Maximum Entropy Boundaries in Lattice Boltzmann Method

Vasili Baranau[1]

1. Department of Chemistry, Philipps-Universität Marburg, Hans-Meerwein-Strasse, 35032 Marburg, Germany

Corresponding author: Vasili Baranau, vasili.baranov@gmail.com

Abstract: We propose a universal approach in the framework of the lattice Boltzmann method (LBM) to modeling constant velocity constraints and constant temperature constraints on curved walls, which doesn't depend on dimensionality, LBM scheme, boundary geometry; which is numerically stable, accurate and local and has a good physical background. This technique, called a maximum entropy method, utilizes the idea of recovering unknown populations on boundary nodes through minimizing node state deviation from equilibrium while assuring velocity or temperature restrictions. Also, theoretical justifications of a popular Zou-He boundaries technique and isothermal boundaries algorithm are provided on the basis of the method derived. Finally, while conducting numerical benchmarks, typical straight boundaries algorithm (Zou-He) was compared to a typical curved boundaries algorithm (Guo-Zheng).

Keywords: Lattice Boltzmann method, boundary conditions, entropy maximization, Zou-He boundary algorithm, Guo-Zheng boundary algorithm.

Table of Contents

1. Introduction

Lattice Boltzmann method (LBM) is a powerful tool for fluid dynamics simulations. It performs better than traditional computational fluid dynamics methods, which are merely based on the finite element method application to the Navier-Stokes equation. LBM is advantageous in porous media flows simulation, multiphase flows simulation; it can be parallelized easier [1-3].

Due to the microscopic nature of the lattice Boltzmann method there is a non-trivial task of implementing boundary conditions to make them correspond to the macroscopic constraints. The most wide-spread boundaries are solid no-slip ones, constant velocity or constant density boundaries [4]. Note that it is always sufficient to specify either velocity or density (not both of them) due to the Navier-Stokes equation properties. A ubiquitous approach for no-slip boundaries implementation is the usage of bounce-back nodes [1]. Also, several specific methods exist for constant density and constant velocity boundaries modeling on straight walls [4]. An accurate method for constant velocity boundaries simulation on curved walls is essential for numerous applications, for example, packed-bed reactors modeling. Unfortunately, all of the existing methods for curved boundaries simulations use interpolation schemes and are not local (which is the most attractive feature of the LBM) or expose poor properties [5-9]

We propose another method for modeling constant velocity constraints on curved walls, which doesn't depend on dimensionality, LBM scheme (D2Q9 or D3Q19, BGK or multiple relaxation times, etc.), boundary geometry; it is numerically stable, accurate and local; it has a good physical background. Also, we provide theoretical justifications of a popular Zou-He boundaries technique and an isothermal boundaries algorithm[10]. These justifications are based on the maximum entropy method derived. Finally, while conducting numerical benchmarks, typical straight boundaries algorithm (Zou-He) was compared to a typical curved boundaries algorithm (Guo-Zheng) [5],[10]. Such a comparison is unknown to the authors.

2. Lattice Boltzmann Method Overview

Continuous Boltzmann equation describes diluted gases on microscopic level, it determines the dynamics of particle velocities distribution in space and time. This equation is known to correspond macroscopically to the Navier-Stokes equation after averaging and obtaining macroscopic quantities, such as density and velocity.

The following approach is applied to bring the Boltzmann equation to fruition for dense liquids: despite the fact that dense liquids are not described with the Boltzmann equation, if they are modeled according to the latter one, the Navier-Stokes equation is still recovered. Therefore, the lower abstractions level (i.e. microscopic equation for dense liquids) is substituted with physically irrelevant implementation (i.e. microscopic equation for dilute gases), so that the upper abstractions layer (the Navier-Stokes equation) still holds.

The Boltzmann equation should be discretized in time, space and velocity directions. Time is discretized uniformly, space is discretized in a uniform way, too, to obtain a rectangular grid; velocity directions from a given node should point to neighboring lattice nodes.

Finally, the master equation for the LBM scheme with the BGK approximation for the collision term looks like

$$f_i(\boldsymbol{x}+\boldsymbol{c}_i\Delta t, t+\Delta t)=f_i(\boldsymbol{x},t)-\frac{1}{\tau}[f_i(\boldsymbol{x},t)-f_i^{eq}(\boldsymbol{x},t)] \quad . \qquad (1)$$

In this equation $f_i(\boldsymbol{x},t)$ is a particle population in direction i, i.e. the probability to find a particle moving in a direction i from a node $\boldsymbol{x}$ at time t. These probabilities are normalized by mass of the fluid. τ is the mass relaxation time, which is related to the kinematic viscosity as $\nu = c_s^2(\tau - 1/2)$, where $c_s = 1/\sqrt{(3)}$ is the sound speed [11].

Equilibrium particle population in direction i is

$$f_i^{eq} = w_i \rho [1 + 3\boldsymbol{\upsilon}\cdot\boldsymbol{c_i} + \frac{9}{2}(\boldsymbol{\upsilon}\cdot\boldsymbol{c_i})^2 - \frac{3}{2}\upsilon^2] \quad , \tag{2}$$

where macroscopic density in the given node is ρ , macroscopic velocity is $\boldsymbol{\upsilon}$. Lattice vectors for the current scheme are $\boldsymbol{c_i}$, while w_i are lattice vectors weights.

Density and velocity are obtained by the following formulas:

$$\rho = \sum_i f_i, \quad \rho\boldsymbol{\upsilon} = \sum_i f_i \boldsymbol{c}_i \tag{3}$$

Note that lattice vectors and their weights are constructed to satisfy symmetry conditions [1] and to recover density and velocity for continuous Boltzmann dynamics [12].

Typical lattice stencil for a two-dimensional space is the D2Q9 one, which contains of 9 lattice vectors with the following coordinates [2]

$$\begin{pmatrix} 0 & +1 & 0 & -1 & 0 & +1 & -1 & -1 & +1 \\ 0 & 0 & +1 & 0 & -1 & +1 & +1 & -1 & -1 \end{pmatrix} \tag{4}$$

Lattice weights for this scheme are the following: $w_i = 4/9$ for a vector with $|\boldsymbol{c}_i|^2 = 0$, $w_i = 1/9$ for vectors $|\boldsymbol{c}_i|^2 = 1$, $w_i = 1/36$ for vectors $|\boldsymbol{c}_i|^2 = 2$.

Three-dimensional domain is usually handled with the help of the D3Q19 lattice, which possesses the following 19 vectors

$$\begin{pmatrix} 0 & +1 & 0 & 0 & -1 & 0 & 0 & +1 & -1 & -1 & +1 & +1 & 0 & -1 & 0 & +1 & 0 & -1 & 0 \\ 0 & 0 & +1 & 0 & 0 & -1 & 0 & +1 & +1 & -1 & -1 & 0 & +1 & 0 & -1 & 0 & +1 & 0 & -1 \\ 0 & 0 & 0 & +1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \end{pmatrix} \quad . \tag{5}$$

These vectors are supplied with the following weights: $w_i = 1/3$ for a vector with $|\boldsymbol{c}_i|^2 = 0$, $w_i = 1/18$ for vectors $|\boldsymbol{c}_i|^2 = 1$, $w_i = 1/36$ for vectors $|\boldsymbol{c}_i|^2 = 2$.

An important point in the LBM framework is specific units conversion: all the physical values, mentioned above, are measured in lattice units [13].

LBM method analysis and linking it to the Navier-Stokes equation are usually performed through one of the following techniques: 1. Taylor expansion [14] 2. asymptotic analysis [15] 3. Chapman-Enskog expansion [1].

3. Maximum Entropy Boundaries

In this section maximum entropy boundaries method will be explained.

