Lattice Boltzmann boundary conditions via singular forces: irregular expansion analysis and numerical investigations

Alfonso Caiazzo [a], Shankar Maddu [b]

[a] *University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, the Netherlands*

[b] *Fraunhofer ITWM, Fraunhofer-Platz 1, D-67663 Kaiserslautern, Germany*

Abstract

We present an investigation of a boundary condition algorithm for the lattice Boltzmann method, which introduces Dirichlet conditions on velocity using a singular force applied on the fluid-solid interface (immersed boundary method). The algorithm has been proposed in the literature in different versions and mainly numerically tested, only in specific cases. An approach based on a generalized asymptotic expansion technique will be used to understand properties and point out problems of the scheme. As a result, we found that the algorithm achieves a first order accurate velocity in a strong sense, while accuracy for the pressure can be stated only considering a weak norm. Moreover, the analysis predicts a first order accuracy for the boundary force although the precision is affected by stability limitations. We benchmark the method on lattice Boltzmann flows past a rigid disk, comparing its numerical performances with standard boundary condition approaches.

Key words: lattice Boltzmann method, boundary conditions, asymptotic analysis, immersed boundary method.

1 Introduction

The lattice Boltzmann method (LBM) [8,14,19], in virtue of its simple formulation and favorable implementation, is an alternative numerical method for solving the incompressible Navier-Stokes equations used in many applications.

Email addresses: acaiazzo@science.uva.nl (Alfonso Caiazzo), maddu@itwm.fhg.de (Shankar Maddu).

Due to its relative novelty, many features are object of current investigations and still represent challenging tasks. We focus on Dirichlet boundary conditions.

One possibility, which we will call *classical BC approach*, consists of dividing the domain into a fluid and a solid part, and specifying a boundary rule for the computational fluid nodes close to the interface. Algorithms belonging to this family have been proposed and analyzed in several works (see for example [2,6,10,15]). The *immersed boundary* methods are based on a different idea. In this approach, the *computational* fluid domain discretizes both fluid and solid *physical* sub-domains, while the information about the interface and the boundary conditions is introduced via an additional force applied at the boundary nodes, i.e. the lattice nodes close to the interface. As a consequence the algorithmic interface is *immersed* into the discrete lattice, and it spreads over a grid cell. Using finer discretizations the width of cells decreases, while the intensity of the applied force increases, resulting in a *singular source* in the limit. Basic ideas of a singular force approach for the Navier-Stokes equation (out of the LB framework) can be found, for example, in [13].

Within the LBM, a first immersed boundary approach was described in [7]. Later on a novel version was proposed in [18]. Both these approaches can be directly related to the original idea of immersed boundary described above.

Earlier [1], a model to simulate infinitely small particles in flow was originally proposed, based on a coupling of an LB-flow solver with a Molecular Dynamics (MD) model. The solid particles were modeled as points, introducing a friction force depending on velocity (according to Stokes law) at the closest lattice nodes. A later generalization of this approach [5] has been used for simulating colloidal particles in flow. Even if derived from different ideas, the algorithms in this form can be written as an immersed boundary methods.

To our knowledge, only restricted sets of numerical examples have been presented in all these formulations. It must be remarked that in [1,5] the algorithm was designed for a very specific application. However, in the interest of the community, we investigate possible generalizations.

The scope of this article is twofold. First we aim at a better understanding of the algorithm based on an asymptotic analysis. Of practical interest is the study of possible realizations and performances of the algorithm, while comparing with classical approaches, to investigate whether it represents a valid alternative. Secondly, from the theoretical point of view, due to the presence of singular terms we need to develop a generalized formulation of the classical analysis approach, which allows non-smoothness restricted to regions close to boundary. We show how weak estimates can be derived, verifying them numerically on a benchmark problem.

The article is organized as follows. The lattice Boltzmann method is briefly introduced in section 2. In section 3 an implementation of boundary conditions via singular force is described together with a first numerical test. Section 4 presents the asymptotic analysis, the generalization to non-smooth expansions and further numerical investigations concerning force evaluation and stability issues. Conclusions and discussion are presented in section 5.

2 Lattice Boltzmann method

Let us consider a domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$, divided into a fluid part $\Omega_F(t)$, a solid part $\Omega_S(t)$ and the interface $\Gamma(t)$ between them:

$$\Omega = \Omega_F(t) \cup \Gamma(t) \cup \Omega_S(t), \tag{1}$$

In the fluid sub-domain, we consider an incompressible Navier-Stokes problem

$$\begin{aligned} &\begin{cases} \nabla \cdot \mathbf{u} = 0 & t \in (0,T],\ \mathbf{x} \in \Omega_F(t) \\ \partial_t \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \Delta \mathbf{u} + \mathbf{G} & t \in (0,T],\ \mathbf{x} \in \Omega_F(t) \end{cases} \\ &\mathbf{u}(t,\mathbf{x}) = \mathbf{u}_B(t,\mathbf{x}) \quad t \in (0,T],\ \mathbf{x} \in \Gamma(t) \\ &\mathbf{u}(0,\mathbf{x}) = \mathbf{u}_0(\mathbf{x}) \quad \mathbf{x} \in \Omega_F(0), \end{aligned} \tag{2}$$

$\mathbf{u}_0(\mathbf{x})$ being the initial fluid velocity and $\mathbf{u}_B(t,\mathbf{x})$ the prescribed velocity at the fluid-solid interface. We assume that $\Omega_S(t)$ is a rigid body with a given motion, i.e. $\Gamma(t)$ is known function of time.

The lattice Boltzmann method is employed to solve numerically (2). The algorithm is defined by discretizing the spatial domain with a Cartesian lattice $h\mathbb{Z}^d$, where $\mathbf{j} \in \mathbb{Z}^d$ denotes the coordinates of a generic node, and the time domain $[0,T]$ with discrete nodes $t_n = h^2 n$, $n \geq 0$. The relation $\Delta t = \Delta x^2$, called in literature the *diffusive scaling*, is necessary to recover the incompressible problem (2) [9]. In what follows, we introduce

$$\mathbb{G}(h) = \mathbb{Z}^d \cap h^{-1}\Omega = \left\{ \mathbf{j} \in \mathbb{Z}^d \mid h\mathbf{j} \in \Omega \right\}$$

to denote the set of integer coordinates of all computational nodes.

