XL CILAMCE
IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
NOVEMBER 11-14, 2019
Praiamar Natal Hotel & Convention Natal, RN-BRAZIL

NUMERICAL INVESTIGATION OF UNIFORM FLOW PAST STATIONARY SPHERE USING IMMERSED BOUNDARY-LATTICE BOLTZMANN METHOD

Rodrigo S. Romanus
Alan Lugarini
Admilson T. Franco
rodrigo.romanus@gmail.com
alansouza@utfpr.edu.br
admilson@utfpr.edu.br
Research Center for Rheology and Non-Newtonian Fluids, Federal University of Technology – Paraná, Curitiba, PR, 81280-340, Brazil

Abstract. The Lattice-Boltzmann method (LBM) has been attracting lot of interest as a tool to solve hydrodynamic problems, mostly due its suitability for parallelization. However, the implementation of boundary conditions in LBM has some hardships, especially for flow domains containing immersed particles. An alternative to overcome this problem is the immersed boundary method (IBM), a numerical technique that uses a secondary Lagrangian mesh and imposes a boundary condition through a force field. This study presents the development of an IB-LBM based algorithm that simulates a free flow around a stationary sphere, which can also be described as a uniform jet stream that encounters a spherical obstacle. The fluid domain is described using a D3Q19 lattice arrangement and, a geodesic polyhedron based mesh delineates the spherical boundary. The flow evolution for fluid mesh is simulated using LBGK equation, whose stability is enhanced through regularization, a process which discards non-physical high order moments from LBM. In order to verify algorithm consistency, this document presents comparison with literature benchmark data of flow profiles and drag coefficients for different Reynolds numbers, showing good agreement between present work and available studies.

Keywords: LBM, IBM, Regularization, Drag

1 Introduction

Problems involving fluid-structure interaction are ubiquitous in Nature and therefore, object of great interest in fluid dynamics. This interdisciplinary subject is discussed in many fields such as, hemodynamics, Crosetto et al. [1], food processing, Oda et al. [2], and aerodynamics, Fairuz et al. [3], to name but a few. However, since this category of problems involves complicated solid-fluid interplay, analytic solutions are mostly non-existent and, in order to represent this phenomenon, one has to resort to experimental or numerical methods.

The purpose of this study is the development of a numerical model capable of predicting the behavior of flows with immersed solid boundaries. The algorithm validation was performed through analysis of a Newtonian flow past a stationary sphere, a case continuously studied over the years that has extensive literature benchmark data available, in which, the drag coefficient C_D and velocity profile for different Reynolds numbers is presented.

In 1850, Charles Gabriel Stokes derived an analytical solution for this flow by solving the Navier-Stokes equations (NSE) for a creeping flow, under this assumption, the drag coefficient in this situation is given by, Stokes [4]:

$$C_D = \frac{Re}{24}. \tag{1}$$

This solution entitled Stokes Law provides reliable results for $Re \leq 0.1$, Bird et al. [5]. As the Reynolds number increases, the inertial terms become significant in the momentum equation and no analytical solution is available. Therefore, several experimental and numerical studies were conducted to characterize flows around a spherical surface for $Re > 0.1$. As a result, many correlations for the drag coefficient as a function of Reynolds number were developed using different methods, for example, empirical fit of experimental data, Schiller and Naumann [6], by numerically solving the NSE, Concha and Almendra [7], among others. Photography of streamlines patterns for different Reynolds numbers can also be found in literature, those images can be registered via techniques such as aluminium dust method, Taneda [8].

The present work uses the combination of immersed boundary and lattice-Boltzmann methods, also known by the acronym 'IB-LBM', as phenomenon numerical description. Due its suitability for parallel computing, the lattice-Boltzmann method (LBM) has been attracting a lot of interest as a tool to solve hydrodynamic problems. Through discretization of domain in arrangements denoted lattices, the LBM numerically solves Boltzmann equation and, with the proper choice of collision operator Ω, can be used to solve the quasi-incompressible Navier-Stokes equation (NSE), Chapman et al. [9]. Despite its advantages, there are some hardships at implementation of boundary conditions in LBM, which require special attention. Cases were the boundary consists of a curved surface or a moving immersed solid can be quite challenging for LBM since lattices are most frequently structured in a Cartesian arrangement.

An alternative to overcome the issue of curved boundaries arises from the method developed in 1972 by Charles Peskin, immersed boundary method (IBM), which uses a secondary mesh for solid surface and imposes desired boundary condition by spreading force terms in fluid domain mesh, Peskin [10].

The collision model used is the regularized single relaxation-time BGK, which reduces stability issues from original BGK model caused by non-physical high-order moments present in any LBM simulation, Mattila et al. [11]. The aim of this work is to present a IB-LBM algorithm capable of producing an accurate simulation of a uniform flow around a spherical surface at different Reynolds number.

Some works combining IB-LBM techniques are available in literature, Dash et al. [12] uses a 3D IB-LBM scheme to simulate flows past stationary and moving spheres; Eshghinejadfard et al. [13], with a direct forcing immersed boundary combined with lattice-Boltzmann method, simulate sedimentation of two circular particles; Li et al. [14] models a 2D flow over cylinder using a regularized IB-LBM algorithm. The novelty of present study resides in the development of a three dimensional regularized IB-LBM numerical model that is validated for a flow over spherical surface.

2 Numerical method

This section provides a description of LBM and IBM, discussing their fundamental concepts, equations and the regularization process at BGK collision operator to increase LBM stability.

2.1 Lattice-Boltzmann method

The lattice-Boltzmann method was first introduced in 1988 by McNamara and Zanetti [15] and differs from traditional computational fluid dynamics methods because, instead of solving the equations of mass, momentum and energy on discrete nodes, elements or volumes, the fluid is replaced by particles that stream along given directions, often referred as lattice links, and collide at the lattice sites, Mohamad [16]. The lattice-Boltzmann equation is then divided in two major processes for each time step Δt.