3.1. Method Idea

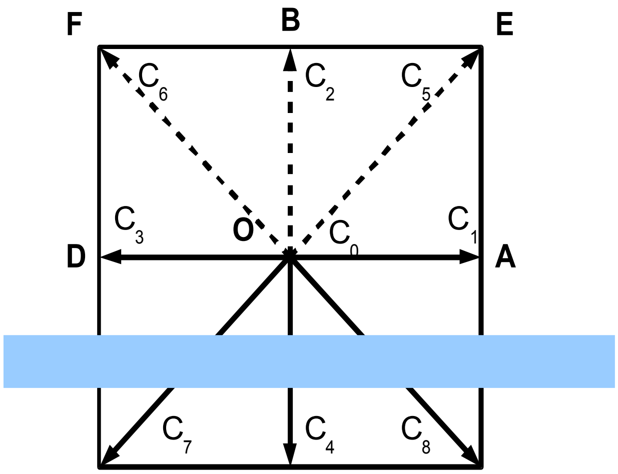

Fig 1 Boundary node example

Assume that the node O is a boundary one. After collision and streaming step dashed particle populations are unknown, as they originate from empty space, not from the fluid. It is necessary to fill them before the next collision and streaming steps to provide particle populations for the nodes E, B, F. The main idea of the method is to fill the missing populations as close as possible to the equilibrium values, but to preserve known populations and to satisfy the predefined velocity constraint. Note that collision step should be executed as usual for such nodes. Also note that equilibrium distributions depend on local density, which is unknown before unknown populations are recovered, but this difficulty will be easily handled. Choosing the unknown particle populations to be closest to the equilibrium distribution is physically relevant, as we are aiming at modeling boundary nodes in the most equilibrium state; also, this implies maximum entropy. For this reason we'll call such boundary nodes as maximum entropy nodes.

3.2. *Main Algorithm*

In this section the basic version of the algorithm will be presented.

3.2.1. Mathematical Formulation

Let's introduce several additional notations. A set of indexes for known particle populations is K (in Fig 1 $K=\{0,1,3,4,7,8\}$). A set of indexes for unknown particle populations is U (in Fig 1 $U=\{2,5,6\}$). Number of dimensions is d . Number of lattice vectors (from the DnQm notation) is q . Number of known populations is k . Number of unknown populations is $n=q-k$. Known density is

$$\Delta\rho=\sum_{i\in K} f_i \tag{6}$$

Known momentum is

$$\Delta \boldsymbol{M}=\sum_{i\in K} f_i \boldsymbol{c_i} \tag{7}$$

As far as we will deal primarily with unknown distributions since this point, it is a good idea to rearrange indexes and lattice vectors, so that unknown lattice vectors had indexes from 1 to $n=q-k$.

Also, let's introduce a special notation for equilibrium distribution functions for unit density nodes:

$$\varphi_i=w_i[1+3\boldsymbol{\upsilon}\cdot\boldsymbol{c_i}+\frac{9}{2}(\boldsymbol{\upsilon}\cdot\boldsymbol{c_i})^2-\frac{3}{2}\upsilon^2] \tag{8}$$

These values are known for each boundary node, as they include just predefined velocity.

Finally, we are able to formulate the optimization problem to reach the maximum entropy with velocity constraint. We will use a usual quadratic (least-squares) minimization approach; i.e. minimize L_2 vector norm of a difference between unknown particle populations and equilibrium particle populations (subject to a constraint of predefined velocity). The objective function will look like

$$F=\sum_{j\in U}(f_j-\rho\varphi_j)^2=\sum_{j=1}^{n}(f_j-\rho\varphi_j)^2 \tag{9}$$

Note that

$$\rho=\sum_{j} f_j=\sum_{j\in U} f_j+\sum_{j\in K} f_j=\sum_{j=1}^{n} f_j+\Delta\rho \tag{10}$$

Velocity restrictions look like

$$\rho\boldsymbol{\upsilon}=\sum_{j} f_j\boldsymbol{c_j}=\sum_{j\in U} f_j\boldsymbol{c_j}+\sum_{j\in K} f_j\boldsymbol{c_j}=\sum_{j=1}^{n} f_j\boldsymbol{c_j}+\Delta\boldsymbol{M} \quad \text{or} \tag{11}$$

$$\sum_{j=1}^{n} f_j(\boldsymbol{c_j})_i+\Delta M_i-(\sum_{j=1}^{n} f_j+\Delta\rho)\upsilon_i=0 \tag{12}$$

Finally, the Lagrangian of the optimization problem, subject to velocity constraint, will look like

$$\begin{gathered} L=\sum_{j=1}^{n}(f_j-(\sum_{j=1}^{n} f_j+\Delta\rho)\varphi_j)^2+ \\ \sum_{j=1}^{d}\lambda_j(\sum_{p=1}^{n} f_p(\boldsymbol{c_p})_j+\Delta M_j-(\sum_{j\in 1}^{n} f_j+\Delta\rho)\upsilon_j) \end{gathered} \tag{13}$$

where λ_j are Lagrange multipliers. To solve the optimization problem one needs to find partial derivatives of (13) by all the f_i and λ_i and equal them to zero. This leads to the linear system of equations of dimensionality $n+d$ (see Appendix):

$$\Gamma\boldsymbol{x}=\boldsymbol{b} \tag{14}$$

Where

$$\boldsymbol{b}=\begin{pmatrix}\Delta\rho\boldsymbol{\alpha}\\ \Delta\rho\boldsymbol{\upsilon}-\Delta\boldsymbol{M}\end{pmatrix} \tag{15}$$

$$\alpha_i=2(\varphi_i-\sum_{j=1}^{n}\varphi_j^2), i=\overline{1,n} \tag{16}$$

$$\boldsymbol{x}=(f_1,\ldots,f_n,\lambda_1,\ldots,\lambda_d)^T \tag{17}$$

$$\Gamma=\begin{pmatrix}\Gamma^{11} & \Gamma^{12}\\ \Gamma^{21} & \Gamma^{22}\end{pmatrix} \tag{18}$$

$$\begin{gathered}\Gamma_{ij}^{11}=2\delta_{ij}-2\varphi_j-\alpha_i,\ \ i,j=\overline{1,n},\\ \Gamma_{ij}^{12}=(\boldsymbol{c}_i)_j-\boldsymbol{\upsilon}_j,\ \ i=\overline{1,n}, j=\overline{1,d},\\ \Gamma_{ij}^{21}=(\boldsymbol{c}_j)_i-\boldsymbol{\upsilon}_i,\ \ i=\overline{1,d}, j=\overline{1,n},\\ \Gamma_{ij}^{22}=0,\ \ i,j=\overline{1,d}\end{gathered} \tag{19}$$

3.2.2. The Final Algorithm

On each time step for each boundary node the program has to do the following:

1. find known and unknown population indexes K and U
2. find known density and velocity by (6) and (7)
3. Fill the system of linear equations (14) by (15)-(19) and solve it
4. Distribute found values by unknown populations
5. Perform collision (even for maximum entropy boundary nodes) and streaming steps as usual

3.2.3. Solvability Conditions

Equation (14) is solvable if and only if $det\,\Gamma\neq 0\Leftrightarrow det\,\Gamma^{21}\Gamma^{12}\neq 0$, which is equivalent to linear independence of Γ^{12} columns. Also note that the least-squares Lagrangian can always be minimized if velocity restrictions (12) are solvable for particle populations. It leads to exactly the same condition: Γ^{12} columns should be linearly independent. System (14) is almost always solvable in the incompressible limit for sufficiently smooth boundaries, except for rare nodes with too few unknown populations. Nodes on straight or convex boundaries are always solvable.

3.3. Averaged Density Maximum Entropy Algorithm

A slightly modified version of the maximum entropy algorithm is presented here. It uses the idea of density averaging, i.e. during the recovering of boundary node parameters density is obtained through extrapolation of densities on the previous step from neighboring nodes, then unknown particle populations are discovered through entropy maximization.

