Proceedings of the Canadian Society for Mechanical Engineering International Congress
32nd Annual Conference of the Computational Fluid Dynamics Society of Canada
Canadian Society of Rheology Symposium
CSME/CFDSC/CSR 2025
May 25–28, 2025, Montréal, Québec, Canada

A sharp interface Immersed Boundary approach for Simplified and Highly Stable Lattice Boltzmann Method

Md Sujaat Ali[1]∗, Jean-Yves Trépanier[1], Sébastien Leclaire[1]

[1]Department of Mechanical Engineering, Polytechnique Montréal, Montréal, Canada
∗md-sujaat.ali@polymtl.ca

March 23, 2025

***Abstract*—**The Lattice Boltzmann Method is a mesoscopic method and has been used for some time as a computational fluid dynamics solver. The recently developed Simplified and Highly Stable Lattice Boltzmann Method (SHSLBM) simplifies the boundary condition implementation to a great extent as it can be implemented using the macroscopic variables instead of distribution functions in traditional Lattice Boltzmann Method. Moreover, it can simulate incompressible flows without evolution of the distribution function, making it less computationally expensive. However, it is still a challenge to implement it for flow around geometry not aligned with the mesh. Immersed boundary Method is used for the flow around geometries without body fitted mesh therefore making it as an ideal choice to extend SHSLBM to simulate flow around geometries not aligned with the mesh. In the present work, for the first time a combination of a sharp interface Immersed Boundary approach is implemented for the boundary treatment of the non-aligned geometry for Simplified and Highly Stable Lattice Boltzmann Method. The implementation is done using an extrapolation of the macroscopic flow variables to the ghost (solid) nodes using a Least-Square technique, for both Neumann and Dirichlet boundary type. The present work is verified for a Poiseuille flow test case and a code-to-code verification is done for flow around square cylinder in a channel, with Immersed Boundary, and a comparison is made with flow around square cylinder without Immersed Boundary.

***Keywords-component*—Lattice Boltzmann Method; Viscous flow; Immersed Boundary Method;**

I. I. INTRODUCTION

Computational Fluid Dynamics (CFD) have been in use for many decades, trying to mimic the various physical phenomenon with the aim to provide a more insightful understanding of the fluid flows. The traditional methods include Finite Volume Method, Finite Element Methods, Finite Difference Method etc. Lattice Boltzmann Method (LBM) is a mesoscopic solver and it provides an alternative to the traditional approaches mentioned above.

For a typical lattice boltzmann simulation, it is required to store the particle distribution functions for each lattice points in order to complete the streaming and the collision steps. There are certain problems which are present in the traditional Lattice Boltzmann (LB) models. One of the most common drawbacks of LB model compared to the continuum N-S solvers is the high virtual memory cost, as the calculation of distribution functions are required at each cell for every time step. Another major drawback is that boundary conditions are not implemented directly using the macroscopic variables instead it is done using distribution functions and their implementation in LBM is much more complex compared to tradition continuum N-S solvers.

Keeping the above mentioned drawbacks in mind, Chen et al. [1] first introduced the Simplified LBM (SLBM) and used the fractional step method to convert the calculation of the macroscopic variables in a predictor and corrector step using the non-equilibrium part of the distribution functions. Chen et al. [2] extended the idea to 3-D with heat transfer. The same group of authors in Ref. [3] modified the scheme to be based only on the calculation of the equilibrium distribution function in both predictor and corrector step, which they called Simplified and Highly Stable LBM (SHSLBM). SHSLBM is relatively more stable than traditional LB methods. It can simulate flow with high Reynolds number (10^4) and with relaxation time very close to the numerical limit [3].

Immersed Boundary Method (IBM) has been in use for some time and was introduced by Peskin [4] to handle elastic boundaries for simulating blood flow in the heart. IB method

can be distinguished from the body fitted mesh as the mesh is non-conforming to the geometry. A typical implementation of the immersed boundary method is to use the cartesian mesh, to simulate flow around a geometry, irrespective of the shape of the geometry. In other words, the geometry is immersed in the mesh.