The collision, a process that represents variations of particle populations due collisions and action of external forces Krüger et al. [17]. In this step, post-collision distribution populations $f_i|_{out}$ are calculated.

$$f_i|_{out}\left(\boldsymbol{x}, t\right) = f_i\left(\boldsymbol{x}, t\right) + \Omega_i + S_i. \tag{2}$$

And streaming, also referred as propagation, in which post-collision particle populations move to neighbor lattices; with their destinations being dictated by the velocity directions $\boldsymbol{e}_i$.

$$f_i\left(\boldsymbol{x} + \boldsymbol{e}_i\Delta t, t + \Delta t\right) = f_i|_{out}\left(\boldsymbol{x}, t\right). \tag{3}$$

In the above equations, $\boldsymbol{e}_i$ is the velocity direction, Ω_i the collision operator and S_i the source term associated with external forces acting on the fluid domain. In LBM, it is common practice to work with variables in lattice units and therefore, holding no physical dimension, Krüger et al. [17]. The source term can be deducted via different forms from Boltzmann equation, one path is through Hermite polynomial expansion, in which the source term is found as, Guo et al. [18]:

$$S_i = \left(1 - \frac{1}{2\tau}\right) w_i \left[\frac{\left(\boldsymbol{e}_i - \boldsymbol{u}\right)}{c_s^2} + \frac{\left(\boldsymbol{e}_i \cdot \boldsymbol{u}\right)}{c_s^4}\boldsymbol{e}_i\right] \cdot \boldsymbol{F}\left(\boldsymbol{x}, t\right). \tag{4}$$

In which, $\boldsymbol{F}\left(\boldsymbol{x}, t\right)$ is the macroscopic force density being exerted on the fluid, $\boldsymbol{u}\left(\boldsymbol{x}, t\right)$ the flow velocity field, c_s the speed of sound, equals $1/\sqrt{3}$ for usual LBM velocity sets, τ the relaxation time and w_i the weighting factor, whose value holds a lattice configuration dependence, Krüger et al. [17].

The particle distribution function f can be seen as a generalization of fluid density ρ that also takes the microscopic particle velocity into account. Thus, instead of a deterministic approach, where $\rho\left(\boldsymbol{x}, t\right)$ represents the density of particles in position x at time t, the fluid flow is described in a probabilistic fashion and $f_i\left(\boldsymbol{x}, t\right)$ can be seen as particles travelling with velocity $\boldsymbol{e}_i$, located in $\boldsymbol{x}$ at time t. This designates a mesoscopic description for the LBM, since it lies between macroscopic and microscopic scales, Mohamad [16]. The macroscopic moments of density and momentum are then obtained through summation of f_i and $\boldsymbol{e}_i$ moments.

$$\rho = \sum_i f_i. \tag{5a}$$

$$\rho\boldsymbol{u} = \sum_i f_i\boldsymbol{e}_i + \frac{\Delta t}{2}\boldsymbol{F}. \tag{5b}$$

The LBM development for a fluid flow occurs not only through discretization of physical space and time but also velocities domain, therefore, each discrete node has a set of discrete velocities $\{\boldsymbol{e}_i\}$ and their corresponding weights $\{w_i\}$. This configuration, often referred as lattice arrangement or velocity set, is named according to the number d of its spatial dimensions and discrete velocities q, thus, a $DdQq$ set is define in d dimensions and has q discrete velocities, Krüger et al. [17].

2.2 Regularized BGK collision operator

Boltzmann original collision operator considers all possible outcomes of two-particle collisions for any choice of intermolecular forces, Succi [19], and results in a burdensome integral over velocity space that can be highly non-linear. To avoid this kind of complication, Bhatnagar, Gross and Krook (1954) introduced a much simpler operator, Bhatnagar et al. [20]:

$$\Omega_{BGK}(f) = -\frac{1}{\tau}\left(f - f^{eq}\right). \tag{6}$$

Named after its inventors, the BGK collision operator uses a mean free-path treatment and expresses the fact that collisions tend to relax the distribution function towards equilibrium f^{eq}, Bhatnagar et al. [20]. For this expression, τ is a suitable average collision time, known as relaxation time.

The variable f^{eq} is the Maxwell-Boltzmann distribution function. Originally conceived for a continuous domain, this function can be projected in a discretized LBM by using orthogonal Hermite polynomials, Krüger et al. [17], the discrete equilibrium function for a second order expansion is:

$$f_i^{eq} = \rho w_i \left[1 + \frac{(\boldsymbol{e}_i \cdot \boldsymbol{u})}{c_s^2} - \frac{(\boldsymbol{u} \cdot \boldsymbol{u})}{c_s^2} + \frac{(\boldsymbol{e}_i \cdot \boldsymbol{u})^2}{c_s^4}\right]. \tag{7}$$

Being the non-equilibrium part of distribution function $f_i^{neq} = f_i - f_i^{eq}$ the standard lattice-BGK (LBGK) collision equation can be written as:

$$f_i|_{out}\left(\boldsymbol{x}, t\right) = f_i^{eq}\left(\boldsymbol{x}, t\right) + \left(1 - \frac{1}{\tau}\right) f_i^{neq}\left(\boldsymbol{x}, t\right) + S_i. \tag{8}$$

Using a second order Hermite polynomial expansion, the non-equilibrium populations can be written as, Mattila et al. [11]:

$$f_i^{neq} = \frac{w_i}{2c_s^4}\left[\boldsymbol{\Pi}^{neq} : \left(\boldsymbol{e}_i\boldsymbol{e}_i - c_s^2\boldsymbol{\delta}\right) - c_s^2\Delta t\left(\boldsymbol{F} \cdot \boldsymbol{e_i}\right)\right]. \tag{9}$$

With $\boldsymbol{\delta}$ being the Kronecker delta tensor and $\boldsymbol{\Pi}^{neq}$ the second order moment tensor for the non-equilibrium populations, given by:

$$\boldsymbol{\Pi}^{neq} = \sum_i f_i^{neq}\boldsymbol{e}_i\boldsymbol{e}_i. \tag{10}$$

Through a Chapman-Enskog expansion it is possible to return the NSE from Boltzmann equation and verify $\boldsymbol{\Pi}^{neq}$ as related to the viscous stress tensor, Regulski et al. [21].