3.3.1. Mathematical Formulation

Lagrangian for the entropy maximization procedure will look as following, similar to (13) (note that density is known after extrapolation, velocity is predefined, so equilibrium populations are known):

$$L=\sum_{j=1}^{n}(f_j-f_j^{eq})^2+\sum_{j=1}^{d}\lambda_j(\sum_{p=1}^{n}f_p(\vec{c_p})_j+\Delta M_j-\rho\upsilon_j) \tag{20}$$

After computing derivatives and equaling them to zero one will obtain the system of linear equations, similar to (14), but with a different matrix and a different right side:

$$\boldsymbol{b}=(2f_1^{eq},\ldots,2f_n^{eq},\rho\upsilon_{1,}\ldots,\rho\upsilon_d)^T \tag{21}$$

$$\Gamma=\begin{pmatrix}\Gamma^{11} & \Gamma^{12}\\ \Gamma^{21} & \Gamma^{22}\end{pmatrix} \tag{22}$$

$$\begin{gathered}\Gamma_{ij}^{11}=2\delta_{ij},\ \ i,j=\overline{1,n},\\ \Gamma_{ij}^{12}=(\boldsymbol{c_i})_j,\ \ i=\overline{1,n},j=\overline{1,d},\\ \Gamma_{ij}^{21}=(\boldsymbol{c_j})_i,\ \ i=\overline{1,d},j=\overline{1,n},\\ \Gamma_{ij}^{22}=0,\ \ i,j=\overline{1,d}\end{gathered} \tag{23}$$

3.3.2. Solvability Conditions

Solvability conditions can be derived in the same way as for general maximum entropy algorithm: Γ^{12} columns should be linearly independent; which holds for plane and convex boundaries.

3.4. *Maximum Entropy Method for Isothermal Boundaries*

Entropy maximization idea is applied for isothermal boundaries below.

In this section we will use f_i to denote heat populations and assume that they have already been rearranged as in the previous section, so that unknown populations come first. Temperature from the known populations will be denoted as ΔT .

Lagrangian for the optimization problem looks like

$$L=\sum_{j=1}^{n}(f_j-f_j^{eq})^2+\lambda(\sum_{j=1}^{n}f_j+\Delta T-T) \tag{24}$$

Once again, one has to compute derivatives and equate them to zero. It leads to the usual linear system of equations, but system matrix will look like

$$\Gamma=\begin{pmatrix}2 & 0 & 0 & \dots & 1\\ 0 & 2 & 0 & \dots & 1\\ 0 & 0 & 2 & \dots & 1\\ & & \dots & & \\ 1 & 1 & 1 & \dots & 0\end{pmatrix} \tag{25}$$

Right-side vector has the following form:

$$\boldsymbol{b}=(2f_1^{eq},\dots,2f_n^{eq},T-\Delta T)^T \tag{26}$$

Surprisingly, this system has always an exact analytical solution (see Appendix):

$$f_i=f_i^{eq}-\frac{1}{n}(\sum_{i\in K}f_i-\sum_{i\in K}f_i^{eq}) \quad , \quad i=\overline{1,n} \tag{27}$$

It has a very simple meaning: we assign equilibrium distributions to the unknown lattice vectors, then compute temperature excess for the known populations and then uniformly distribute this excess by unknown populations.

3.5. Zou-He Boundaries Justification

Zou and He intuitive procedure for 3D case is justified in this section.

Zou and He proposed a certain method for modeling straight density and velocity boundaries [10],[4]. It works as following: at first one has to determine either an unknown density or an unknown velocity for the given boundary node. Then unknown distributions are populated with equilibrium values. Afterward the bounce-back of non-equilibrium distribution parts is performed to obtain unknown lattice vectors. Normal velocity component is recovered on this stage . To recover tangent velocity component the authors propose to compute momentum excess for populations after bounce-back and then to distribute it in a uniform way over unknown populations.

We are going to show that the uniform excess distribution procedure corresponds to the minimization of the difference between final particle distributions and bounced-back distributions with the constraint of compensating the momentum excess.

Let's denote bounced-back distributions as f_i^{bb} :

$$f_i^{bb}=f_i,\ \ i\in K\ ;\ \ f_i^{bb}=f_i^{eq}+f_{i'}-f_{i'}^{eq},\ \ i\in U \tag{28}$$

Here i' means a vector, opposite to the vector i . This vector is always known for nodes on plane boundaries.

Momentum excess for bounced-back populations over equilibrium distributions is

$$\boldsymbol{E}=\boldsymbol{M}^{\boldsymbol{bb}}-\boldsymbol{M}^{\boldsymbol{eq}}=\sum_{j=1}^{q} f_j^{bb}\boldsymbol{c}_j-\sum_{j=1}^{q} f_j^{eq}\boldsymbol{c}_j \tag{29}$$

In [4] this variable is called $\boldsymbol{\delta}$. It can be shown (see Appendix) that this vector is parallel to the wall (so let's say it vanishes in the last dimension, $E_d = 0$).

Finally, Lagrangian of the problem looks like (note the constraints term)

$$L = \sum_{j=1}^{n} (f_j - f_j^{bb})^2 + \sum_{j=1}^{d} \lambda_j \left(\sum_{p=1}^{n} f_p (\boldsymbol{c_p})_j - \sum_{p=1}^{n} f_p^{bb} (\boldsymbol{c_p})_j + E_j \right) \qquad (30)$$

Zou and He proposed to use the following expression for the final particle populations:

$$f_i = f_i^{bb} - \boldsymbol{c_i} \cdot \boldsymbol{e} \quad , \qquad (31)$$

where

$$e_i = E_i / n_i , \quad i = \overline{1, d} \quad , \qquad (32)$$

where n_i is the number of lattice vectors with non-zero coordinates along the axis i :

$$n_i = \sum_{p=1}^{n} |(\boldsymbol{c_p})_i| = \sum_{p=1}^{n} (\boldsymbol{c_p})_i^2 \quad . \qquad (33)$$

After inserting equations (31) into derivatives of (30) it is possible to show that they are equal to zero, while the Lagrangian multipliers can be determined unambiguously (see Appendix). So (31) really minimizes (30).

Taking into consideration results from the section 3.2.3Solvability Conditions, the solution (31) for the optimization problem on plane boundaries is unique.

4. Numerical Results

Numerical results for different systems and for different node types are presented in this section. All physical parameters below are provided in lattice units. All systems were tested with Guo-Zheng nodes, maximum entropy nodes, Zou-He nodes (which support just plane boundary conditions), Chen-Martinez boundary nodes (which support curved boundary conditions, but need specific treatment for curved boundaries, so they were used just for plane walls) [5],[8],[10].

4.1. Poiseuille Flow

We consider an academic two-dimensional Poiseuille flow within the following setting. Flow is force-driven to prevent pressure and density changes and to make the simulation more accurate. Solid walls are implemented through bounce-back nodes; but inlet and outlet nodes are implemented through constant velocity nodes with stationary parabolic profile. As far as flow is force-driven, it is not necessary to set the outlet boundary to the constant density (and pressure) type to ensure proper pressure gradient.

System dimensionality was set to 2; D2Q9 lattice Boltzman scheme was utilized; system size by flow direction (X axis) is twice as big as system size by Y direction; kinematic viscosity in lattice units is 0.1; Reynolds number is 5.

Converged flow solutions for different velocity boundaries and for different system widths (Y axis) were obtained in the first set of experiments and errors between these solutions and expected solutions were measured. Solutions error was computed according to the Maier L_2 norm[16] in the space of possible system states by the following formula:

$$\epsilon = \frac{\sqrt{\sum_{nodes} |\boldsymbol{u}^{theor} - \boldsymbol{u}^{\exp}|^2}}{\sqrt{\sum_{nodes} |\boldsymbol{u}^{theor}|^2}} \tag{34}$$

Numerical simulations were stopped when the difference by the given norm between two solutions separated with 20 iterations was less than 10^{-5} .

The results are presented in the following figure.