As per Rajat Mittal et al. [5] the IB methods can be classified primarily into Continuous Forcing(CF) and Discrete Forcing (DF) approaches. DF methods are effective in simulating fluid flow around solid objects. Within the DF framework, the "sharp interface" or Direct Boundary Condition (DBC) enforcement is particularly suitable for solid boundaries to ensure local accuracy for boundary layers [5]. The DBC approach can be executed through various means, including Ghost Cell and Cut-Cell methodologies. Ghost cells refer to cells located within the solid region that have at least one neighboring cell within the fluid. The Cut-Cell method involves utilizing a control volume entirely within the fluid domain but is an irregular polygon near the boundary [6]. These polygons arise from modified cells containing the boundary.

For the implementation of the immersed boundary methods, the first step is the cell classification. This can be achieved using a variety of algorithms including ray tracing [7], [8], point in a polygon [9], [10] etc. Assuming external flow cases are solved, the cells which are inside the geometry are called solid cells and are not solved. Cells outside the geometry are fluid cells and Ghost cells are the solid cells for which at least one neighbour is a fluid cell. For the fluid cells, the fluid solver is used and for the ghost cells, the flow variables are reconstructed using an extrapolation function which includes information from near by fluid nodes and the boundary. The interpolation methods include, least-square [11]–[14] radial basis function [15]–[17], bilinear interpolation [9], [10] etc.

The present work introduces the implementation of sharp interface immersed boundary approach in SHSLBM. The Sharp interface IB methods can provide local accuracy, especially for boundary layers, which will be difficult to achieve using other approaches [5]. Also, as the boundary conditions are implemented using the macroscopic variables in SHSLBM, instead of using distribution function, makes the sharp interface IB method using ghost cells an ideal choice. Moreover, it has already been implemented and tested for different cases using traditional fluid flow solvers like Finite Volume method [18]. In this paper, flow around square cylinder and immersed Poiseuille flow are presented. The methodology remains the same and can be applied in the same way to various other cases like flow around an airfoil or circular cylinders.

This paper is divided into four sections. Section II defines numerical model which includes the formulations for the traditional Lattice Boltzmann Method, SHSLBM, Immersed Boundary Method and the extrapolation function. Section III includes the results and discussions for Poiseuille flow and flow around square cylinders and comparison of the results with the analytical and previously cited results respectively. Section IV deals with the conclusion based on the results and discussions.

II. Lattice Boltzmann method

The LBM for incompressible fluid flows with BGK approximation looks like [19]:

$$f_\alpha(\mathbf{r}+\mathbf{e}_\alpha\delta_t, t+\delta_t) = f_\alpha(\mathbf{r},t) + \frac{f_\alpha^{eq}(\mathbf{r},t) - f_\alpha(\mathbf{r},t)}{\tau} \quad (1)$$

where $\alpha = 0, 1, ...N-1$, f_α is the density distribution function and f_α^{eq} is the equilibrium state of f_α, τ is the relaxation parameter, δ_t is the streaming time step, $N-1$ is the total number of the discrete particles for f_α, $\mathbf{e}_\alpha$ is the velocity vector which depend on the selection of the lattice discretization model, t is the time and $\mathbf{r}$ is the position vector. The macroscopic properties can be obtained by:

$$\rho = \sum_{\alpha=0}^{N-1} f_\alpha, \rho\mathbf{u} = \sum_{\alpha=0}^{N-1} f_\alpha \mathbf{e}_\alpha \quad (2)$$

The equilibrium density distribution function is given by:

$$f_\alpha^{eq}(\mathbf{r},t) = \rho\omega_\alpha[1 + \frac{\mathbf{e}_\alpha\cdot\mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha\cdot\mathbf{u})^2 - (c_s \mid \mathbf{u} \mid)^2}{2c_s^4}] \quad (3)$$

The kinematic viscosity and the relaxation parameter are related by:

$$\nu = (\tau - 1/2)c_s^2\delta_t \quad (4)$$

In the lattice Boltzmann method (LBM), the equilibrium distribution functions also satisfy the conservation of mass and momentum at any physical location. Thus, we have

$$\rho = \sum_{\alpha=0}^{N-1} f_\alpha^{eq}, \rho\mathbf{u} = \sum_{\alpha=0}^{N-1} f_\alpha^{eq}\mathbf{e}_\alpha \quad (5)$$

