=1,ks
    write(*,*),bs(1,i),bs(2,i)
    write(11,*),bs(1,i),bs(2,i)
    write(30,*)'Geometry X=',bs(1,i),',Y=',bs(2,i),',T=CIRCLE, CS=GRID,
C=BLACK, FC=BLACK, F=POINT, S=GLOBAL'
    write(30,*)'0.1'
enddo

write(*,*),'East'
do i=1,ke
    write(*,*),be(1,i),be(2,i)
    write(12,*),be(1,i),be(2,i)
    write(30,*)'Geometry X=',be(1,i),',Y=',be(2,i),',T=CIRCLE, CS=GRID,
C=BLACK, FC=BLACK, F=POINT, S=GLOBAL'
    write(30,*)'0.1'
enddo

write(*,*),'West'
do i=1,kw
    write(*,*),bw(1,i),bw(2,i)
    write(13,*),bw(1,i),bw(2,i)
    write(30,*)'Geometry X=',bw(1,i),',Y=',bw(2,i),',T=CIRCLE, CS=GRID,
C=BLACK, FC=BLACK, F=POINT, S=GLOBAL'
    write(30,*)'0.1'
enddo

write(*,*),'The total number of Boundary points are:',jj+ii

pause
return
End
!============end of the program=====================================
```