The general iteration of the algorithm reads

$$\hat{f}_i(n+1, \mathbf{j} + \mathbf{c}_i) = \hat{f}_i(n, \mathbf{j}) + J_i(\hat{f})(n, \mathbf{j}) + g_i(n, \mathbf{j}) \tag{3}$$

where $\mathbb{V} = \{\mathbf{c}_i \mid i = 1, \ldots, b\}$ is a discrete velocity set, compatible with the Cartesian lattice (i.e. the discrete velocity vectors connect neighboring nodes of the lattice). In this work, we consider the D2Q9 model with nine velocities in 2D (for details, see [19] and the references therein), depicted in figure

1a. The variable $\hat{f}_i(n, \mathbf{j})$ represents the numerical solution for the density of particles moving in direction $\mathbf{c}_i$ at time $t_n = h^2 n$ and position $\mathbf{x} = h\mathbf{j}$. For the collision operator $J_i(\hat{f})$ on the right hand side of equation (3), we use the BGK approximation

$$J(\hat{f}) = \frac{1}{\tau}(f^{eq}(\hat{f}) - \hat{f}), \tag{4}$$

i.e. a linear relaxation [1] towards an *equilibrium distribution*

$$f^{eq}(\hat{f}) = H_i^{eq}\left(\rho(\hat{f}), \mathbf{u}(\hat{f})\right),$$

which is a function of $\hat{f}$ through the local density $\hat{\rho} = \rho(\hat{f}) = \sum_i \hat{f}_i$ and the local velocity $\hat{\mathbf{u}} = \mathbf{u}(\hat{f}) = \sum_i \mathbf{c}_i \hat{f}_i$ related to the particle distributions.

For the D2Q9 model

$$H_i^{eq}(\rho, \mathbf{u}) = f_i^* \left(\rho + c_s^{-2}\mathbf{c}_i \cdot \mathbf{u} + \frac{c_s^{-4}}{2}\left(|\mathbf{c}_i \cdot \mathbf{u}|^2 - c_s^2 \mathbf{u}^2\right)\right), \tag{5}$$

where the lattice sound speed c_s and the weights f_i^* are model dependent constants [19].

Finally, τ is related to the viscosity via $\nu = c_s^2(\tau - \frac{1}{2})$ and the additional term g_i is used to include the volume force $\mathbf{G}$ appearing in the Navier-Stokes problem (2):

$$g_i(n, \mathbf{j}) = h^3 c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{G}(t_n, \mathbf{x_j}). \tag{6}$$

The classical implementation of algorithm (3) is usually split into *collision* and *propagation* sub-steps

$$\begin{aligned} &(C) \quad \hat{f}_i^C(n, \mathbf{j}) = \hat{f}_i(n, \mathbf{j}) + \frac{1}{\tau}(f_i^{eq}(\hat{f}) - \hat{f}_i)(n, \mathbf{j}) + g_i(n, \mathbf{j}), \\ &(P) \quad \hat{f}_i(n+1, \mathbf{j} + \mathbf{c}_i) = \hat{f}_i^C(n, \mathbf{j}) \end{aligned} \tag{7}$$

(where $\hat{f}^C$ is called post-collision distribution).

LB boundary conditions In case of *classical BC approaches*, Ω is divided into solid and fluid sub-domains and the LB variables are defined only on the fluid part. Hence, for a node which has at least a neighbor in the solid domain, a boundary condition rule is necessary (instead of the update (7)) for the *incoming directions*, i.e. the LB variables entering the fluid domain.

The immersed boundary approaches (section 3), does not distinguish fluid and solid nodes. However, in order to make comparisons with classical BC rules

[1] We focus on a single relaxation time model. However, the analysis can be applied also to more general *multiple relaxation time* (MRT) models.

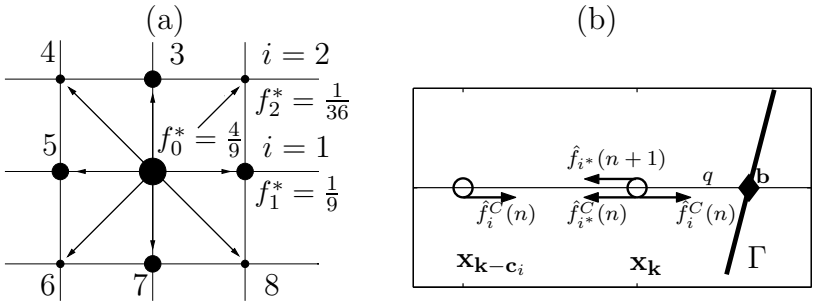

Figure 1. **(a)** The D2Q9 model. Bigger circles indicate larger weights f_i^*. **(b)** Sketch of the BFL boundary conditions (8). To update $\hat{f}_{i^*}$ at the boundary node $\mathbf{k}$, a combination of the populations *after collision* at two neighbor nodes is used, depending on the distance q between the boundary and $\mathbf{x_k}$, and the boundary velocity.

for LBM, we use the scheme proposed by Bouzidi, Firdaouss, Lallemand in [2] (BFL rule), where the incoming variables are defined extrapolating from the LB variables on the neighboring nodes:

$$\hat{f}_{i^*}(n+1,\mathbf{k}) = \begin{cases} 2q\hat{f}_i^C(n,\mathbf{k}) + (1-2q)\hat{f}_i^C(n,\mathbf{k}-\mathbf{c}_i) + \\ \qquad 2c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}_B & q \le \frac{1}{2} \\ \frac{1}{2q}\hat{f}_i^C(n,\mathbf{k}) + (1-\frac{1}{2q})\hat{f}_{i^*}^C(n,\mathbf{k}) + \\ \qquad \frac{1}{q} c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}_B & q > \frac{1}{2} \end{cases} \tag{8}$$

where q denotes the relative distance between the fluid node $\mathbf{k}$ and boundary along the link $\mathbf{c}_i$ and $\mathbf{u}_B$ is the velocity of the boundary point along $\mathbf{c}_i$ (see figure 1b).

3 Force boundary conditions

In this section we describe in detail an immersed boundary approach, based on [1,5], which can be seen as a particular version of the method in [7,18].

Interface discretization. Primarily, the physical fluid-solid interface Γ is discretized using a set of points (see figure 2a), called *boundary molecules*:

$$\mathcal{M} = \{\mathbf{P}_1, \ldots, \mathbf{P}_{N_b}\} \subset \Gamma. \tag{9}$$

The set $\mathcal{M}$ is constructed using a partition of amplitude $O(h)$ of the boundary. The number of points N_b depends on h and on the dimension of the interface (a discretization as fine as the LB grid gives $N_b \sim \frac{1}{h}$ in 2D and $N_b \sim \frac{1}{h^2}$ in 3D).