All the cases simulated in this work are two dimensional and a D2Q9 model is used for which:

$$\omega_\alpha = \begin{cases} 4/9, & \alpha = 0 \\ 1/9, & \alpha = 1 \sim 4, \\ 1/36, & \alpha = 5 \sim 8 \end{cases} \quad (6)$$

$$\mathbf{e}_\alpha = c\begin{cases} (0,0), & \alpha = 0 \\ (\cos[(\alpha-1)\pi/2], \sin[(\alpha-1)\pi/2]), & \alpha = 1 \sim 4 \\ (\cos[(2\alpha-9)\pi/4], \sin[(2\alpha-9)\pi/4]), & \alpha = 5 \sim 8 \end{cases} \quad (7)$$

A. Simplified and Highly Stable LBM

The C-E analysis recovers the following macroscopic equations from the standard LBE [20]:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\left(\sum_\alpha \mathbf{e}_\alpha f_\alpha^{eq}\right) = 0; \quad (8)$$

$$\frac{\partial\rho\mathbf{u}}{\partial t} + \nabla\cdot\sum_\alpha(\mathbf{e}_\alpha)_\beta(\mathbf{e}_\alpha)_\gamma\left[f_\alpha^{eq} + (1-\frac{1}{2\tau})f_\alpha^{neq}\right] = 0; \quad (9)$$

The f^{neq}, the non-equilibrium distribution function is related to the equilibrium distribution function by:

$$f_\alpha^{neq} = -\tau\delta_t \left(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\right) f_\alpha^{eq} \tag{10}$$

using the fractional step technique, the governing equations above can be modified to a predictor-corrector model as:
Predictor step:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\sum_\alpha \mathbf{e}_\alpha f_\alpha^{eq}\right) = 0; \tag{11}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \left[\sum_\alpha (\mathbf{e}_\alpha)_\beta (\mathbf{e}_\alpha)_\gamma f_\alpha^{eq} + \left(\frac{1}{2\tau}\right) \sum_\alpha (\mathbf{e}_\alpha)_\beta (\mathbf{e}_\alpha)_\gamma f_\alpha^{neq}\right] = 0; \tag{12}$$

corrector step:

$$\frac{\partial \rho}{\partial t} = 0 \tag{13}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \left[\left(1 - \frac{1}{\tau}\right) \sum_\alpha (\mathbf{e}_\alpha)_\beta (\mathbf{e}_\alpha)_\gamma f_\alpha^{neq}\right] = 0; \tag{14}$$

For SHSLBM, the fluid solver is very straightforward. As it is based on the fractional step method, it has two steps namely predictor and corrector. The relationship between kinematic viscosity and the relaxation time is given by Eq. (4)

The predictor and corrector step for SHSLBM is given by:
The predictor step:

$$\rho^* = \sum_\alpha f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t) \tag{15}$$

$$\rho^* \mathbf{u}^* = \sum_\alpha \mathbf{e}_\alpha f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t) \tag{16}$$

The corrector step:

$$\rho(\mathbf{r}, t) = \rho^* \tag{17}$$

$$\rho(\mathbf{r}, t)\mathbf{u}(r, t) = \rho^* \mathbf{u}^* + (\tau - 1)\mathbf{T} \tag{18}$$

where, $\mathbf{T}$ is:

$$\mathbf{T} = \sum \mathbf{e}_\alpha f_\alpha^{eq}(\mathbf{r} + \mathbf{e}_\alpha \delta_t, t) - (\tau - 1)\rho(\mathbf{r}, t - \delta_t)\mathbf{u}(\mathbf{r}, t - \delta_t) \tag{19}$$

B. Immersed Boundary Method and extrapolation technique

In the present work, the geometry is represented by a set of points and the nodes are classified using the point in a polygon scheme. Here, a cross-product is taken into consideration between the nodes of the mesh and the points representing the geometry. Using this, the nodes are identified as inside the geometry or outside the geometry. Depending on which side is solved, the nodes are classified as fluid and solid cells. The solid nodes for which at least one neighbor is a fluid cell are called ghost nodes. The cell classification and the methodology is presented in Fig. 1.