This procedure that consists in projecting the non-equilibrium moments into a Hermitian representation is called regularization and increases stability of LBM, Mattila et al. [11], allowing simulations at higher Reynolds numbers to be performed. Montessori et al. [22], demonstrate a significant enhancement of numerical stability for lid-driven cavity flows using the regularized LBGK equation.

2.3 Immersed boundary method

The immersed boundary was originally developed to study flow patterns around heart valves, later evolving to a very useful tool for fluid-structure interaction problems. In the IBM, the fluid is represented on an Eulerian coordinate system and the structure on an Lagrangian, Peskin [10].

Velocity, forces and density described separately for each system are then related using the Dirac delta, a singular pulse function. This concept is extended to a numerical scheme through use of a secondary Lagrangian mesh L_h delineating the immersed boundary surface as illustrated in Fig 1.

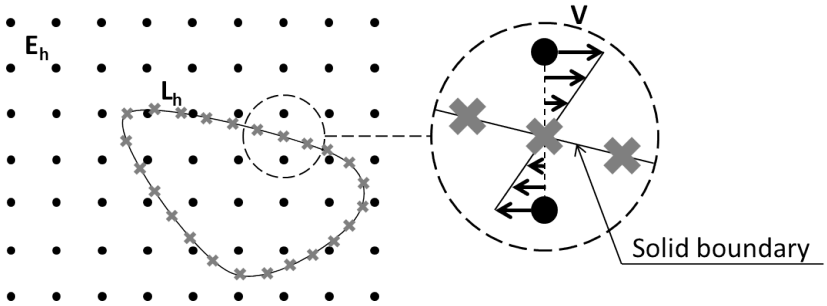

Figure 1. Exemplification of numerical IBM scheme, where velocities at Lagrangian nodes are obtained through interpolation of velocities from fluid nodes. E_h represents the Eulerian mesh

The velocity at each Lagrangian node located at $\boldsymbol{X}$ is given by interpolation of velocity field $\boldsymbol{u}$ from near Eulerian points $\boldsymbol{x}$. The equation which describes this process is, Peskin [10]:

$$\frac{d\boldsymbol{X}}{dt} = \sum_{\boldsymbol{x}\in E_h} \boldsymbol{u}\delta_h\left(\boldsymbol{x}-\boldsymbol{X}\right)\Delta x^3. \tag{11}$$

In Eq. (11), δ_h works as a weighting function and can be seen as a discrete equivalent of three dimensional Dirac delta. The representation of $\delta_h(\boldsymbol{x})$ for a Eulerian grid with Δx distance between nodes is:

$$\delta_h\left(\boldsymbol{x}\right) = \frac{1}{\Delta x^3}\phi\left(\frac{x_1}{\Delta x}\right)\phi\left(\frac{x_2}{\Delta x}\right)\phi\left(\frac{x_3}{\Delta x}\right). \tag{12}$$

In which, x_1, x_2 and x_3 are the components of a three dimensional vector $\boldsymbol{x}$ and ϕ is a one-dimensional discrete delta function which can be constructed through different paths, Yang et al. [23]. A commonly used form for ϕ is the four point piecewise, given by, Peskin [10]:

$$\phi(r) = \begin{cases} \frac{1}{8}\left(3-2|r|-\sqrt{1+4|r|-4|r|^2}\right) & ,\, 0\le|r|\le 1; \\ \frac{1}{8}\left(5-2|r|-\sqrt{-7+12|r|-4|r|^2}\right) & ,\, 1\le|r|\le 2; \\ 0 & ,\, |r|\ge 2. \end{cases} \tag{13}$$

This form satisfies restrictions that are imposed for the discrete Dirac function, being therefore, appropriate for IBM numerical schemes. In order to attain a specified boundary condition at the Lagrangian nodes, the IBM uses a process in which forces calculated at structure points are spread over the fluid domain as ilustrated in Fig. 2.

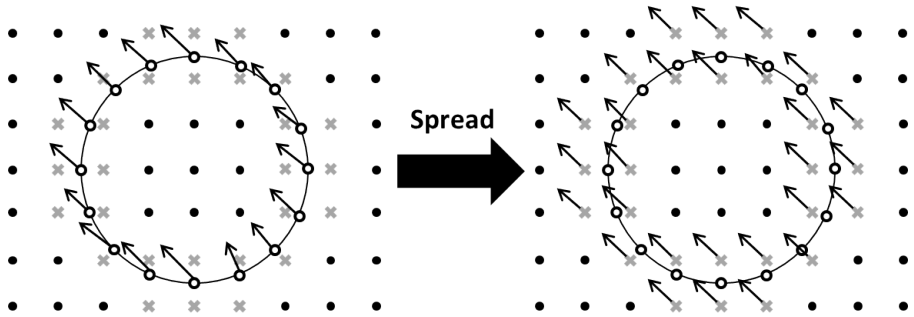

Figure 2. Illustration of spreading process of forces from IBM Lagrangian mesh to Eulerian nodes at fluid domain

That way, the boundary conditions are satisfied by means of a force field acting on the fluid that will shape the velocity field to satisfy the specified boundary conditions. The spreading process of Lagrangian forces can be written as:

$$\boldsymbol{F} = \sum_{\boldsymbol{X} \in L_h} \boldsymbol{f} \delta_h \left(\boldsymbol{x} - \boldsymbol{X} \right) \Delta A \Delta x. \tag{14}$$

In which, $\boldsymbol{f}$ is the force acting on Lagrangian nodes, ΔA is area element and $\boldsymbol{F}$ is the force on Eulerian domain. The implementation of those equations in an IB-LBM algorithm will be better detailed in the numerical modeling of present work.

3 Numerical Modeling

Figure 3 illustrates the flow domain considered, along with its boundary conditions. The fluid leaves inlet with a uniform velocity and collides with a spherical obstacle.