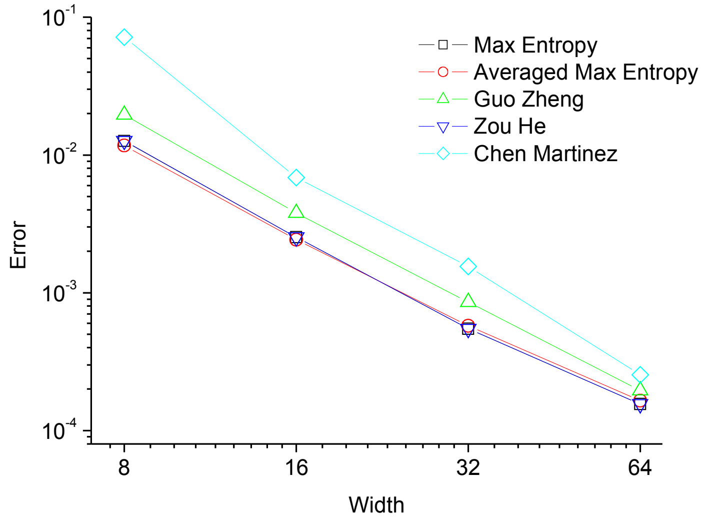

Fig 2 Poiseuille flow errors

Both axes are in logarithmic scales. The slope of these lines is close to 2, which corresponds to the fact that Guo-Zheng and Zou-He nodes are of the second order of convergence[4],[8] and suggests that maximum entropy nodes expose the same convergence behavior.

Also, maximum Reynolds numbers are measured by varying flow viscosity for several boundary node types and for several system widths. Flow is force-driven, system is two-dimensional; D2Q9 scheme is used. Solid walls are implemented through the bounce-back scheme; inlet and outlet are implemented through constant velocity nodes with stationary profile. System size by flow axis is twice as big as system width, system width varies, and maximum flow velocity is 0.01. These results are presented in the following figure.

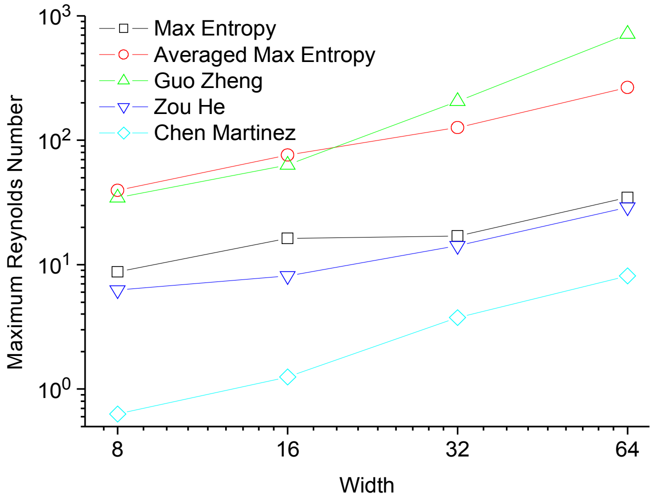

Fig 3 Maximum Reynolds numbers for force-driven Poiseuille flows

4.2. *Cylindrical Couette Flow*

The next test was conducted for cylindrical Couette flow, which is a flow between two coaxial rotating cylinders. There is a well-known stationary solution for this flow[17]:

$$\upsilon = ar + \frac{b}{r} \tag{35}$$

where

$$a = \frac{\upsilon_2 R_2}{R_2^2 - R_1^2}, \quad b = \frac{-\upsilon_2 R_1^2 R_2}{R_2^2 - R_1^2} \tag{36}$$

Here υ_2 is the velocity of the outer cylinder (if the inner one is not rotating), $R_{1,}$ R_2 are inner and outer cylinders radii respectively.

A three-dimensional case was considered and a D3Q19 scheme was utilized. System is periodic by z, its length along z is 2, and inner radius is two times less than the outer one, outer radius is varied. Boundary nodes have equal node type. Viscosity is 0.1; Reynolds number is 5.

The first set of experiments measured errors between the converged solutions and theoretical solutions for different boundary node types and system sizes. Error was measured by (34), convergence was accepted if the error between two solutions, separated with 20 iterations, was less than 10^{-5}. Chen Martinez nodes were not used for tests, as it is unclear how to make them support curved boundaries.

The results are depicted in the following figure.

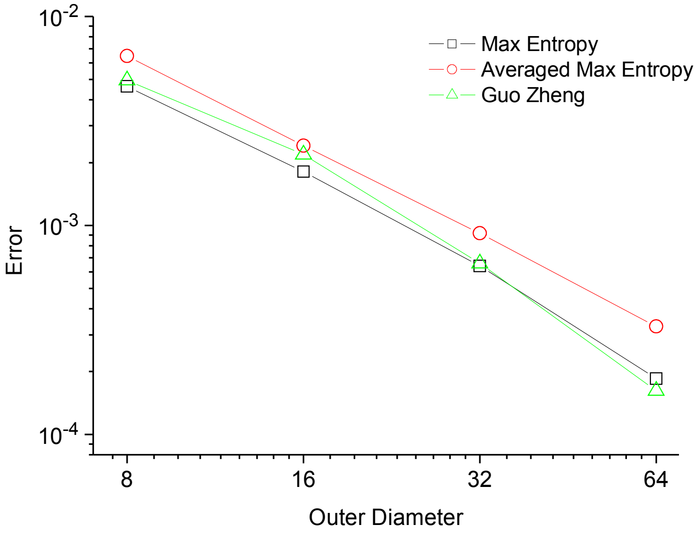

Fig 4 Cylindrical Couette flow errors

Note that in this setting the matrix of the optimization equation was singular for some nodes, so we used Guo-Zheng method for them. There were 176 boundary nodes, and 48 singular nodes among them for outer diameter 16. These proportions preserve for higher resolutions.

Also, maximum supported Reynolds numbers were measured by varying flow viscosity in a manner similar to the previous section. Measurement parameters are the following: D3Q19 scheme is used; system is periodic by z, inner radius is two times less than the outer one, outer radius is varied; boundaries have equal node type; outer velocity is 0.01. The results are depicted below.

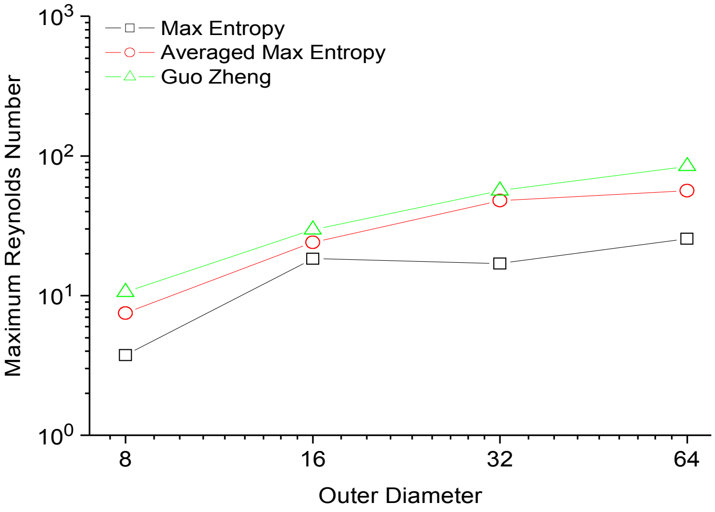

Fig 5 Maximum Reynolds numbers for Couette flows

4.3. Duct Flow

We also tested a three-dimensional square duct flow[10],[4]. System setting is similar to the Poiseuille flow setting, except dimensionality, i.e. D3Q19 scheme is brought to fruition, duct has a square profile, duct length is equal to the duct square side, duct square side is varied, solid walls are implemented through bounce-back algorithm, and inlet and outlet boundaries are of constant velocity type with a stationary velocity distribution. Viscosity is 0.1, Reynolds number is 5; flow is force-driven. Convergence threshold is preserved. Converged solutions are checked against analytical ones.