The flow variables at the ghost nodes are calculated using the extrapolation scheme that is implemented using the information from the boundary condition and the points close to the ghost nodes. It has to be noted that for velocity, a no-slip boundary condition is used and for the density, a Neumann boundary condition $\frac{\partial \rho}{\partial n} = 0$ is implemented. The polynomial used for the interpolation is:

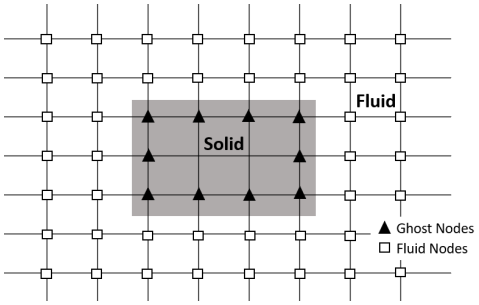

Figure. 1: Cell classification

$$\phi = C_0 + C_1 x + C_2 y + C_3 xy + C_4 x^2 + C_5 y^2 \tag{20}$$

here, the unknowns are the six coefficients. In the present case a least square extrapolation is implemented using 16 points. For both the Neumann and the Dirichlet boundaries, three points from the boundary are selected in the interpolation stencil. For velocity, the system of equations looks like:

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2 y_2 & x_2^2 & y_2^2 \\ 1 & x_3 & y_3 & x_3 y_3 & x_3^2 & y_3^2 \\ 1 & x_4 & y_4 & x_4 y_4 & x_4^2 & y_4^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{16} & y_{16} & x_{16} y_{16} & x_{16}^2 & y_{16}^2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ \vdots \\ C_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \phi_4 \\ \vdots \\ \phi_{16} \end{bmatrix} \tag{21}$$

The first three lines are for the boundary points in Eq. (21). The right hand side vector is zero for boundary points as no slip boundary condition is implemented and the velocity will be zero on the boundary.

The Neumann boundary condition implementation for the density $\frac{\partial \rho}{\partial n} = 0$, the extrapolation polynomial remains the same. However, it is required to incorporate a Neumann boundary condition for density, $\frac{\partial \rho}{\partial n} = 0$. Therefore, a small modification is required whenever boundary points are used in the system of equation.

$$\frac{\partial \phi}{\partial n} = 0 \tag{22}$$

Eq. (22) can also be written as Eq. (23) for a two-dimensional space.

$$\frac{\partial \phi}{\partial x} n_x + \frac{\partial \phi}{\partial y} n_y = 0 \tag{23}$$

$$\frac{\partial \phi}{\partial x} = C_1 + C_3 y + 2C_4 x \tag{24}$$

$$\frac{\partial \phi}{\partial y} = C_2 + C_3 x + 2C_5 y \tag{25}$$

Using Eq. (24) and Eq. (25) and replacing $\frac{\partial \rho}{\partial x}$ and $\frac{\partial \rho}{\partial y}$ in Eq. (23) and rearranging the terms, finally Eq. (26) is obtained.

$$\frac{\partial \phi}{\partial n} = C_1 n_x + C_2 n_y + C_3(n_x y + n_y x) + C_4(2x n_x) + C_5(2y n_y) \tag{26}$$

which can be written as following system of equations:

$$\begin{bmatrix} 0 & n_x & n_y & n_x y_1 + n_y x_1 & 2x_1 n_x & 2y_1 n_y \\ 0 & n_x & n_y & n_x y_2 + n_y x_2 & 2x_2 n_x & 2y_2 n_y \\ 0 & n_x & n_y & n_x y_3 + n_y x_3 & 2x_3 n_x & 2y_3 n_y \\ 1 & x_4 & y_4 & x_4 y_4 & x_4^2 & y_4^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{16} & y_{16} & x_{16} y_{16} & x_{16}^2 & y_{16}^2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ \vdots \\ C_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \phi_4 \\ \vdots \\ \phi_{16} \end{bmatrix} \tag{27}$$

here again, the first three lines represents the Neumann boundary condition.