Friction force. According to the original model [1,5], each boundary molecule feels an ideal friction force due to the fluid flow, which depends on the difference in velocity. We observe that in [7] an analogous force is introduced, in

the form of a *penalty term*, i.e. an additional source designed to reproduce the desired boundary condition.

Considering a generic $\mathbf{P}_m \in \mathcal{M}$, calling $\mathbf{V}_m$ its velocity, we have:

$$\mathbf{F}_m(t) = \xi(\mathbf{u}(t, \mathbf{P}_m) - \mathbf{V}_m(t)), \tag{10}$$

where $\mathbf{u}(t, \mathbf{P}_m)$ is the fluid velocity at $\mathbf{P}_m$ and ξ is an opportune friction coefficient. Since the fluid velocity at $\mathbf{P}_m$ is in general not available, we use the force

$$\tilde{\mathbf{F}}_m(h, t, \mathbf{u}) = -\hat{\xi}_h \left(\tilde{\mathbf{u}}_m(h) - \mathbf{V}_m(t)\right), \tag{11}$$

depending on $\tilde{\mathbf{u}}_m$, i.e. an approximation of $\mathbf{u}(t, \mathbf{P}_m)$ based on the values at the nearest lattice nodes. Note that the resulting force depends on h through the approximated velocity. Additionally, we have introduced a h-dependent friction coefficient $\hat{\xi}_h$ (the need of this parameter will be clarified later on).

In practice, for the algorithm described below and for the presented numerical results, we adopted a bilinear interpolation for $\tilde{\mathbf{u}}_m$ [2].

The singular force on the LB nodes. Finally, an equal intensity force has to be applied on the fluid. In the LBM (3) this is done using a forcing term in the collision step at the nodes close to the interface. For each $\mathbf{P}_m$ we select a neighborhood

$$\mathcal{B}_h(\mathbf{P}_m) = \left\{\mathbf{j} \in \mathbb{G}(h) \mid \left|\mathbf{x}_\mathbf{j} - \mathbf{P}_m\right| < 2h\right\} \tag{12}$$

and distribute the friction force on the nodes belonging to it (figure 2a). Namely, the force

$$\mathbf{F}_m^{LB}(h, t_n, \mathbf{x}_\mathbf{j}, \mathbf{u}) = w_m^F(h, \mathbf{j})(-\tilde{\mathbf{F}}_m(h, t_n, \mathbf{u})), \tag{13}$$

such that

$$\sum_\mathbf{j} (2h)^d \mathbf{F}_m^{LB} = -\tilde{\mathbf{F}}_m \tag{14}$$

(being $(2h)^d$ the volume of $\mathcal{B}_h(\mathbf{P}_m)$, i.e. of the fluid which feels the interface

[2] As possible alternatives, we have investigated higher order approximations and *side-dependent* interpolations (i.e. computing $\tilde{\mathbf{u}}_m$ using lattice nodes on the same side of the interface). More complicated approximation routines could produce better results in some cases, but lead in general to considerable increase of computational effort. Furthermore, using wider stencils of points, the level of technicality of the algorithm increases, also due to the need of special geometrical configurations to be taken into account, making it comparable in efficiency with classical BC approaches.

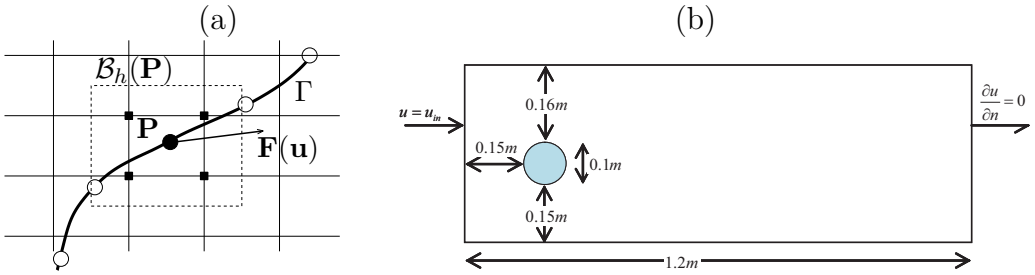

Figure 2. **(a)** The fluid-solid interface Γ is discretized using a set of boundary molecules ($\circ$), with mutual distance of order of h. After computing the friction force on a boundary molecule $\mathbf{P}$, it is distributed on the LB nodes ($\blacksquare$) belonging to h-neighborhood $\mathcal{B}_h(\mathbf{P})$(dashed-edges square). **(b)** Considered Benchmark. In a channel of length $L = 1.2m$ and width $W = 0.41m$ we place a rigid disk of radius $R = 0.05m$. The center is located at $x_C = 0.2m$, $y_C = 0.2m$ (small vertical offset). We use a parabolic inflow (with maximum velocity $U = 0.3\frac{m}{s}$) on the right boundary and a homogeneous Neumann condition at the outflow, using the implementation proposed and analyzed in [11]. Viscosity is $\nu = 0.005$.

force) is applied to the LB nodes close to the interface via

$$g_i^{BC}(n, \mathbf{j}) = h^3 f_i^* c_s^{-2} \mathbf{c}_i \cdot \sum_{m=1}^{N_b(h)} \mathbf{F}_m^{LB}(h, t_n, \mathbf{x}_\mathbf{j}, \mathbf{u}). \tag{15}$$

Expression (13) and (15) can be defined globally on the whole LB lattice, setting $w_m^F(h, \mathbf{j}) = 0$ at the interior nodes (the nodes far from the boundary).

Remarks. The friction force acting on the boundary molecule can be interpreted as a discrete approximation (according to the weights $w_m^F(h, \mathbf{j})$) of a delta function centered at $\mathbf{P}_m$. The numerical interface Γ spreads over a narrow band of width $O(h)$. As the width decreases (finer grids), the intensity of the force increases proportionally, resulting for $h \to 0$ in a singular force. Observe that the force vanishes when the boundary condition

$$\mathbf{u}(t, \mathbf{x}) = \mathbf{u}_\Gamma(t, \mathbf{x}), \ \forall \mathbf{x} \in \Gamma(t)$$

is satisfied.