The following results were obtained:

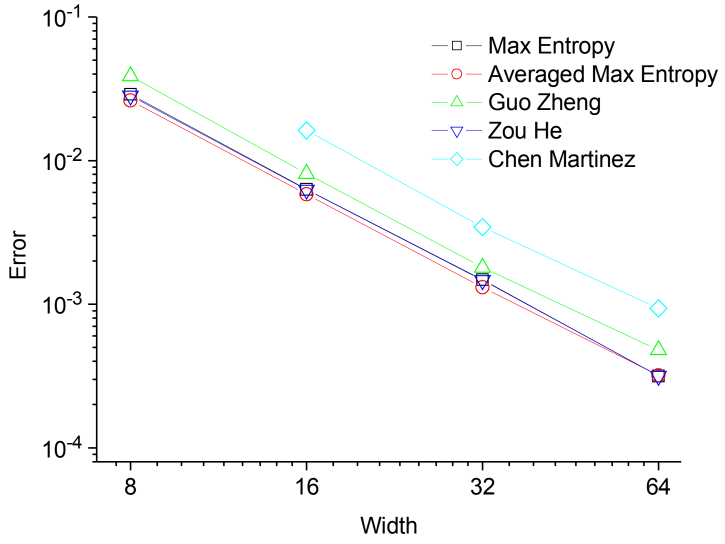

Fig 6 Duct flow errors

On the next step maximum available Reynolds numbers for the rectangular duct flow were found. Flow parameters were left the same, but viscosity was varied, maximum velocity was 0.01.

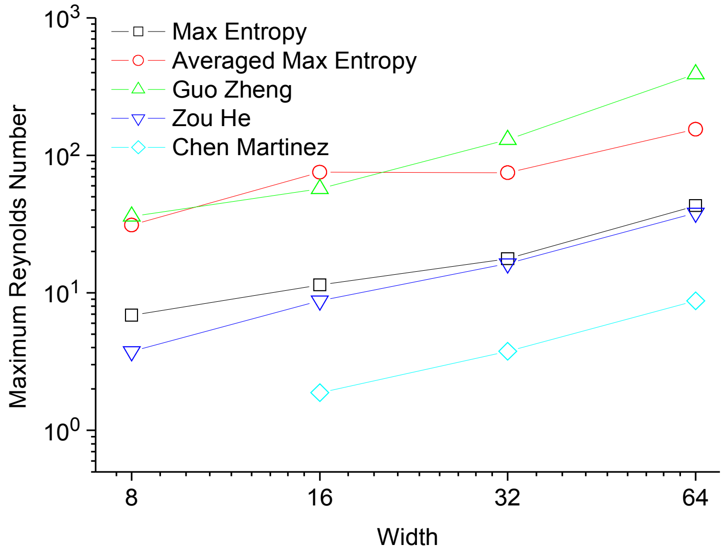

Fig 7 Maximum Reynolds numbers for square duct flows

4.4. Cylindrical Semi Permeable Membranes Flow

This test is conducted for a stationary flow in a cylindrical tube with semi permeable membranes. There are numerous approximate, as well as exact analytical solutions for Navier-Stokes equations for the case of constant suction or injection rate and for the case when walls are operating according to the Darcy's law[18-22]. Fluid velocity through membranes is assumed to be constant and predefined for simplicity. Despite there is an exact analytical solution for such a case[21], it turns out that an approximate solution is simpler and easier for analysis and implementation.

The following variables are introduced: axial velocity is υ, radial velocity is u, pressure is p, membrane velocity, directed outside, is V, cylinder radius is a, non-dimensional radial coordinate $\eta = r^2/a^2$.

A general solution for such a problem has the form[20]

$$\begin{gathered} \upsilon = V\, f(\eta)/\sqrt{(\eta)} \\ u = u_0\, f'(\eta)(1 - 2\, V\, z/a) \\ p = p_0 + \rho(A z^2 + Bz) + \rho V^2 (2 f'/R - f^2/\eta) \end{gathered} \tag{37}$$

where u_0 is a characteristic velocity, $R = V\,a/\nu$ is suction Reynolds number (don't mix it with the main Reynolds number, $R_{main} = u_0\, a/\nu$), $f(\eta)$ is the solution for the following equation with boundary conditions

$$\begin{gathered} \eta\, f''' + f'' + R/2\,(f'^2 - f\, f') = K\,, \\ f'(1) = 0,\ \ f(0) = 0,\ \ \lim_{\eta \to 0} \sqrt{(\eta)}\, f''(\eta) = 0,\ \ f(1) = 1\ \ . \end{gathered} \tag{38}$$

$$A = -4\, V^2\, K/(R\, a^2)\,, B = 4\, V\, K\, u_0/(R\, a) \tag{39}$$

Constant K should be determined during the solution of the boundary-value problem (38) due to sufficient boundary conditions.

The approximate solution to this problem, obtained with the perturbation theory, can be found in[22],[23]

$$f(\eta)=2\eta-\eta^2+\frac{R}{36}(4\eta-9\eta^2+6\eta^3-\eta^4),$$
$$K=-2+\frac{3}{2}R \tag{40}$$

This solution is applicable, when wall Reynolds number R is in the range $[-2;2]$.

A D3Q19 scheme was utilized in the simulation. Cylinder length is equal to the cylinder diameter, which is varied, cylindrical wall nodes type is varied, inlet and outlet boundaries are built with constant velocity Guo-Zheng nodes. Viscosity is 0.1, main Reynolds number is 5, suction Reynolds number is 1, external force is absent. Convergence threshold from the previous section is used. Converged solutions are checked against analytical ones. Unfortunately, basic maximum entropy algorithm turned out to be divergent for this benchmark, so just averaged maximum entropy method was left.

The results are depicted in the following figure.

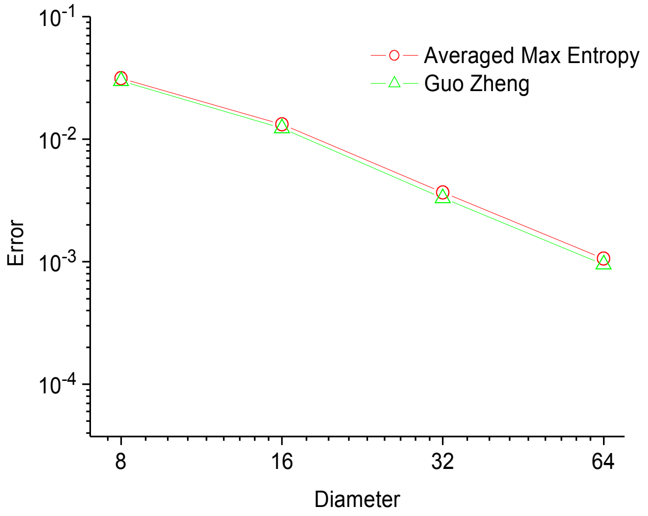

Fig 8 Cylindrical flow errors

As usual, maximum supported main Reynolds numbers were measured. System setting is exactly the same, except for preserving maximum flow velocity on the level 0.01. Consequently, for any given main Reynolds number flow viscosity is computed, then suction rate is defined to preserve suction Reynolds number. The results are depicted below.

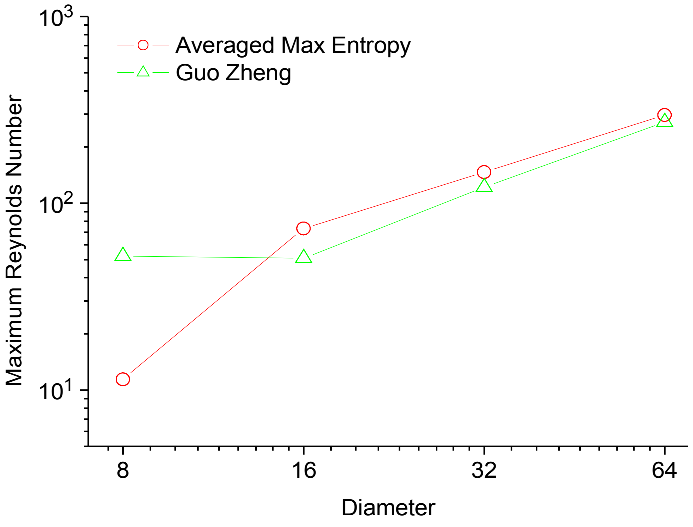

Fig 9 Maximum Reynolds numbers for cylindrical flows

4.5. Unsteady Plane Couette Flow

The last test is aimed at determining boundaries quality while dealing with unsteady flow. All the previous examples lack this feature.