III. Results and discussions

Simulation results for various cases are presented here including Poiseuille flow, flow around a square cylinder for steady as well as unsteady cases.

A. Poiseuille flow

Poiseuille flow is a 2-D steady flow in a straight channel, which extends in the x direction between $x = 0$ and $x = 1$ and in the y direction between $y = 0$ and $y = 1$. The walls of the channel are parallel to the x-axis. A no-slip condition for the velocity is enforced on these walls. The flow is characterized by a constant pressure gradient along the channel. In the present case, the velocity profile from the analytical solution of the flow is enforced on the inlet and the density is imposed on the outlet. The velocity is parallel to the walls, u_y is zero and u_x is independent of x. The height $l_y = 1$ of the channel represents the characteristic length L.

The analytical solution for velocity and density was found out and the domain was initialised with a constant arbitrary value, although close to the maximum velocity and density. The analytical solution of Poiseuille flow is given by:

$$u_x(y) = -\frac{dP_x}{dx}\frac{1}{2\mu} y(H - y) \tag{28}$$

here, μ is the dynamic viscosity and H is the height of the channel, which is equal to 1. The pressure gradient in x-direction is given by $\frac{dP_x}{dx}$.

A comparison of numerical and analytical solution is presented in the Fig. 2 for different mesh and the convergence plot is presented in Fig. 3. The convergence order obtained is 2. The error is obtained using the L_2 norms of relative error of horizontal velocity u_x. This is given by:

$$L_2 = \sqrt{\frac{1}{N_{total}} \sum_{k=1}^{N_{total}} \left(\frac{u_{numerical} - u_{analytical}}{U_0} \right)^2} \tag{29}$$

where U_0 is the maximum velocity of analytical solution and N_{total} is the total number of nodes in the domain.

A typical SHSLBM is expected to provide a convergence order of 2. This suggest that the implementation of SHSLBM is correct and now the Immersed Boundary method can be implemented using it.

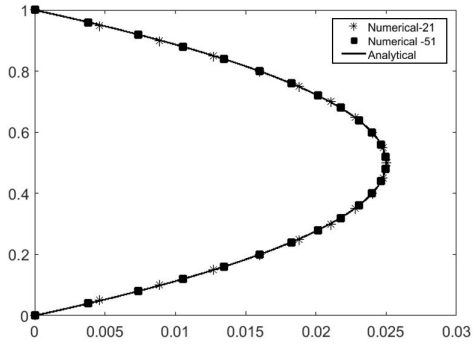

Figure. 2: Comparison of analytical and numerical velocity for mesh size of 51×51 and 21×21

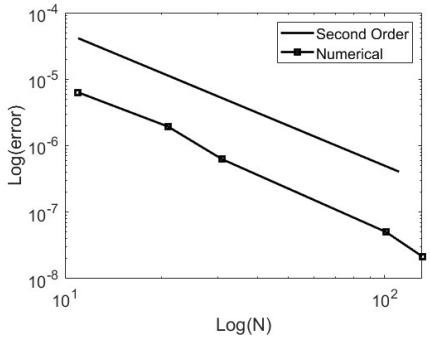

Figure. 3: Convergence for Poiseuille flow on a log-log scale. Here, $N = N_{total}$

B. Poiseuille flow with Immersed Boundary

In the present case, the channel presented in the last section is immersed in the Cartesian mesh in order to implement and test the implementation of IB method in SHSLBM. The boundary conditions are the same as presented in the last section but are implemented using the extrapolation method presented in Sec II.B. The schematic for the case is presented in Fig. 4. The length L_T and height L_H are both 1.0. The width L is 0.1. This makes the length and the width of the channel to be 0.8.

Comparison of the numerical results with the analytical solution for two different mesh sizes, 51×51 and 21×21, is presented in Fig. 5. The solution on a very coarser grid for immersed Poiseuille flow is not very good. However, this problem is not faced for finer mesh which is presented in Fig. 5.