Evaluation of the boundary force. The total boundary force acting on the solid is evaluated (as proposed in [5]) taking the sum of all the friction forces acting on the ideal particles [3]:

$$\hat{\mathbf{F}}_h^S(t_n) = \sum_{m=1}^{N_b(h)} \tilde{\mathbf{F}}_m(h, t_n, \mathbf{u}) \, . \tag{16}$$

[3] The total boundary torque can be computed in a similar way, considering the torques of the local friction force respect to the center of mass of the solid.

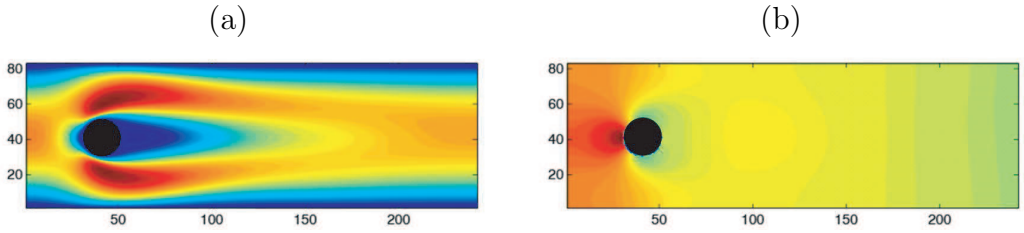

Figure 3. Qualitative results of forceBC algorithm. **(a)** Contour lines of $u^2 + v^2$.**(b)** Pressure field.

The force boundary condition algorithm can be summarized as follows:

Algorithm 1 (forceBC)
at time t_n, given $\mathbf{u}(t_n, \cdot)$, $\Gamma(t_n)$:
  `define the set of boundary molecules` $\mathcal{M} = \{\mathbf{P}_1, \ldots, \mathbf{P}_{N_b(h)}\}$
P-loop: `for` $m = 1, \ldots, N_b(h)$
  `compute weights:` $w_m^{\mathbf{u}}(h, \mathbf{j}_l)$, $\mathbf{j}_l \in \mathcal{B}_h(\mathbf{P}_m)$ *(for closest nodes)*
  `compute` $\tilde{\mathbf{u}}_m(t_n)$, $\tilde{\mathbf{F}}_m = \hat{\xi}_h(\tilde{\mathbf{u}}_m(t_n) - \mathbf{V}_m(t_n))$
  `for` $\mathbf{j}_l \in \mathcal{B}_h(\mathbf{P}_m)$
    `compute` $w_m^F(h, \mathbf{j}_l)$, $\mathbf{F}_m^{LB}(h, t_n, \mathbf{x}_\mathbf{j}, \mathbf{u})$
      $g_i(n, \mathbf{j}) = g_i(n, \mathbf{j}) + h^3 f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{F}_m^{LB}(h, t_n, \mathbf{x}_\mathbf{j}, \mathbf{u})$
      (note: contributions have to be summed up on each LB node)
  `end`
`end` *(P-loop)*
`LB-collision` (with additional force where needed)
`LB-propagation`
`compute force and torque on particle:`
  $\hat{\mathbf{F}}_h^S(n) = \sum_{m=1}^{N_b(h)} \tilde{\mathbf{F}}_m$, $\hat{\mathbf{T}}_h^S(n) = \sum_{m=1}^{N_b(h)} (\mathbf{x}_C - \mathbf{P}_m) \times \tilde{\mathbf{F}}_m$
`update particle position and velocity`

Benchmarks: disk in channel The force boundary condition algorithm is used to simulate the flow in a channel past a rigid disk, according to the benchmark proposed in [17]. The parameters are provided in detail in figure 2b. We compare the results of the force boundary condition algorithm with the ones obtained employing the BFL rule (8). As an example of application, in figure 3 we report the results for the flow field at the steady state.

4 Asymptotic Analysis and Numerical Tests

Algorithm 1 is analyzed using the asymptotic expansion technique. We will first summarize the argument for the classical LBM (following [9]), which, in

our case, corresponds to the algorithm employed at the *interior* lattice node, i.e. far from the interface.

Basically we search an approximation of the LB solution $\hat{f}_h$ in form of power series of the lattice step h

$$\hat{f}_h(n,\mathbf{j}) \approx F_h(n,\mathbf{j}) = f_i^{(0)}(t_n,\mathbf{x_j}) + hf_i^{(1)}(t_n,\mathbf{x_j}) + h^2 f_i^{(2)}(t_n,\mathbf{x_j}) + \dots, \quad (17)$$

with h-independent coefficients $f^{(k)}$, smooth functions of the physical time t and space $\mathbf{x}$. The function F_h is called *ansatz*. To determine the functions $f^{(k)}$ we insert the ansatz (17) into the algorithm

$$\hat{f}_i(n+1,\mathbf{j}+\mathbf{c}_i) = \hat{f}_i(n,\mathbf{j}) + \frac{1}{\tau}\left(f_i^{eq}(f(n,\mathbf{j})) - f_i(n,\mathbf{j})\right) + g_i^{BC}(n,\mathbf{j}). \quad (18)$$

Using a Taylor expansion, sorting the orders in h and equating each order to zero in the resulting expression, we obtain a set of (partial differential) equations for the coefficients $f^{(k)}$. It can be shown that the choices (see [9] for details)

$$\begin{aligned} f_i^{reg,(0)} &= f_i^*, \\ f_i^{reg,(1)} &= f_i^* c_s^{-2}\mathbf{c}_i\cdot\mathbf{u}, \\ f_i^{reg,(2)} &= f_i^* c_s^{-2} p + \frac{f_i^* c_s^{-4}}{2}\left(|\mathbf{c}_i\cdot\mathbf{u}|^2 - c_s^2\mathbf{u}^2\right) - \tau f_i^* c_s^{-2}\mathbf{c}_i\cdot\nabla\mathbf{u}\cdot\mathbf{c}_i, \end{aligned} \quad (19)$$

where $\mathbf{u}$ and p are a solution of the Navier-Stokes problem (2), define a prediction F_h satisfying (18) up to a residue of order h^3. To stress that the above relations have been obtained considering a regular expansion for the algorithm in the bulk flow, in what follows we refer to the functions in (19) as *interior* or *regular coefficients*, denoting them with $f^{reg,(k)}$. The (interior) numerical method is analyzed using the truncated expansion

$$F_{ih}^{reg} = f_i^{reg,(0)} + hf_i^{reg,(1)} + h^2 f_i^{reg,(2)}, \quad (20)$$