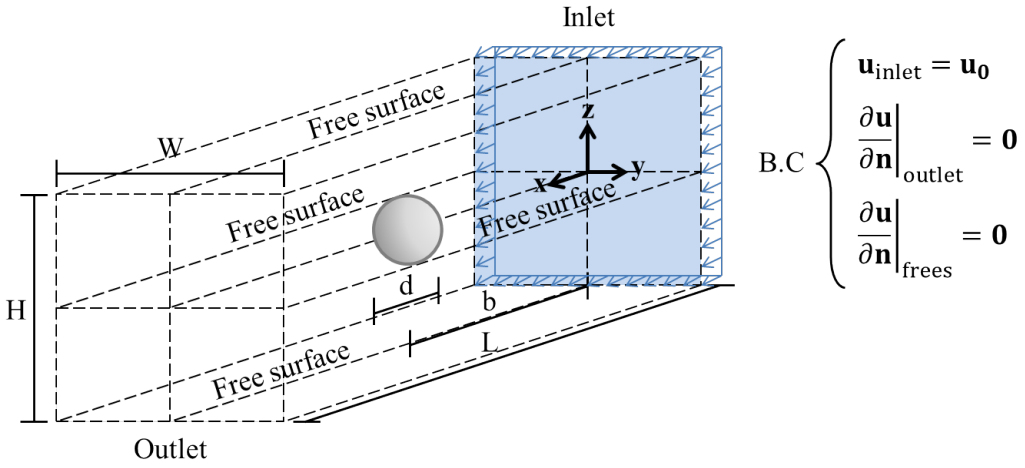

Figure 3. Mathematical representation of uniform flow past sphere, for each plane is assigned a boundary condition

A Dirichlet boundary condition with a uniform velocity normal to $-x$ plane is given at flow inlet; all other domain surfaces are taken as free surfaces, including the outlet. Those use a Neumann boundary condition in which the derivative of velocity vector towards normal plane direction has a given value, Yang [24], assumed as null for a free flow.

Being d the sphere diameter, the dimensions for the simulations were $W = H = 2.5d$, $L = 5d$ and $b = 1.5d$, the domain was discretized in a D3Q19 lattice arrangement, which is illustrated in Fig. 4.

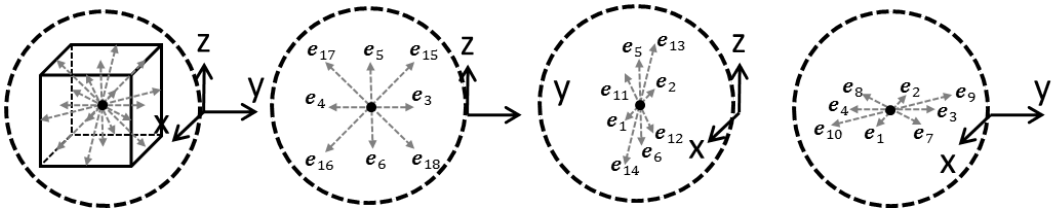

Figure 4. D3Q19 lattice arrangement, it has 19 velocities, being $\boldsymbol{e}_0$ the null direction and $c_s = 1/\sqrt{3}$

For this velocity set, $\{f_1, f_7, f_{10}, f_{12}, f_{13}\}$ are unknown populations at inlet during streaming, for a Dirichlet boundary condition, those are given by, Krüger et al. [17]:

$$f_{-i}\left(\boldsymbol{x}_{in}, t + \Delta t\right) = f_i|_{out}\left(\boldsymbol{x}_{in}, t\right) - 2w_i\rho_{in}\frac{\boldsymbol{e}_i \cdot \boldsymbol{u}_0}{c_s^2}. \tag{15}$$

With $-i$ indicating the opposite velocity direction. For the Neumann boundary condition implementation, ghost nodes are used, as illustrated in Fig. 5.

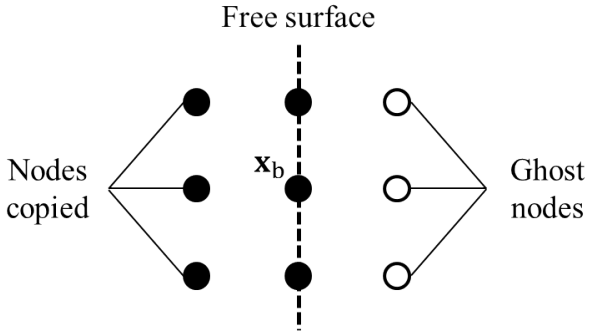

Figure 5. Illustration of ghost nodes used for application of Neumann boundary condition

By copying the nodes from boundary neighborhood the zero normal derivative is assured during streaming, which is then performed normally using the copied nodes, Junk and Yang [25].

The non-slip condition at sphere surface is attained through IBM using a icosahedron based mesh. Amongst regular polyhedrons, this geometry presents the highest number of sides and is favorable for generating a spherical mesh with almost evenly spaced nodes through hierarchical triangular mesh (HTM) method, Szalay et al. [26]. This technique is illustrated in Fig. 6.

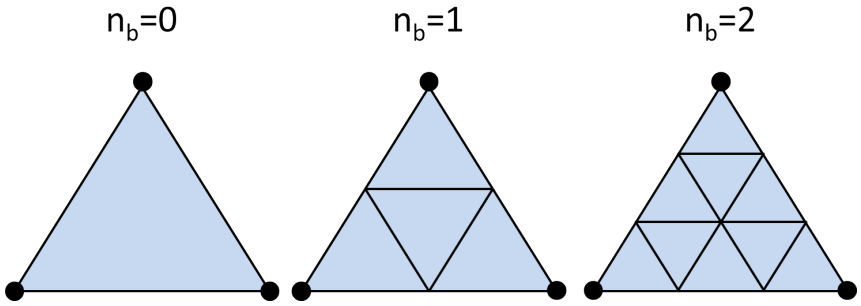

Figure 6. Representation of HTM method, a triangular plane is subdivided in more triangular faces for each refinement degree n_b

This subdivision procedure is applied to each face of original icosahedron and the nodes are projected into a circumscribed sphere resulting in a geodesic dome, as shown in Fig. 7.