Plane Couette flow is equivalent to the cylindrical one, if cylinders radii tend to infinity; i.e. the bottom plate has zero velocity, the top plate has some specified velocity. If the initial fluid flow is zero, the time-dependent solution is given in [8].

Other system parameters are as follows: system is two-dimensional, and is modeled with D2Q19 scheme, system is periodic by X, number of nodes by flow direction (i.e. by

X axis direction) are 10, number of nodes by Y axis is 50, viscosity is 0.1, and Reynolds number is 5.

Errors are reported in the figure below.

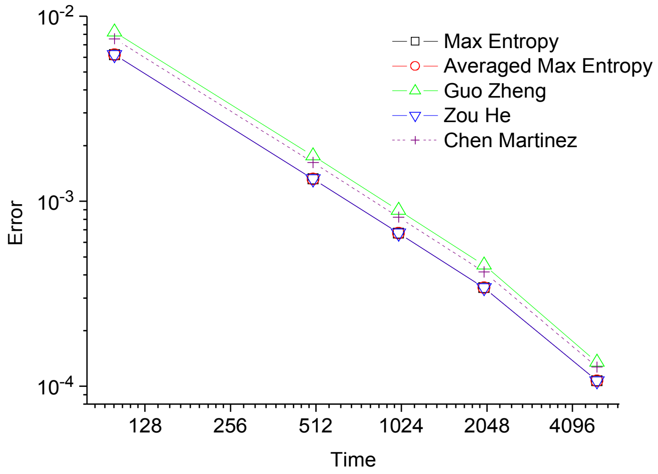

Fig 10 Unsteady plane Couette flow errors

Note that maximum entropy, averaged maximum entropy and Zou-He graphs are very close to each other.

5. Conclusion

Obviously, for straight boundaries Zou-He, maximum entropy and averaged maximum entropy algorithms produce very similar errors; Guo-Zheng algorithm is slightly worse. Nevertheless, Guo-Zheng algorithm is superior in stability, followed by averaged maximum entropy method.

Guo-Zheng produces slightly bigger errors for cylindrical Couette flow than maximum entropy algorithm; but simulation errors are hardly distinguishable for flow in a porous tube. Also, Guo-Zheng algorithm is more stable than other algorithms for cylindrical Couette flow; but it is outperformed by averaged maximum entropy method for porous tube flow.

The reason for a good accuracy of the maximum entropy method, especially for straight boundaries, is its underlying physical nature and the absence of extrapolations.

The reason for Guo-Zheng and averaged maximum entropy stability advantage is their non-local nature; as several neighboring fluid nodes prevent the instability growth. Also, maximum entropy algorithms (main version and with averaging) are inapplicable for some nodes in curved boundaries, and they are replaced with Guo-Zheng nodes. This replacement leads to numerical instabilities.

It is known, that local numerical schemes reveal better accuracy than non-local ones for flows with high spatial harmonics[4]; so maximum entropy method may be more suitable in such tasks than extrapolation scheme, like Guo-Zheng, Filippova-Hanel, etc.

One more advantage of the maximum entropy algorithms is their independence over collision scheme, while, for example, Guo-Zheng method in its current state is tightly bound to the BGK scheme.

Also, during our benchmarks we compared straight boundaries algorithm to curved boundaries algorithms; and it turned out that the latter ones are comparable in accuracy but are superior in stability. So if you seek generality better use curved boundaries algorithms at once.

Finally, entropy maximization idea provides a convenient framework for analyzing lattice Boltzmann boundary schemes, whose capabilities were applied for the Zou-He boundaries method.

6. Appendix

6.1. Lagrangian Derivatives Computation

In this part we will compute derivatives for the Lagrangian (13) and come to the linear equations (14).

At first, let's compute derivatives by particle distributions ($i=\overline{1,n}$)

$$\begin{aligned}
\frac{\partial L}{\partial f_i} &= \sum_{j=1}^{n} 2(f_j-[\sum_{p=1}^{n} f_p+\Delta\rho]\varphi_j)(\delta_{ij}-\sum_{p=1}^{n}\delta_{pi}\varphi_j)+\\
&\quad+\sum_{j=1}^{d}\lambda_j[\sum_{p=1}^{n}(\boldsymbol{c}_p)_j\delta_{pi}-\sum_{p=1}^{n}\delta_{pi}\upsilon_j]=\\
&=\sum_{j=1}^{n} 2(f_j-[\sum_{p=1}^{n} f_p+\Delta\rho]\varphi_j)(\delta_{ij}-\varphi_j)+\sum_{j=1}^{d}\lambda_j[\sum_{p=1}^{n}(\boldsymbol{c}_i)_j-\upsilon_j]=\\
&=\sum_{j=1}^{n} 2(f_j-[\sum_{p=1}^{n} f_p]\varphi_j-\Delta\rho\varphi_j)(\delta_{ij}-\varphi_j)+\sum_{j=1}^{d}\lambda_j[\sum_{p=1}^{n}(\boldsymbol{c}_i)_j-\upsilon_j]=\\
&=\sum_{j=1}^{n} 2(\delta_{ij}-\varphi_j)f_j-\sum_{p=1}^{n} f_p\sum_{j=1}^{n}2(\delta_{ij}-\varphi_j)\varphi_j-\\
&\quad-\Delta\rho\sum_{j=1}^{n}2(\delta_{ij}-\varphi_j)\varphi_j+\sum_{j=1}^{d}\lambda_j[\sum_{p=1}^{n}(\boldsymbol{c}_i)_j-\upsilon_j]=\\
&=\sum_{j=1}^{n} 2(\delta_{ij}-\varphi_j)f_j-\sum_{p=1}^{n} f_p\alpha_i-\Delta\rho\alpha_i+\sum_{j=1}^{d}\lambda_j[\sum_{p=1}^{n}(\boldsymbol{c}_i)_j-\upsilon_j].
\end{aligned}$$

Finally,

$$\frac{\partial L}{\partial f_i}=\sum_{j=1}^{n}(2\delta_{ij}-2\varphi_j-\alpha_i)f_j-\Delta\rho\alpha_i+\sum_{j=1}^{d}\lambda_j[\sum_{p=1}^{n}(\boldsymbol{c}_i)_j-\upsilon_j]\quad,\quad i=\overline{1,n} \tag{41}$$

Where $\alpha_i=\sum_{j=1}^{n}2(\delta_{ij}-\varphi_j)\varphi_j=2(\varphi_i-\sum_{j=1}^{n}\varphi_j^2)$, as in formula (16).

Now it's high time to compute derivatives by Lagrangian multipliers ($i=\overline{1,d}$):

$$\begin{aligned}
\frac{\partial L}{\partial \lambda_i} &= \sum_{j=1}^{d}\delta_{ij}(\sum_{p=1}^{n}(\boldsymbol{c}_{\boldsymbol{p}})_j f_p+\Delta M_j-[\sum_{p=1}^{n} f_p+\Delta\rho]\upsilon_j)=\\
&=\sum_{p=1}^{n}(\boldsymbol{c}_{\boldsymbol{p}})_i f_p+\Delta M_i-[\sum_{p=1}^{n} f_p+\Delta\rho]\upsilon_i.
\end{aligned}$$

So,

$$\frac{\partial L}{\partial \lambda_i} = \sum_{j=1}^{n} [(\boldsymbol{c}_j)_i - \upsilon_i] f_j - (\Delta \rho \upsilon_i - \Delta M_i) \quad , \quad i = \overline{1, d} \quad . \tag{42}$$

Finally, after equating (41), (42) to zeros one obtains (15) and (19).