which we call *regular prediction*. Since we can extract the Navier-Stokes solution taking suitable moments of F_h^{reg}, we conclude that the corresponding moments of the numerical solution

$$\begin{aligned} h^{-1}\hat{\mathbf{u}} &= \frac{\sum_i \mathbf{c}_i \hat{f}_{ih}}{h} = \mathbf{u} + O(h^2), \\ \hat{p} &= c_s^2 \frac{\sum_i \hat{f}_{ih} - 1}{h^2} = p + O(h), \end{aligned} \quad (21)$$

yield a second order accurate velocity and a first order accurate pressure. Additionally, we can approximate (up to first order in h) the viscous stress

tensor using

$$\begin{aligned}\hat{\mathbf{S}}[\mathbf{u}] &= -\frac{\nu}{c_s^2 \tau h^2} \sum_i \mathbf{c}_i \otimes \mathbf{c}_i \left(\hat{f}_i - f_i^{eq}(\hat{f}) \right) = \\ &= \nu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) + O(h).\end{aligned} \tag{22}$$

The BFL boundary condition scheme (8) can be analyzed in a similar way [10], giving results consistent with (19), that proves the same accuracy properties (21)-(22).

The forceBC algorithm 1 differs from the LBM (18) only at the lattice nodes close to the boundary, where

$$g_i^{BC}(n, \mathbf{j}) = f_i^* c_s^{-2} \mathbf{c}_i \cdot \sum_{m=1}^{N_b(h)} -\hat{\xi}_h h^3 w_m^F(h, \mathbf{j})(\tilde{\mathbf{u}}_m - \mathbf{V}_m) \neq 0. \tag{23}$$

To investigate the numerical solution at those points, we introduce a new ansatz, which has to be consistent with the interior prediction at the lattice nodes far from the interface.

Dealing with a singular source, we extend the classical analysis approach by requiring the coefficients $f^{(k)}$ to be smooth only *far from the interface*, i.e. on $\Omega \backslash \Gamma$, and allowing jumps across the boundary. In general, we can write the coefficient of order k as

$$f_i^{(k)} = f_i^{reg,(k)} + \phi_i^{(k)}(g^{BC}), \tag{24}$$

where $f_i^{reg,(k)}$ is the regular coefficient defined in (19) and $\phi_i^{(k)}$ depends on the forcing term g_i^{BC} (23) and contains the irregular parts.

Leading orders: boundary condition on velocity. We assume that the velocity is continuous across Γ, differentiable on both sides, with bounded gradient. Also, we assume the pressure to be smooth on the fluid and solid domain. A consequence of the continuity condition on the velocity is that the leading orders $f^{(0)}$ and $f^{(1)}$ of the expansion must be at least continuous as well. Comparing the coefficients on both sides of the interface yields the condition

$$\phi_i^{(0)} = 0, \ \phi_i^{(1)} = 0, \ \forall i = 1, \dots, b. \tag{25}$$

According to the analysis, allowing the friction coefficient to depend on h and defining $\hat{\xi}_h$ proportional to h^{d-2} (where d is the number of dimensions, i.e. ξ is independent from h in 2D and $O(h)$ in 3D), equation (25) gives

$$\phi_i^{(0)} = 0 \text{ (identically verified)}, \tag{26}$$

$$\phi_i^{(1)} = 0 \iff \mathbf{c}_i \cdot \left(\mathbf{u}^{(1)} - \mathbf{u}_\Gamma \right), \ \forall i = \{1, \dots, b\} \iff \mathbf{u}^{(1)} = \mathbf{u}_\Gamma. \tag{27}$$

In other words, when the Dirichlet boundary condition is satisfied the leading order of the singularity vanishes and interior (regular) expansion holds up to first order in h.

In general, without continuity properties of the coefficients, it is not possible to compare pointwise the quantities of higher orders. Therefore, strong accuracy properties of the velocity higher than first order in h cannot be stated. On the other hand, possible irregularity can be taken into account by considering integral fields and weak norms. This will be discussed in the following sections.

Higher orders and force evaluation. Considering the second order coefficient, we have to take into account the jumps in the numerical pressure and velocity gradient across Γ. For this purpose, for any continuous function $v : \Omega \backslash \Gamma \to \mathbb{R}$ we introduce the quantities

$$\forall \mathbf{b} \in \Gamma \, v^F(\mathbf{b}) = \lim_{\mathbf{x} \to \mathbf{b}, \mathbf{x} \in \Omega_F} v(\mathbf{x}), \; v^S(\mathbf{b}) = \lim_{\mathbf{x} \to \mathbf{b}, \mathbf{x} \in \Omega_S} v(\mathbf{x}), \tag{28}$$

and the jump across the interface in $\mathbf{b}$:

$$[v]_{\mathbf{b}} = v^F(\mathbf{b}) - v^S(\mathbf{b}).$$

In view of the previous observations, $\left[f^{(0)}\right]_{\mathbf{b}} = \left[f^{(1)}\right]_{\mathbf{b}} = 0$, for all $\mathbf{b} \in \Gamma$.

To analyze the force computation, the approach can be generalized considering $f^{(2)}$ as distributions and introducing an integral *semi-norm* based on a discrete integration over the h-size lattice

$$\forall v : \Omega \to \mathbb{R}, \; \|v\|_{1,h} := \Big\| \sum_{\mathbf{j} \in \mathbb{G}(h)} h^2 v(\mathbf{x_j}) \Big\|. \tag{29}$$

We investigate the quantity

$$\Delta^{(2)}_{1,h} = \| \sum_i \left(f^{(2)}_{i,h} - f^{reg,(2)}_{i,h} \right) \|_{1,h}. \tag{30}$$

i.e. the semi-norm of the difference $\rho(f^{(2)}) - \rho(f^{reg,(2)})$.

To evaluate (30), we need to sum over all the lattice nodes and over the discrete directions $\mathbf{c}_i$, for $i = 1, \ldots, b$.