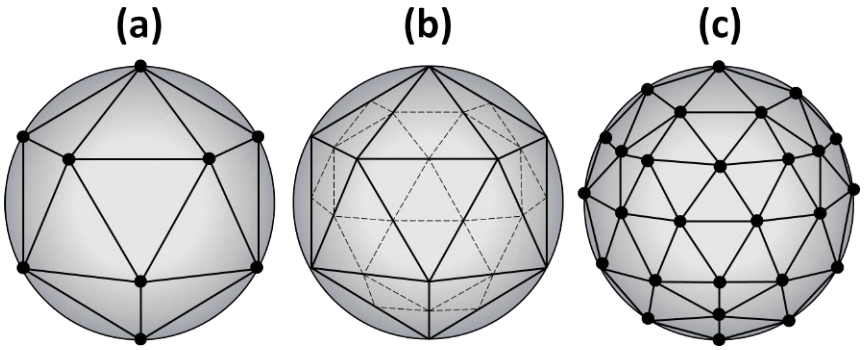

Figure 7. Schematic of meshing process for sphere using a polyhedron (a) regular icosahedron with vertexes at sphere surface (b) division of each face in 4 equal triangles (c) the new vertexes are normalized and extended to sphere surface creating a 80 faces circumscribed polyhedron

The amount of Lagrangian nodes will be number of vertexes V_{poly} for a convex polyhedron with F_{poly} faces and E_{poly} edges. For a icosahedron with n_b divisions:

$$F_{poly} = 20\left(1 + n_b + n_b^2\right); \tag{16a}$$

$$E_{poly} = 30\left(1 + n_b + n_b^2\right); \tag{16b}$$

$$V_{poly} = 2 + E_{poly} - F_{poly}. \tag{16c}$$

That way, the element ΔA from Eq. (14) is assumed as:

$$\Delta A = \frac{\pi d^2}{V_{poly}}. \tag{17}$$

To satisfy the no-slip condition within a convergence limit, an implicit calculation of the force density scheme, with a single Lagrangian velocity correction term, is prescribed, Dash et al. [12]. The boundary velocity is written in terms of its interpolation $\boldsymbol{u}_b = \boldsymbol{u}_{int} + \delta\boldsymbol{u}$ and an iterative process for the force term is given by:

$$\boldsymbol{f}^{(n)} = \boldsymbol{f}^{(n-1)} + \frac{2\rho\delta\boldsymbol{u}}{\Delta t}. \tag{18}$$

The forces are spread into Eulerian domain, the velocity field is recalculated and a iterative process follows until a error criterion for velocity $|\delta\boldsymbol{u}| \leq L$ is attained. Source terms in collision equation S_i are then calculated from spread forces $\boldsymbol{F}$ and evolution of LBM proceeds.

Therefore, the complete IB-LBM simulation starts with a initialization, in which populations are set equal to their equilibrium parts for a initial velocity and density, then a implicit force scheme is applied and is followed by collision and streaming. The numerical algorithm is summarized in the flowchart in Fig. (8). The simulation is performed until drag coefficient variation between 1000 time steps is negligible and flow regime is assumed as steady.

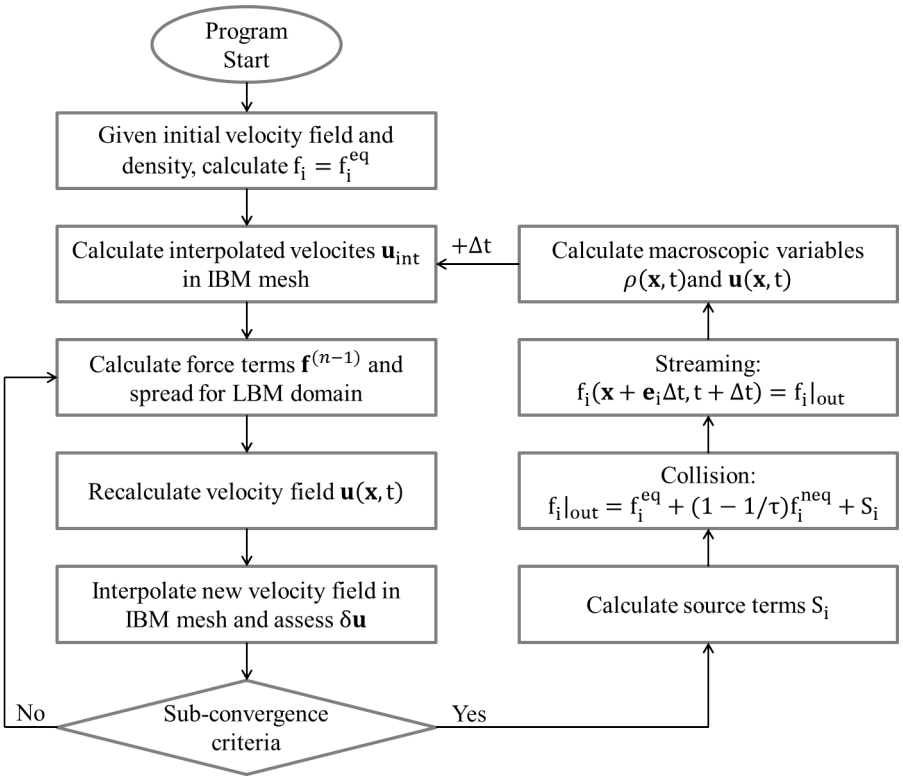

Figure 8. Flowchart of immersed boundary-lattice Boltzmann method algorithm

As mentioned, it is possible to return the NSE from Boltzmann equation. That way, it is possible to relate the relaxation time τ, with the kinematic viscosity ν as, Krüger et al. [17]:

$$\nu = \left(\tau - \frac{1}{2}\right) c_s^2. \tag{19}$$

Therefore, taking the reference length as the sphere diameter d for considered phenomenon, the Reynolds number for each simulation will be given by:

$$Re = \frac{u_0 d}{\nu}. \tag{20}$$

4 Results and discussion

Simulations from developed algorithm were performed with different Reynolds numbers which, for comparison of streamlines patterns, were chosen to coincide with experimental work from Taneda [8]. Also, the drag coefficient C_D for each simulation was given by:

$$C_D = \frac{2F_D}{\rho_0 u_o^2 A_p}. \tag{21}$$

Being A_p, the projected area from spherical surface $\pi(d/2)^2$ and F_D, the total drag force over sphere, calculated through momentum balance of a control volume containing the immersed boundary.