6.2. Isothermal Boundaries Solution

In this part we will show that (27) is really a solution to the system of linear equations with matrix (25) and right-side (26). First of all, matrix (25) is nicely invertible, and its inverse has the form:

$$\Gamma^{-1} = \begin{pmatrix} (n-1)/2\mathrm{n} & -1/2\mathrm{n} & -1/2\mathrm{n} & \ldots & -1/2\mathrm{n} & 1/n \\ -1/2\mathrm{n} & (n-1)/2\mathrm{n} & -1/2\mathrm{n} & \ldots & -1/2\mathrm{n} & 1/n \\ & & & \ldots & & \\ -1/2\mathrm{n} & -1/2\mathrm{n} & -1/2\mathrm{n} & \ldots & (n-1)/2\mathrm{n} & 1/n \\ 1/n & 1/n & 1/n & \ldots & 1/n & -2/n \end{pmatrix} \tag{43}$$

Note that n here is not the system dimensionality, but the number of unknown lattice vectors, while the dimensionality is $n+1$.

The validity of the inversion can be checked manually.

Consequently, unknown populations are obtained in the following manner:

$$\begin{aligned} f_i &= \sum_{j=1}^{n+1} \Gamma^{-1}{}_{ij} b_j = \Gamma^{-1}{}_{ii} b_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} \Gamma^{-1}{}_{ij} b_j + \Gamma^{-1}{}_{in} b_n = \\ &= \frac{n-1}{2n} 2 f_i^{eq} - \frac{1}{2n} \sum_{\substack{j=1 \\ j \neq i}}^{n} 2 f_j^{eq} + \frac{1}{n} (T - \Delta T) = \\ &= f_i^{eq} - \frac{1}{n} f_i^{eq} - \frac{1}{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} f_j^{eq} + \frac{1}{n} \Big(\sum_j f_j^{eq} - \sum_{j \in K} f_j \Big) = \\ &= f_i^{eq} - \frac{1}{n} \sum_{j \in U} f_j^{eq} + \frac{1}{n} \Big(\sum_j f_j^{eq} - \sum_{j \in K} f_j \Big) . \end{aligned}$$

At last,

$$f_i = f_i^{eq} - \frac{1}{n}\left(\sum_{j\in K} f_j - \sum_{j\in K} f_j^{eq}\right) \qquad (44)$$

As a reminder, K denotes known populations indexes, U denotes unknown population indexes; also, it is assumed (everywhere in this paper) that lattice vectors are rearranged so that $U=\overline{1,n}$ before unknown populations recovering.

6.3. Detailed Zou-He Boundaries Justification

First of all, let's show that momentum excess vector is induced just by the wall particle populations and therefore is parallel to the wall after the bounce-back procedure.

Let's introduce the following particle indexes sets: W for wall particle populations, I for particle populations inside the fluid, O for particle populations outside the fluid. It's obvious that $I=U$, $K=O\cup W$. Also, as usual, stroke denotes an opposite vector index. Each unknown population, directed inside the fluid, has a known population outside the fluid. Now we are going to compute node momentum after the bounce-back procedure (28):

$$\boldsymbol{M}^{\boldsymbol{bb}}=\sum_{i\in W}\boldsymbol{c}_i f_i^{bb}+\sum_{i\in I}\boldsymbol{c}_i f_i^{bb}+\sum_{i\in O}\boldsymbol{c}_i f_i^{bb}=\sum_{i\in W}\boldsymbol{c}_i f_i+\sum_{i\in I}(\boldsymbol{c}_i f_i^{bb}+\boldsymbol{c}_{i'} f_{i'}) \quad .$$

Let's investigate the behavior of each pair of known-unknown populations from the second sum:

$$\begin{aligned}\boldsymbol{c}_i f_i^{bb}+\boldsymbol{c}_{i'} f_{i'}=\boldsymbol{c}_i(f_i^{eq}+f_{i'}-f_{i'}^{eq})+\boldsymbol{c}_{i'} f_{i'}=\boldsymbol{c}_i f_i^{eq}+\boldsymbol{c}_i f_{i'}-\boldsymbol{c}_i f_{i'}^{eq}-\boldsymbol{c}_i f_{i'}= \\ =\boldsymbol{c}_i f_i^{eq}-\boldsymbol{c}_i f_{i'}^{eq}=\boldsymbol{c}_i f_i^{eq}+\boldsymbol{c}_{i'} f_{i'}^{eq} .\end{aligned}$$

Therefore, $\boldsymbol{M}^{\boldsymbol{bb}}=\sum_{i\in W}\boldsymbol{c}_i f_i+\sum_{i\in I}(\boldsymbol{c}_i f_i^{eq}+\boldsymbol{c}_{i'} f_{i'}^{eq})=\sum_{i\in W}\boldsymbol{c}_i f_i+\sum_{i\in I}\boldsymbol{c}_i f_i^{eq}+\sum_{i\in O}\boldsymbol{c}_i f_i^{eq}$.

According to (29), momentum excess is $\boldsymbol{E}=\boldsymbol{M}^{\boldsymbol{bb}}-\boldsymbol{M}^{\boldsymbol{eq}}=\sum_{i\in W}\boldsymbol{c}_i f_i-\sum_{i\in W}\boldsymbol{c}_i f_i^{eq}$, so it's indeed induced only by wall lattice vectors and, therefore, is parallel to the wall.

In the next step we show that the solution (31) actually minimizes (30).

First of all, as usual, let's compute derivatives of (30):

$$\frac{\partial L}{\partial f_i}=2(f_i-f_i^{bb})+\sum_{j=1}^{d}\lambda_j(\boldsymbol{c_i})_j=2(f_i-f_i^{bb})+\boldsymbol{\lambda}\cdot\boldsymbol{c_i}\quad,\quad i=\overline{1,n} \tag{45}$$

$$\frac{\partial L}{\partial\lambda_i}=\sum_{p=1}^{n}f_p(\boldsymbol{c_p})_i-\sum_{p=1}^{n}f_p^{bb}(\boldsymbol{c_p})_i+E_i\quad,\quad i=\overline{1,d} \tag{46}$$

We are going to insert the solution (31) ($f_i=f_i^{bb}-\boldsymbol{c_i}\cdot\boldsymbol{e}$, $e_i=E_i/n_i$, n_i is the number of lattice vectors with unknown populations which are not zero in direction i), into these equations and check, whether they become zero and whether the constraints are determined unambiguously. After that, taking into consideration the solvability conditions from the corresponding section, we conclude that the Zou-He solution is unique and always achievable on plane walls.

First of all, let's insert the solution into (46). Actually, the following proof is redundant, as it just ensures that the proposed solution resolves the boundary velocity constraints, and this is mentioned in the original paper.

Nevertheless, one obtains

$$\frac{\partial L}{\partial\lambda_i}=\sum_{p=1}^{n}(\boldsymbol{c_p})_i(f_p^{bb}-\boldsymbol{c_p}\cdot\boldsymbol{e})-\sum_{p=1}^{n}f_p^{bb}(\boldsymbol{c_p})_i+E_i=E_i-\sum_{p=1}^{n}(\boldsymbol{c_p})_i(\boldsymbol{c_p}\cdot\boldsymbol{e})=$$
$$=E_i-\sum_{p=1}^{n}(\boldsymbol{c_p})_i\sum_{j=1}^{d}(\boldsymbol{c_p})_je_j=E_i-\sum_{j=1}^{d}e_j\sum_{p=1}^{n}(\boldsymbol{c_p})_i(\boldsymbol{c_p})_j.$$

Consequently,

$$\frac{\partial L}{\partial\lambda_i}=E_i-\sum_{j=1}^{d}T_{ij}e_j\quad\text{or}\quad\frac{\partial L}{\partial\boldsymbol{\lambda}}=\boldsymbol{E}-T\boldsymbol{e}\quad, \tag{47}$$

Where $T_{ij}=\sum_{p=1}^{n}(\boldsymbol{c_p})_i(\boldsymbol{c_p})_j\quad,\quad i,j=\overline{1,d}$.