First of all, we remark that $\phi^{(2)}_i$ can be different from zero only for the links $\mathbf{c}_i$ crossing the boundary. This can be seen considering the procedure we used to derive the regular prediction, starting from equation (18). At a boundary node $\mathbf{j}$, if $\mathbf{j}$ and $\mathbf{j}+\mathbf{c}_i$ are located on the same part with respect to the interface (fluid-fluid or solid-solid links) the residue can be Taylor expanded in term of smooth functions around $(t_n, \mathbf{x_j})$. In other words, the prediction contains only regular parts, i.e. $\phi^{(k)}_i = 0$ if $\mathbf{c}_i$ does not cross the interface. Hence, only the links connecting fluid and solid nodes give non-zero contribution in (30). Since these links identify uniquely intersection points between lattice and interface, the summation (30) can be rewritten as a sum over the set of points $\mathbf{b}(\mathbf{k}, i) \in \Gamma$ where a crossing link $\mathbf{c}_i$ starting from $\mathbf{k}$ intersects the boundary. According

to (19), assuming that the regular part of the second order coefficient has the same structure on both sides of the interface, i.e. for $\mathbf{b} \in \Gamma$

$$\begin{aligned} f_i^{F,(2)} &= f_i^* c_s^{-2} p^F + \frac{f_i^* c_s^{-4}}{2}\left(|\mathbf{c}_i \cdot \mathbf{u}|^2 - c_s^2 \mathbf{u}^2\right) - \tau f_i^* c_s^{-2} \mathbf{c}_i \cdot [\nabla \mathbf{u}]^F \cdot \mathbf{c}_i \\ f_i^{S,(2)} &= f_i^* c_s^{-2} p^S + \frac{f_i^* c_s^{-4}}{2}\left(|\mathbf{c}_i \cdot \mathbf{u}|^2 - c_s^2 \mathbf{u}^2\right) - \tau f_i^* c_s^{-2} \mathbf{c}_i \cdot [\nabla \mathbf{u}]^S \cdot \mathbf{c}_i \end{aligned} \tag{31}$$

(where F and S have the same meaning as in (28)), (30) can be expressed as a sum of discontinuities in pressure and stress tensor (i.e. gradient of the velocity) across the interface and a sum of the singular forces along the boundary:

$$\begin{aligned} &\Big\| \sum_{\mathbf{b}(\mathbf{k},i)} 2 f_i^* c_s^{-2} \left([p]_{\mathbf{b}} + \frac{c_s^{-2}}{2} \nu \left[\mathbf{c}_i \cdot \nabla \mathbf{u}_B \cdot \mathbf{c}_i \right]_{\mathbf{b}} \right) \mathbf{c}_i - \sum_{m=1}^{N_b(h)} \psi_m(\hat{\mathbf{u}}) \Big\| = \\ =&\Big\| \sum_{\mathbf{b}(\mathbf{k},i)} 2 f_i^* c_s^{-2} \left([p]_{\mathbf{b}} + \frac{c_s^{-2}}{2} \nu \left[\mathbf{c}_i \cdot \nabla \mathbf{u}_B \cdot \mathbf{c}_i \right]_{\mathbf{b}} \right) \mathbf{c}_i - \hat{\mathbf{F}}_h \Big\|, \end{aligned} \tag{32}$$

where $\psi_m(\hat{\mathbf{u}})$ abbreviates the friction force at $\mathbf{P}_m$ and

$$\hat{\mathbf{F}}_h(t_n) = \sum_{m=1}^{N_b(h)} \psi_m(\hat{\mathbf{u}}(t_n, \cdot)), \tag{33}$$

is the approximation of the boundary force used in algorithm 1. Following [3,4], the sum over $\mathbf{b}(\mathbf{k}, i)$ in (32) can be related to the integral [4]

$$\mathbf{F}_S(t_n) = \int_{\Gamma(t_n)} \Big(-p(t_n, \mathbf{x}) + \mathbf{S}[\mathbf{u}(t_n, \mathbf{x})] \Big) \cdot \mathbf{n}(\mathbf{x}) d\mathbf{x} \tag{34}$$

(being $\mathbf{S}[\mathbf{u}]$ the viscous stress tensor and $\mathbf{n}(\mathbf{x})$ the local outgoing normal to the interface at point $\mathbf{x}$), defining the force acting on the solid due to the fluid flow.

Result (32) represents an interesting starting point to relate the force computed using the immersed boundary approach, which has a numerical origin, to the physical force emerging in problem (2).

However, a practical estimate can be derived only under certain assumptions. First of all, we observe that if the solution of the original Navier-Stokes problem can be extended inside the solid domain, defining the pressure to be constantly zero, and the velocity in a way compatible with the rigid body boundary condition (in the considered benchmark, the velocity is also extended by

[4] In [3,4], a summation analogous to the one in (32) has been investigated in the context of the analysis of the Momentum Exchange Algorithm. It has been shown (detailed proof in [3]) that in virtue of the lattice symmetries it leads a first order approximation of the integral (34).

zero), pressure and velocity gradient jumps correspond to the values of those fields at the interface. Introducing a discretization error, from equation (32), we need the hypotheses

(i) the jumps in pressure and stress tensor obtained via forceBC algorithm approximate these fields at the interface (i.e. finer discretizations produces better approximation of the real jumps), and
(ii) the seminorm (30) can be estimated as $O(h)$.

In this case, according to (32) the forceBC algorithm yields a first order approximation of the boundary force (34). Moreover, since (30) is connected with the seminorm of the pressure difference, combining hypothesis (ii) with the definition (21) we conclude that a first order accurate pressure can be also achieved, but only in a weak sense.

Numerical experiments To test the theoretical estimates, we perform further numerical investigations on the benchmark described in figure 2b, comparing the forceBC algorithm with the BFL.

Cross-sectional views of velocity and pressure fields along the center of channel are shown in figure 4, which clearly reflects the fact that there is a good qualitative agreement with the BFL results. It should also be noted that the forceBC algorithm yields to smooth fields across the numerical interface. A decrease in the grid size results in a sharper interface, while the magnitudes of pressure and velocity inside the solid domain decrease, achieving a better approximation for the pressure jump (figure 4b). In figure 5 we measure the order of the difference between forceBC and BFL algorithms. Since the BFL conditions yield a second order in velocity field and first order in force, we verify whether the forceBC algorithm achieves a similar order of accuracy. In an affirmative case, the difference between the results should be of the same order of the error produced by the BFL. But, the order plot in figure 5a confirm that we can achieve only a first order accurate *pointwise* approximation of velocity [5] .

Figure 5a also demonstrates that the difference in pressure, evaluated according to the seminorm $\|\cdot\|_{1,h}$ is first order accurate and the force is approximated up to the first order in h as well, which is consistent with the predictions derived with the analysis. In addition, detailed results for drag and lift coefficients are also reported in table 1. They show satisfying agreement with the BFL-computed values, although the BFL results are closer to the literature reference [17].