C. Flow around square cylinders

The flow around cylinders is one of the most studied cases in the fluid mechanics, both experimentally and computationally.

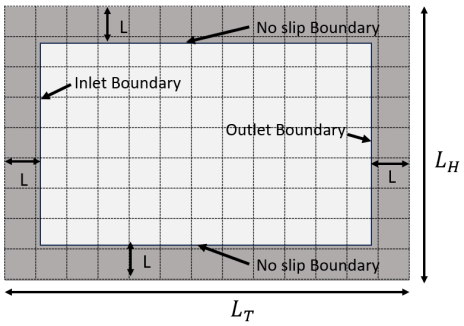

Figure. 4: Schematic for immersed Poiseuille flow

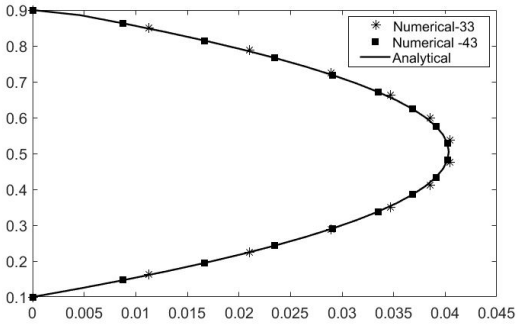

Figure. 5: Comparison of analytical vs numerical velocity for immersed Poiseuille flow for mesh size of 43×43 and 33×33. The mesh size is for the whole domain of 1.0×1.0

In this section, flow around a square cylinder, with Immersed Boundary as shown in Fig. 6a and without immersed boundary, as shown in Fig. 6b, is simulated. The results for the case with IBM are compared to the case without IBM and a code-to-code verification is done by comparing results from Breuer et al. [21]. The domain is initialized with a constant non-zero velocity, close to the maximum Poiseuille velocity and the density is initialized as per the Poiseuille density obtained from the analytical solution. The boundary conditions are a Poiseuille velocity at the inlet, obtained from the analytical solution, and a transmissive boundary at the outlet. It has to be noted that an improper initialization leads to noise in the results and will subsequently need more number of time steps for the solution to get rid of this noise. The disappearance of the noise takes more time steps as the number of nodes are increased in the simulation domain. The schematic for this case is presented in Fig. 7 and is the same as that of the test case presented in [21]. Here, the width of the square cylinder is D and is centered at height $H/2$. The total height of the channel is H. The blockage ratio is fixed at $B = 1/8$ and $B = D/H$. The length of the channel is $L = D \times 50$ and $l = L/4$. Thus, all the dimensions of the test case are known. For all the test cases in this paper, $D = 13$.

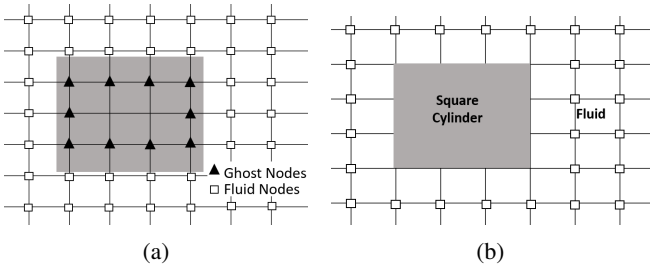

Figure. 6: Square cylinder a) with Immersed Boundary b) without IBM

The number of time steps required for the square cylinder with IBM to obtain a desired results increases considerably compared to the cylinder without IBM. One thing to consider while simulating a square cylinder with IB method is that the cylinder or the geometry should be approximately equidistant from the nodes for all the sides. In other words, the distance between the top face of the square cylinder in Fig. 6a and the ghost nodes corresponding to this face should be equal to the distance between the bottom face and the corresponding bottom ghost nodes. If this is not maintained there will be a non-zero lift, which is undesired. However, the effect on the drag coefficient is negligible.