4.1 Mesh refinement

Mesh refinement is determined by the number of lattices occupied by IB-mesh diameter. Simulations with M1: $d = 10$, M2: $d = 20$ and M3: $d = 40$, were performed for $Re = 118$; the distance between IB nodes was fixed at ~ 1 lattice unit, and Table 1 presents obtained C_D results.

Table 1. Coefficients calculated for each mesh refinement compared with literature correlations.

M1	M2	M3	Schiller and Naumann [6]	Concha and Almendra [7]
1.117	1.090	1.067	1.012	0.942

Schiller and Naumann [6] results were calculated from their correlation developed through experimental data fit for laminar flows:

$$C_D = \begin{cases} \frac{24\left(1+0.15Re^{0.687}\right)}{Re} & \Rightarrow Re \leq 1000; \\ 0.44 & \Rightarrow Re > 1000. \end{cases} \tag{22}$$

Whilst Concha and Almendra [7] correlation was obtained combining boundary-layer theory and experimental data from pressure distribution and boundary-layer thickness, and is written as:

$$C_D = 0.28\left(1 + \frac{9.06}{Re^{1/2}}\right)^2. \tag{23}$$

The improvement of results through mesh refinement can indicate that the solution converges to an asymptotic value, being very close from expected results for M2 and M3. Fig. 9 presents streamlines and pathlines obtained for each simulation.

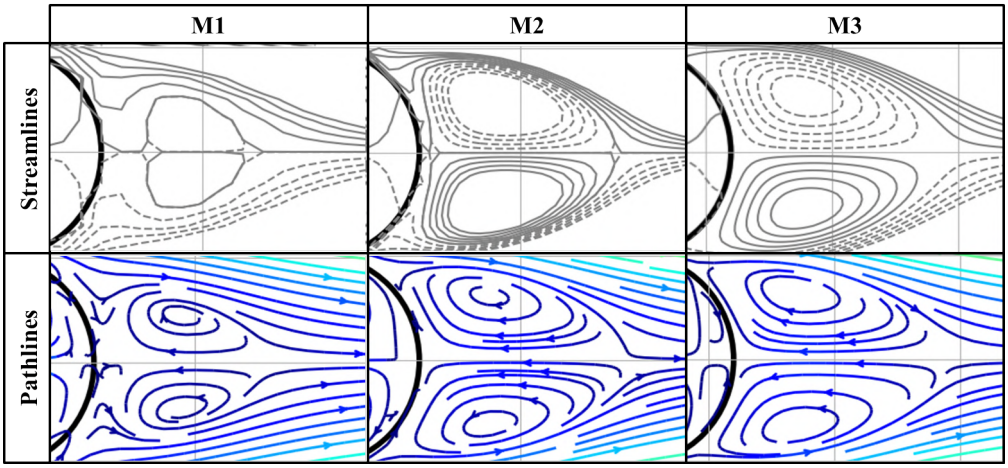

Figure 9. Streamlines and pathlines for $M1$, $M2$ and $M3$ simulations

It can be seen that the flow pattern changes severely from $M1$ to $M3$, indicating that $M1$ is to coarse too correctly represent the phenomenon, there is still a small change from $M2$ to $M3$ and, to assure accurate phenomenon representation, other simulations were performed using $M3$.

4.2 Streamlines and drag coefficient

The streamlines in this work are plotted at plane $z = 0$ using the velocity components from x and y directions. For the considered two dimensional plane, a stream function Ψ can be defined as having its

derivatives matching velocity components u and v.

$$v = -\frac{\partial \psi}{\partial x} \quad ; \quad u = \frac{\partial \psi}{\partial y}$$

And being the vorticity ω defined as the curl of velocity vector, for the flow at plane xy with velocity components u and v:

$$\omega = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = \frac{\partial^2 \Psi}{\partial y^2} - \frac{\partial^2 \Psi}{\partial x^2}. \tag{24}$$

Using a finite difference scheme to numerically solve this Poisson equation for a given node (i, j) of a discretized domain, vorticity $\omega_{(i,j)}$ can be written as a function of $\Psi_{(i,j)}$ and its neighbor values:

$$\frac{\Psi_{(i+1,j)} - 4\Psi_{(i,j)} + \Psi_{(i-1,j)}}{\Delta x} + \frac{\Psi_{(i,j+1)} - 4\Psi_{(i,j)} + \Psi_{(i,j-1)}}{\Delta y} = -\omega_{(i,j)}. \tag{25}$$

The 2-D vorticity is calculated using components x and y from velocity field and a central difference derivative, which gives:

$$\omega_{(i,j)} = \frac{u_{(i,j+1)} - u_{(i,j-1)}}{\Delta y} - \frac{v_{(i+1,j)} - v_{(i-1,j)}}{\Delta x}. \tag{26}$$

Using a iterative process, it is possible to find values for the stream function and draw its contour lines, Fig. 10 shows calculated streamlines and path lines from present work sided with photos from Taneda [8] experimental study for $Re = 9.15$ and $Re = 25.5$.