[5] As remarked in section 3, the presented numerical results have been obtained using a bilinear interpolation for the interface velocity. Experiments with more accurate approximation routines did not bring considerable improvements.

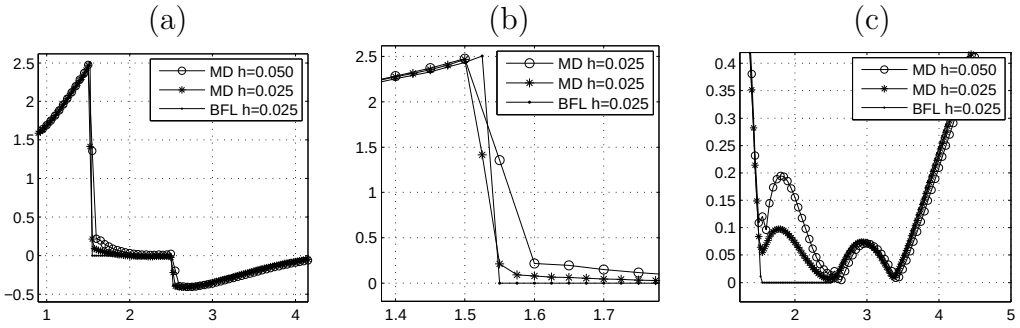

Figure 4. Cross section for $y = y_C$ of pressure and velocity squared near the disk. Qualitative comparison of forceBC and BFL algorithms on different grids ($\circ$: $h = 0.050$,$\star$: $h = 0.025$). Note that the fields in the solid domain are set equal to zero plotting the results of BFL rule. **(a)** Pressure. **(b)** Zoom of pressure closer to the interface, to show how the jump in pressure is smoothed out on a h-cell by the forceBC. **(c)** Squared velocity $u^2 + v^2$.

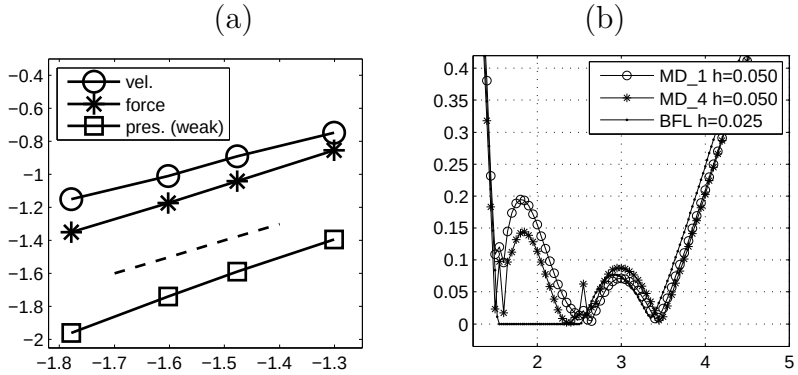

Figure 5. **(a)** Double logarithmic plot of $\left\|\mathbf{u}^{fBC} - \mathbf{u}^{BFL}\right\|$ ($\circ$), $\|F_S^{fBC} - F_S^{BFL}\|$ ($\star$) and $\left\|p^{fBC} - p^{BFL}\right\|_{1,h}$ ($\square$) versus grid size. Reference line of slope 1 is also drawn (dashed line). **(b)** Cross section of the velocity squared (as in figure 4b) using forceBC algorithm with $\xi = 1$ ($\circ$) and forceBC algorithm with $\xi = 4$, with modified friction (36) ($\star$). BFL results are shown for comparison. We remark that the forceBC algorithm is unstable without (36).

Table 1
Values of drag and lift coefficients compared with the one published in [17] and with the results obtained using the BFL boundary conditions (8)

	Standard BC (BFL)		LBM with forceBC		Reference Values
	82× 240	164×480	82× 240	164× 480	
Drag Coefficient	5.5790	5.581	5.8630	5.6930	5.5700 - 5.5900
Lift Coefficient	0.0116	0.0109	0.0114	0.0116	0.0104 - 0.0110

Stability discussion. Performing the analysis we found a strict relationship between the numerical results on the interface, the approximation of the fluid-solid interaction and the size of the fields computed in the solid domain (which is only a fictitious fluid). In fact, equation (32) relates all these quantities in a

single error estimate. This shows that the numerical error can quickly be amplified, yielding instability phenomena. Even if the presented benchmark (see the previous figures 4-5a) showed regular behaviors, a higher friction coefficient easily yields to stability problems. To better understand the mechanism triggering the instability we look at the force included in the LB algorithm (reporting equation (23))

$$g_i^{BC}(n, \mathbf{j}) = f_i^* c_s^{-2} \mathbf{c}_i \cdot \sum_{m=1}^{N_b(h)} -\hat{\xi}_h h^3 w_m^F(h, \mathbf{j})(\tilde{\mathbf{u}}_m - \mathbf{V}_m). \tag{35}$$

It can be seen as a penalty term, employed to achieve the desired velocity at the boundary. However, in case of excessive friction, the penalty can be exaggerate. Moreover, since the force only controls the sum over i of the previous equation, spurious oscillations of opposite signs in the velocity can be introduced and amplified in the following steps.

Although the algorithms as proposed in the original forms [1,5,7,18] show a common structure of the singular force, we observe that the presented investigation is partially independent on the explicit form of the friction force. In fact, the results for the boundary condition only requires the friction to vanish when $\mathbf{u} = \mathbf{u}_B$. Hence, it is theoretically possible to employ a modified friction coefficient, to be able to improve the global stability properties.

A detailed investigation in this direction can be an interesting topic of future research. To show how the idea could work, we test the choice

$$\mathbf{F}_m^{LB}(h, t_n, \mathbf{x_j}, \mathbf{u}) = \max\{\mathbf{F}_m^{LB,0}(h, t_n, \mathbf{x_j}, \mathbf{u}), K(h)\}, \tag{36}$$

(where $\mathbf{F}_m^{LB,0}$ is the force computed as in (13)). It is equivalent to employing an *adaptive friction* coefficient at each lattice node, which drastically bounds the friction force below a pre-assigned value. An optimal choice for the bound $K(h)$ might depend on the specific test case. In simulating a flow field reaching a stationary profile, it is necessary to control the size of the friction force only during the transient flow and a bound $K(h) \sim \sqrt{h}$ (i.e. order of magnitude higher than the lattice size h) might be a good choice. Results of this simple modification are shown in figure 5b. In more general situations, the values of $K(h)$ could also be investigated experimentally.