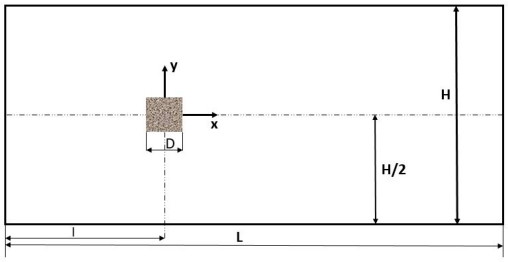

Figure. 7: Schematic for flow around a square cylinder

TABLE. I: Drag Coefficient

Reynolds Number	**Drag coefficients**		
	With IB	Without IB	***Ref. [21]***
30	1.92	1.95	1.95
40	1.72	1.74	1.75
100	1.365	1.355	1.35
133	1.34	1.33	1.32

The comparison of the results are presented in the Table I for drag coefficients for various Reynolds number. The drag coefficient values with immersed boundary method, without IBM and the Ref. [21] are very close to each other all of the cases presented in the Table I. A comparison of difference of maximum and minimum lift coefficient, [max(C_l)

TABLE. II: Lift coefficient comparison

Reynolds Number	**Lift coefficients** [max(C_l) - min(C_l)]		
	With IB	Without IB	***Ref. [21]***
100	0.39	0.39	0.4
133	0.55	0.55	0.55

- $\min(C_l)$], is presented in Table II for the unsteady case, namely Reynolds number of 100 and 133. Also, Strouhal number, $St = fDu_{max}$, comparison is presented for the unsteady cases in Table III. Here, D is the cylinder diameter, u_{max} is the maximum inlet velocity and f is the characteristic frequency determined by a spectral analysis (fast Fourier transformation, FFT) of time series of the lift coefficient C_l.

TABLE. III: Strouhal Number

Reynolds Number	**Strouhal Number**		
	With IB	Without IB	***Ref. [21]***
100	0.14	0.14	0.139
133	0.144	0.145	0.146

Velocity contours are presented in Fig. 8 for Reynolds number of 100 and in Fig. 9 for Reynolds number of 40.

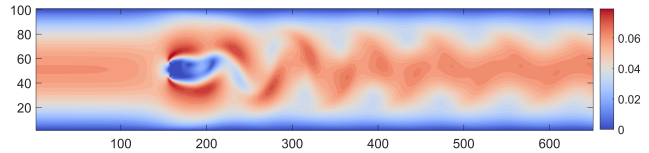

Figure. 8: Velocity contours for flow around a square cylinder with IB method for Reynolds number of 100

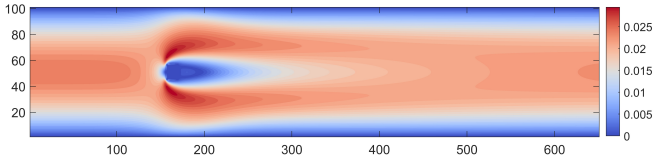

Figure. 9: Velocity contours for flow around a square cylinder with IB method for Reynolds number of 40

CONCLUSION

In this paper a sharp interface Immersed Boundary approach with ghost nodes is implemented for SHSLBM to enhance its usability to simulate flow around geometry not aligned with the mesh. To validate the performance of the newly developed method two benchmark simulations are conducted. The test case with the Poiseuille flow ensure that the SHLBM solver is well implemented. IBM is combined with that of the Poiseuille flow and the result is compared to the analytical solution. The second test case considers flow around square cylinder in a channel where drag coefficient is compared with a previously published work. This further validates the method's capability to handle complex geometrical boundaries and flow separation phenomena. The method is stable for different Reynolds number, retaining the ideal characteristics of a typical SHSLBM scheme.

ACKNOWLEDGMENT

Authors would like to acknowledge the suggestions received from Dr. Sami Ammar, Polytechnique Montreal during various discussions.

REFERENCES

[1] Z. Chen, C. Shu, and D. Tan. A simplified thermal lattice boltzmann method without evolution of distribution functions. *International Journal of Heat and Mass Transfer*, 105:741–757, 2017.

[2] Z. Chen, C. Shu, and D. Tan. Three-dimensional simplified and unconditionally stable lattice boltzmann method for incompressible isothermal and thermal flows. *Physics of Fluids*, 29:053601, 2017.