Now we are going to show, that

$$T_{ij}=\sum_{p=1}^{n}(\boldsymbol{c}_{\boldsymbol{p}})_i(\boldsymbol{c}_{\boldsymbol{p}})_j=n_i\delta_{ij},\quad i,j=\overline{1,d}\quad . \tag{48}$$

If $i=j$ this is simply by definition (33).

In case when $i\neq j$ and $i,j<3$ (i.e. do not include wall normal) let $\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}}$ be a lattice vector c_p , projected on the wall (see Fig 1 as an example of such a projection), let D be a set of diagonal projected lattice vectors of unknown populations, ND a set of non-diagonal projected lattice vectors of unknown populations. Then

$$T_{ij}=\sum_{p=1}^{n}(\boldsymbol{c}_{\boldsymbol{p}})_i(\boldsymbol{c}_{\boldsymbol{p}})_j=\sum_{p=1}^{n}(\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}})_i(\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}})_j=$$
$$=\sum_{p\in ND}(\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}})_i(\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}})_j+\sum_{p\in D}(\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}})_i(\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}})_j=\sum_{p\in D}(\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}})_i(\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}})_j=0.$$

Here it is taken into account that one of the coordinates of non-diagonal vectors will always be zero if $i\neq j$. Finally, due to the lattice symmetry, diagonal projected vectors will always come in pairs, and they would compensate each other.

In case when $i\neq j$ and either $i=3$ or $j=3$ (for example, $i=3$) we need to notice, that all the unknown vectors on plane boundaries have equal non-zero coordinates in the direction of wall normal, z direction (either 1 or -1). Let's denote this coordinate as s . Then $T_{ij}=\sum_{p=1}^{n}(\boldsymbol{c}_{\boldsymbol{p}})_3(\boldsymbol{c}_{\boldsymbol{p}})_j=s\sum_{p=1}^{n}(\boldsymbol{c}_{\boldsymbol{p}})_j=s\sum_{p=1}^{n}(\boldsymbol{c}_{\boldsymbol{p}}^{\boldsymbol{w}})_j$. The sum vanishes due to the lattice symmetry (see Fig 1).

Finally, we have proven (48). Let's insert it into (47). Then $\frac{\partial L}{\partial\lambda_i}=e_i n_i-\sum_{j=1}^{d}\delta_{ij}n_i e_j=0$.

Now we have to prove that the Zou-He solution is capable of minimizing the Lagrangian by particle populations. Indeed, after inserting this solution into (45)one obtains

$\frac{\partial L}{\partial f_i}=2(f_i-f_i^{bb})+\boldsymbol{\lambda}\cdot\boldsymbol{c_i}=-2\boldsymbol{e}\cdot\boldsymbol{c}_i+\boldsymbol{\lambda}\cdot\boldsymbol{c}_i$, $i=\overline{1,n}$; so $\boldsymbol{\lambda}=2\boldsymbol{e}$. It means that d Lagrangian multipliers can be unambiguously determined by n equations.

At last, we have proven that the Zou-He solution is the one that ensures wall velocity constraints with minimum deviation from bounced-back particle populations.

7. Bibliography

[1] R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous, and D. Joseph, “The lattice Boltzmann equation method: theoretical interpretation, numerics and implications,” *International Journal of Multiphase Flow*, vol. 29, Jan. 2003, pp. 117-169.

[2] C. Körner, T. Pohl, U. Rüde, N. Thürey, and T. Zeiser, “Parallel Lattice Boltzmann Methods for CFD Applications,” *Numerical Solution of Partial Differential Equations on Parallel Computers*, 2006, pp. 439-466.

[3] S. Khirevich, A. Höltzel, D. Hlushkou, and U. Tallarek, “Impact of conduit geometry and bed porosity on flow and dispersion in noncylindrical sphere packings,” *Analytical Chemistry*, vol. 79, Dec. 2007, pp. 9340-9349.

[4] J. Latt, B. Chopard, O. Malaspinas, M. Deville, and A. Michler, “Straight velocity boundaries in the lattice Boltzmann method,” *Physical Review E*, vol. 77, May. 2008, p. 56703.

[5] Z. Guo, C. Zheng, and B. Shi, “An extrapolation method for boundary conditions in lattice Boltzmann method,” *Physics of Fluids*, vol. 14, Jun. 2002, pp. 2007-2010.

[6] R. Mei, L. Luo, and W. Shyy, “An Accurate Curved Boundary Treatment in the Lattice Boltzmann Method,” *Journal of Computational Physics*, vol. 155, Nov. 1999, pp. 307-330.

[7] I. Ginzburg and D. d’Humières, “Multireflection boundary conditions for lattice Boltzmann models,” *Physical Review E*, vol. 68, Dec. 2003, p. 066614.

[8] S. Chen, D. Martinez, and R. Mei, “On boundary conditions in lattice Boltzmann methods,” *Physics of Fluids*, vol. 8, 1996, pp. 2527-2536.

[9] M. Junk and Z. Yang, "Asymptotic Analysis of Lattice Boltzmann Boundary Conditions," *Journal of Statistical Physics*, vol. 121, Oct. 2005, pp. 3-35.

[10] Q. Zou and X. He, "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model," *Physics of Fluids*, vol. 9, Jun. 1997, pp. 1591-1598.

[11] G. Barrios, R. Rechtman, J. Rojas, and R. Tovar, "The lattice Boltzmann equation for natural convection in a two-dimensional cavity with a partially heated wall," *Journal of Fluid Mechanics*, vol. 522, Jan. 2005, pp. 91-100.

[12] X. He and L. Luo, "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation," *Physical Review E*, vol. 56, Dec. 1997, p. 6811.

[13] A. Parmigiani, C. Huber, B. Chopard, J. Latt, and O. Bachmann, "Application of the multi distribution function lattice Boltzmann approach to thermal flows," *The European Physical Journal - Special Topics*, vol. 171, Apr. 2009, pp. 37-43.

[14] Y. Peng, C. Shu, and Y.T. Chew, "Simplified thermal lattice Boltzmann model for incompressible thermal flows," *Physical Review E*, vol. 68, 2003, p. 026701.

[15] B.J. Palmer and D.R. Rector, "Lattice-Boltzmann algorithm for simulating thermal two-phase flow," *Physical Review E*, vol. 61, May. 2000, p. 5295.

[16] R. Maier and R. Bernard, "Lattice-Boltzmann accuracy in pore-scale flow simulation," *Journal of Computational Physics*, vol. 229, Jan. 2010, pp. 233-255.

[17] L.D. Landau and E.M. Lifshitz, *Fluid mechanics*, London : Reading, Mass. :: Pergamon Press ; Addison-Wesley, .

[18] E. Marshall and E. Trowbridge, "Flow of a newtonian fluid through a permeable tube: The application to the proximal renal tubule," *Bulletin of Mathematical Biology*, vol. 36, Oct. 1974, pp. 457-476.

[19] S.H. Munson-McGee, “An approximate analytical solution for the fluid dynamics of laminar flow in a porous tube,” *Journal of Membrane Science*, vol. 197, Mar. 2002, pp. 223-230.

[20] R.M. Terrill and P.W. Thomas, “On laminar flow through a uniformly porous pipe,” *Applied Scientific Research*, vol. 21, Jan. 1969, pp. 37-67.

[21] S. Tsangaris, D. Kondaxakis, and N. Vlachakis, “Exact solution for flow in a porous pipe with unsteady wall suction and/or injection,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 12, Oct. 2007, pp. 1181-1189.

[22] S. Yuan and A.B. Filkenstein, “Laminar pipe flow with injection and suction through a porous wall,” 1956.

[23] L. Oxarango, P. Schmitz, and M. Quintard, “Laminar flow in channels with wall suction or injection: a new model to study multi-channel filtration systems,” *Chemical Engineering Science*, vol. 59, Mar. 2004, pp. 1039-1051.