We observe that in practical applications, the reliability of an enhanced friction force can also achieve better qualitative approximation of the boundary condition.

5 Discussion and Conclusions

We have performed an asymptotic analysis of a force-boundary condition algorithm, viewed as an immersed boundary approach for the lattice Boltzmann method. Comparisons with a classical approach have been shown, based on a well known benchmark, to investigate the quality of this approach being an alternative. The investigation is useful from the practical and theoretical points of view.

In practice, the forceBC algorithm has the advantage of not requiring explicit boundary condition rules structurally different from the inner LBM, since it does not distinguish fluid and solid nodes, but only those nodes that are *close to* and *far from* the interface. This simplifies the implementation. On the other hand, depending on the size of fluid and solid domain, it might be more expensive than a classical fluid-solid approach, due to the need of running the LBM on a larger set of nodes.

However, the origin of instability represents a serious drawback. In [1] a statistical noise was included to enhance the stability. This cannot be done in the context of more general Dirichlet conditions. A different correction was proposed in [18], involving higher order interpolation and semi-implicit time discretizations. These features can be extremely costly (especially when accurate geometric characteristic of the interface are required) within the LBM, making the efficiency of the forceBC comparable (or even worse) than the classical approaches.

Concerning the accuracy, we have shown that the analysis predicts in general a first order accurate velocity field, confirmed by the numerical experiments. The presence of singular sources near the boundary does not allow to define properly a *pointwise* error for the pressure field. Thus, we have proposed a way to analyze the algorithm in a *weak sense*, i.e. considering a notion of precision weaker than the usual one. From the theoretical point of view, the introduction of a seminorm to evaluate the accuracy represents an interesting generalization, which can be useful when applying the analysis to problems involving irregularities or singular terms. In the particular case described here, this approach allowed us to conclude a weak first order accuracy for the pressure, also validated numerically. Additionally, the method has been used to derive an interesting result concerning the accuracy of the force computation.

The presented analysis helps to understand in which practical situations the forceBC algorithm can be used. We can conclude that the scheme produces satisfactory results when we are mainly interested in averaged behaviors (such as fluid-solid forces, look for example at table 1). The cases of flows with moderate speed past many particles (which represent a common benchmark

in the previously presented versions of the method [1,5,7,18]) also belong to this class of problems.

Since in those cases the algorithm can indeed improve the global efficiency, we have also proposed the idea of an *adaptive friction*, as an additional parameter to bound the numerical friction force in order to enhance the stability properties.

As a final remark, we observe that the idea of modifying the friction coefficient could be extended to enforce other types of boundary conditions. For example, the implementation of *slip* boundary conditions using a friction depending on the normal component of velocity and/or directed along the normal to the interface might represent an interesting topic for future research.

References

[1] P. Ahlrichs, B. Dünweg. Simulation of a single polymer chain in solution by combining Lattice Boltzmann and molecular dynamics, *J. Chem. Phys.*, **111**, 8225, 1999.

[2] M. Bouzidi, M. Firdaouss, P. Lallemand. Momentum transfer on a Boltzmann lattice fluid with boundaries, *Physics of Fluids*, **13**, 3452–3459, 2001.

[3] A. Caiazzo. Asymptotic Analysis of LBM for Fluid-Structure interaction problems. *PhD thesis*, Scuola Normale Superiore Pisa and Tecnische Universität Kaiserslautern (2007).

[4] A. Caiazzo, M. Junk. Boundary forces in lattice Boltzmann: analysis of Momentum Exchange algorithm, *Computer and Mathematics with Applications*, **55**(7): 1415–1423, **doi:**10.1016/j.camwa.2007.08.004, 2008.

[5] A. Chatterji, J. Horbach. Combining molecular dynamics with Lattice Boltzmann:A hybrid method for the simulation of (charged) colloidal systems, *J. Chem. Phys.*, **122**, 184903, 2005.

[6] I. Ginzburg, D. d'Humiéres. Multireflection boundary conditions for lattice Boltzmann models, *Phys. Rev. E*, 68, 066614, 2003.

[7] Z.-G. Feng, E. Michaelides. The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. *J. Comp. Phys.*, **195**, 602-628, 2004.

[8] X.He, L.-S.Luo. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, *Phys. Rev. E*, **56**:6811–6817, 1997.

[9] M. Junk, A. Klar, L.-S. Luo. Asymptotic analysis of the lattice Boltzmann equation, *J. Comp. Phys.*, **210(2)**, 76-704, 2005.

[10] M. Junk, Z. Yang. Analysis of lattice Boltzmann Boundary Conditions, *Proc. Appl. Math. Mech.*, **3**, 76-79, 2003.

[11] M. Junk, Z. Yang. Outflow Boundary Conditions for the lattice Boltzmann Method, *Proc. ICMMES 2006*, Hampton (Virginia), USA, July 2006.

[12] A. J. C. Ladd. Numerical simulations of particular suspensions via a discretized Boltzmann equation. Part 1, *J. Fluid Mech.*, **271**, 285-310, 1994.

[13] Z. Li, M.-C. Lai. The Immersed Interface Method for the Navier-Stokes Equations with Singular Forces, *J. Comp. Phys.*, **171**, 822-842, 2001.

[14] G.R.McNamara, G.Zanetti. Use of the Boltzmann equation to simulate lattice-gas Automata, *Phys. Rev. Lett.*, **61**:2332-2335, 1988.

[15] R. Mei, D. Yu, W. Shyy, L.-S. Luo. An accurate curved boundary treatment in the lattice Boltzmann method, *J. Comp. Phys.*,**155**, 307-330, 1999.

[16] R. Mei, D. Yu , W. Shyy, L.-S. Luo. Force evaluation in the lattice Boltzmann method involving curved geometry, *Phys. Rew. E*, **65**,041203, 2002.

[17] M. Shäfer, S. Turek. Benchmark Computations of Laminar Flow around a Cylinder, *Notes on Numerical Fluid Mechanics*, **52**, 547-566, Flow Simulation with High-Performance Computers II, Vieweg, co. F. Durst, E. Krause, R. Rannacher, 1996.

[18] C. Shu, N.Liu, Y.T. Chew. A novel immersed boundary velocity correction-lattice Boltzmann method and its application to simulate flow past a circular cylinder. *J. Comp. Phys.*, **226**, 1607-1622, 2007.

[19] Sauro Succi. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. *Oxford University Press*, 2001.