[3] Z. Chen, C. Shu, D. Tan, and C. Wu. On improvements of simplified and highly stable lattice boltzmann method: Formulations, boundary treatment, and stability analysis. *International Journal for Numerical Methods in Fluids*, 87(4):161–179.

[4] Charles S Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):250, 271, 1972.

[5] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, 2005.

[6] F. Nikfarjam, Y. Cheny, and O. Botella. The ls-stag immersed boundary/cut-cell method for non-newtonian flows in 3d extruded geometries. *Computer Physics Communications*, 226:67 – 80, 2018.

[7] M.D. de Tullio, P. De Palma, G. Iaccarino, G. Pascazio, and M. Napolitano. An immersed boundary method for compressible flows using local grid refinement. *Journal of Computational Physics*, 225(2):2098–2117, 2007.

[8] P. De Palma, M.D. de Tullio, G. Pascazio, and M. Napolitano. An immersed-boundary method for compressible viscous flows. *Computers & Fluids*, 35(7):693–702, 2006. Special Issue Dedicated to Professor Stanley G. Rubin on the Occasion of his 65th Birthday.

[9] Md Sujaat Ali, Renan de Holanda Sousa, M. Ossman Awad, Ricardo Camarero, and Jean-Yves Trépanier. *Comparison of Ghost Cell and one-sided Interpolation method for moving bodies in Immersed Boundary Method*.

[10] Md Sujaat Ali. Two dimensional compressible flow solver for moving geometries using immersed boundary method. Master's thesis, Polytechnique Montréal, 2020.

[11] Rahul Bale, Amneet Pal Singh Bhalla, Boyce E. Griffith, and Makoto Tsubokura. A one-sided direct forcing immersed boundary method using moving least squares. *Journal of Computational Physics*, 440:110359, 2021.

[12] Vamsi Spandan, Detlef Lohse, Marco D. de Tullio, and Roberto Verzicco. A fast moving least squares approximation with adaptive lagrangian mesh refinement for large scale immersed boundary simulations. *Journal of Computational Physics*, 375:228–239, 2018.

[13] M. Haji Mohammadi, F. Sotiropoulos, and J. Brinkerhoff. Moving least squares reconstruction for sharp interface immersed boundary methods. *International Journal for Numerical Methods in Fluids*, 90(2):57–80.

[14] Yegao Qu, Ruchao Shi, and Romesh C. Batra. An immersed boundary formulation for simulating high-speed compressible viscous flows with moving solids. *Journal of Computational Physics*, 354:672–691, 2018.

[15] Jianjian Xin, Zhenlei Chen, Fan Shi, Fulong Shi, and Qiu Jin. A radial basis function–based ghost cell method for complex rigid or flexible moving boundary flows. *International Journal of Computational Methods*, 18(01):2050025, 2021.

[16] Jianjian Xin, Ting-qiu Li, and Fu-long Shi. A radial basis function for reconstructing complex immersed boundaries in ghost cell method. *Journal of Hydrodynamics*, 30(05), 2018.

[17] Francisco Toja-Silva, Julien Favier, and Alfredo Pinelli. Radial basis function (rbf)-based interpolation and spreading for the immersed boundary method. *Computers & Fluids*, 105:66–75, 2014.

[18] Manish Kumar, Somnath Roy, and Md Sujaat Ali. An efficient immersed boundary algorithm for simulation of flows in curved and moving geometries. *Computers Fluids*, 129:159–178, 2016.

[19] Timm Krüger, Halim Kusumaatmaja, Alexandr Kuzmin, Orest Shardt, Goncalo Silva, and Erlend Magnus Viggen. *The Lattice Boltzmann Equation*, pages 61–104. Springer International Publishing, Cham, 2017.

[20] *Simplified and Highly Stable Lattice Boltzmann Method*, chapter Chapter 2, pages 34–53.

[21] M. Breuer, J. Bernsdorf, T. Zeiser, and F. Durst. Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-boltzmann and finite-volume. *International Journal of Heat and Fluid Flow*, 21(2):186–196, 2000.