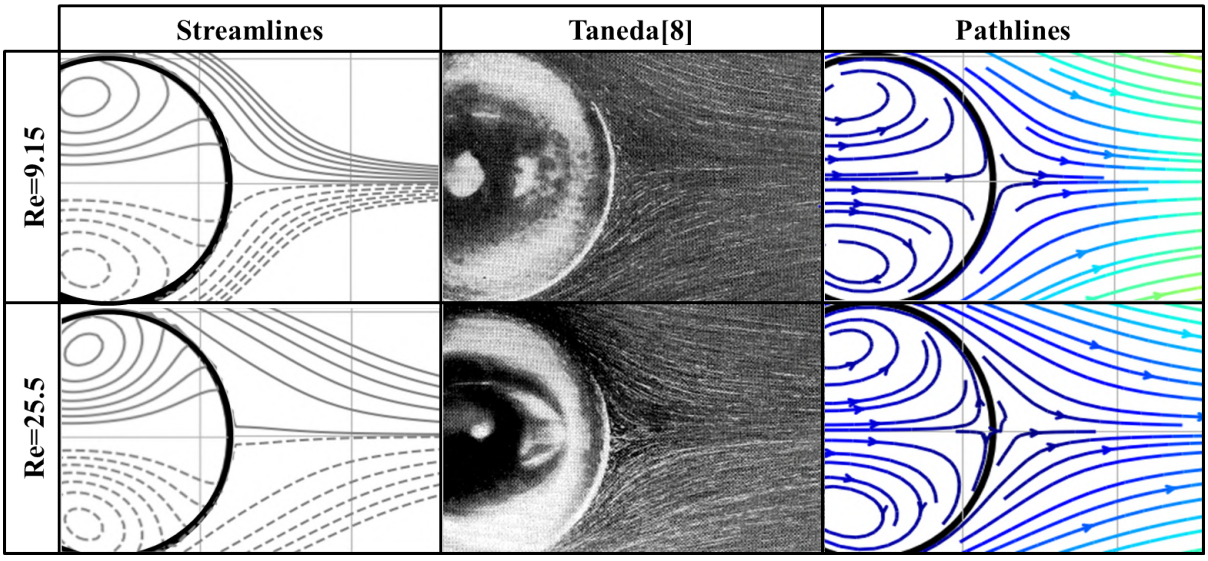

Figure 10. Comparison of flow patterns with Taneda [8], reproduced by permission of (1956) The Physical Society of Japan, for $Re = 9.15$ and $Re = 25.5$

A coherent flow behavior can be seen from both streamlines and pathlines in comparison to experimental study photos. It is interesting to notice that, since IBM uses a force field instead of a physical

boundary, there is still flow inside sphere. At higher Reynolds numbers, there is formation of recirculation zones at wake behind the sphere, this phenomena can be seen in Fig. 11.

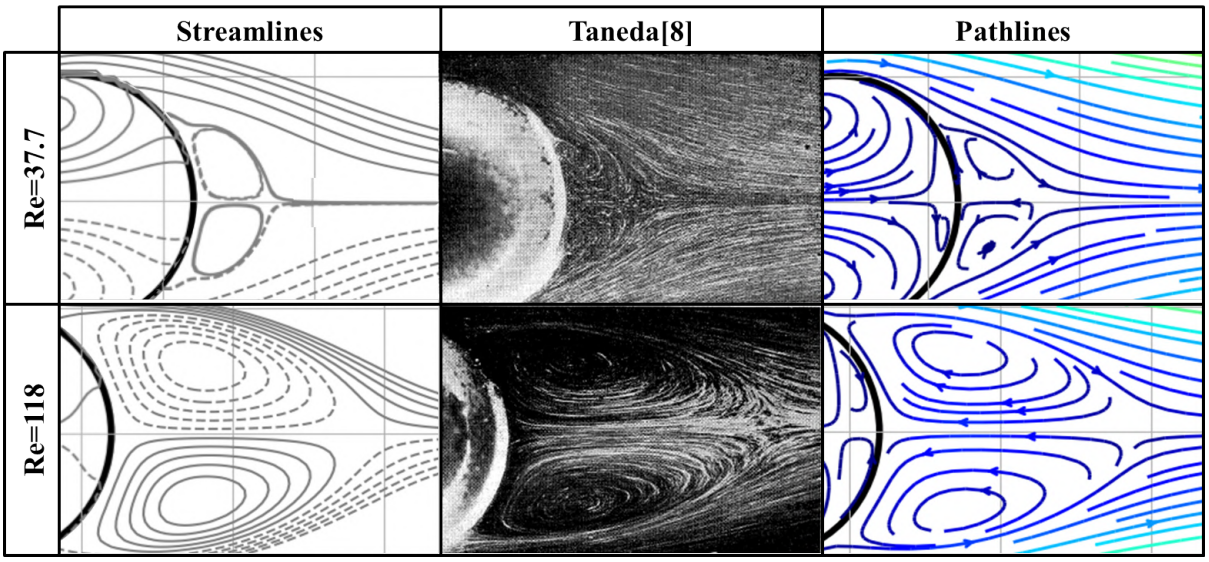

Figure 11. Comparison of flow patterns with Taneda [8], , reproduced by permission of (1956) The Physical Society of Japan, for $Re = 37.7$ and $Re = 118$

In qualitative terms, through comparison of vortex shape with photos from experimental work, the simulation plots suggest a consistent model for representation of Newtonian flow at different Reynolds numbers. Drag coefficient values for performed simulations are presented in Table 2.

Table 2. Drag coefficients obtained for different Reynolds numbers

Reynolds number	9.15	25.5	37.7	118
Present work	4.973	2.446	1.889	1.067
Schiller and Naumann [6]	4.423	2.248	1.792	1.012
Concha and Almendra [7]	4.469	2.186	1.716	0.942

It can be seen a good agreement between values from correlations and obtained from simulations, especially since the drag coefficient is most of the times presented in logarithmic scale. This indicates a consistent numerical description; however, the domain size can also affect simulation results.

In order to verify this influence, four simulations with a $d = 10$ IBM mesh were performed with different domain dimensions, as shown in Table 3.

Table 3. Simulation parameters given in lattice units

Dimension	W	H	L	h
M4	50	50	100	30
M5	100	100	100	30
M6	50	50	200	30
M7	50	50	200	100

For those parameters, simulations were performed at $Re = 9.15$ and the obtained drag coefficient values are presented in Table 4.

Table 4. Coefficients calculated for $M4$, $M5$, $M6$ and $M7$ compared with literature correlations

M4	M5	M6	M7	Schiller and Naumann [6]	Concha and Almendra [7]
5.526	6.070	5.522	5.163	4.423	4.469

From $M5$, the inlet Dirichlet boundary condition too close from spherical obstacle is insufficient for flow development and, by changing only domain width, the region before sphere in which the flow is not well represented increases, causing values obtained for drag coefficient to diverge even more from correlations. This influence of h becomes evident with $M7$, for which the drag coefficient results were the closest from correlations. Repeating one of the previous simulations, with sphere diameter $d = 20$ and $Re = 9.15$, but with sphere placed at domain center, the value obtained for C_D was 4.835.

5 Conclusion

A proposed IB-LBM numerical model was successfully implemented for simulation of flow over spherical surface, accurately predicting drag coefficient and flow patterns for different Reynolds numbers, being therefore, this work main achievement, the development of a highly parallelizable numerical model that can be used for aerodynamics studies involving drag forces. A next step for algorithm improvement would be the implementation of a moving boundary for the immersed surface and validation for different geometries, since the mathematical formulation for this model does not restrict itself only for a sphere simulation.

Acknowledgements

The Physical Society of Japan for allowing use of their published material from research paper *Experimental Investigation of the Wake behind a Sphere at Low Reynolds Numbers (1959)*

References

[1] Crosetto, P., Reymond, P., Deparis, S., Kontaxakis, D., Stergiopulos, N., & Quarteroni, A., 2011. Fluid–structure interaction simulation of aortic blood flow. *Computers & Fluids*, vol. 43, n. 1, pp. 46–57.

[2] Oda, A., Moatamedi, M., & Itoh, S., 2007. Study and analysis of the fluid-structure interaction between apple and underwater shock wave. In *ASME 2007 Pressure Vessels and Piping Conference*, pp. 125–129. American Society of Mechanical Engineers.

[3] Fairuz, Z., Abdullah, M., Yusoff, H., & Abdullah, M., 2013. Fluid structure interaction of unsteady aerodynamics of flapping wing at low reynolds number. *Engineering Applications of Computational Fluid Mechanics*, vol. 7, n. 1, pp. 144–158.

[4] Stokes, G., 1850. On the effect of internal friction of fluids on the motion of pendulums. *Trans. Camb. phi1. S0c*, vol. 9, n. 8, pp. 106.

[5] Bird, R. B., Armstrong, R. C., & Hassager, O., 1987. Dynamics of polymeric liquids. vol. 1: Fluid mechanics.

[6] Schiller, L. & Naumann, A., 1933. Uber die grundlegenden berechnungen bei der schwerkraftaufbereitung. zeitung des vereins deutscher ingenieure.

[7] Concha, F. & Almendra, E., 1979. Settling velocities of particulate systems, 1. settling velocities of individual spherical particles. *International Journal of Mineral Processing*, vol. 5, n. 4, pp. 349–367.

[8] Taneda, S., 1956. Experimental investigation of the wake behind a sphere at low reynolds numbers. *Journal of the Physical Society of Japan*, vol. 11, n. 10, pp. 1104–1108.

[9] Chapman, S., Cowling, T. G., & Burnett, D., 1990. *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press.

[10] Peskin, C. S., 2002. The immersed boundary method. *Acta numerica*, vol. 11, pp. 479–517.

[11] Mattila, K. K., Philippi, P. C., & Hegele Jr, L. A., 2017. High-order regularization in lattice-boltzmann equations. *Physics of Fluids*, vol. 29, n. 4, pp. 046103.

[12] Dash, S., Lee, T., Lim, T., & Huang, H., 2014. A flexible forcing three dimension ib–lbm scheme for flow past stationary and moving spheres. *Computers & Fluids*, vol. 95, pp. 159–170.

[13] Eshghinejadfard, A., Abdelsamie, A., Janiga, G., & Thévenin, D., 2016. Direct-forcing immersed boundary lattice boltzmann simulation of particle/fluid interactions for spherical and non-spherical particles. *Particuology*, vol. 25, pp. 93–103.

[14] Li, Z., Cao, W., & Le Touzé, D., 2019. On the coupling of a direct-forcing immersed boundary method and the regularized lattice boltzmann method for fluid-structure interaction. *Computers & Fluids*, vol. 190, pp. 470–484.

[15] McNamara, G. R. & Zanetti, G., 1988. Use of the boltzmann equation to simulate lattice-gas automata. *Physical review letters*, vol. 61, n. 20, pp. 2332.

[16] Mohamad, A., 2011. *Lattice Boltzmann Method*, volume 70. Springer.

[17] Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., & Viggen, E. M., 2017. The lattice boltzmann method. *Springer International Publishing*, vol. 10, pp. 978–3.

[18] Guo, Z., Zheng, C., & Shi, B., 2002. Discrete lattice effects on the forcing term in the lattice boltzmann method. *Physical Review E*, vol. 65, n. 4, pp. 046308.

[19] Succi, S., 2001. *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford university press.

[20] Bhatnagar, P. L., Gross, E. P., & Krook, M., 1954. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical review*, vol. 94, n. 3, pp. 511.

[21] Regulski, W., Leonardi, C. R., & Szumbarski, J., 2016. On the spatial convergence and transient behaviour of lattice boltzmann methods for modelling fluids with yield stress. *arXiv preprint arXiv:1610.01388*.

[22] Montessori, A., Falcucci, G., Prestininzi, P., La Rocca, M., & Succi, S., 2014. Regularized lattice bhatnagar-gross-krook model for two-and three-dimensional cavity flow simulations. *Physical Review E*, vol. 89, n. 5, pp. 053317.

[23] Yang, X., Zhang, X., Li, Z., & He, G.-W., 2009. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics*, vol. 228, n. 20, pp. 7821–7836.

[24] Yang, Z., 2007. *Analysis of lattice Boltzmann boundary conditions*. PhD thesis.

[25] Junk, M. & Yang, Z., 2008. Outflow boundary conditions for the lattice boltzmann method. *Progress in Computational Fluid Dynamics, an International Journal*, vol. 8, n. 1-4, pp. 38–48.

[26] Szalay, A. S., Gray, J., Fekete, G., Kunszt, P. Z., Kukol, P., & Thakar, A., 2007. Indexing the sphere with the hierarchical triangular mesh. *arXiv preprint cs/0701164*.