Computers and Fluids 289 (2025) 106522

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Integration of Lattice Boltzmann-overset method with non-conforming quadtree mesh based on the combination of spatial and Lagrangian-link interpolated streaming technique

Abdallah ElSherbiny, Sébastien Leclaire *

URPEI, Department of Mechanical Engineering, Polytechnique Montréal, 2500 chemin de Polytechnique, Montréal, H3T 1J4, Canada

ARTICLE INFO

Keywords:
Lattice Boltzmann method
Overset method
Lagrangian-link interpolation
Non-conforming quadtree mesh
Non-uniform mesh
Moving boundary treatment

ABSTRACT

This study integrates the two-dimensional Lattice Boltzmann overset approach with a non-conforming quadtree mesh to address fluid flow problems involving dynamic boundaries. The Lattice Boltzmann overset method employs two grids, one fixed and one movable, which can be computationally intensive due to the dual grid setup. A quadtree mesh is employed to reduce the number of nodes to mitigate this resource-demanding issue. Nonetheless, the use of the quadtree introduces challenges related to varying cell levels and spatial displacements. One of the approaches to address these challenges involves the use of an interpolated particle distribution function streaming technique. This study introduces an interpolation method, which initially applies spatial interpolation as a predictor step. Subsequently, this spatial predictor-interpolated value is utilized for a Lagrangian-link corrector interpolation. Furthermore, the study introduces a node-splitting technique aimed at enhancing the efficiency of the proposed interpolation scheme. The method's order of accuracy is maintained without any degradation as a second order, and the flow around a rotating cylinder validates the method as the results align with previously published data.

1. Introduction

Solving the fluid flow with immersed moving bodies has long been a question of great interest with the lattice Boltzmann method (LBM) due to its wide application in industry. Several approaches solved this problem by performing moving boundary treatments, such as the moving overset method. The moving overset method was initially integrated with LBM as the sliding mesh interface approach [1–3], where the computational domain is divided into fixed and moving grids that interact through a shared interface ring. However, employing a single interface led to mass and momentum conservation challenges and generated numerical noise due to the relative motion [4]. Then, the LB-overset method was developed to solve this issue by introducing two communication borders: fixed-to-moving and moving-to-fixed borders [5,6]. These communication borders act as boundary conditions [5]; the fixed-to-moving border acts as a boundary condition for the moving grid, and the moving-to-fixed border acts as a boundary condition for the fixed grid. The LB-overset method is a second-order accuracy method that defines the computational domain into two grids with their respective coordinate systems; these grids are the fixed grid and the moving grid, and the solid body is attached to the latter. Both grids share an overlapping region and communicate through two interfaces using interpolations.

The core of the LB-overset method is the reconstruction of PDFs on the communication borders, acting as boundary conditions for both grids. The PDFs' reconstruction is implemented by macroscopic interpolation [7] and microscopic interpolation, the so-called direct coupling method [6]. The macroscopic interpolation method interpolates the velocity, density, and flux tensor between the two grids on the borders. Then, the equilibrium and non-equilibrium PDFs are reconstructed from the interpolated macroscopic variables using LBM momentum equations. On the other hand, the microscopic interpolation performs the interpolation from the PDFs and then calculates the macroscopic variables from the interpolated PDFs to reconstruct the PDFs through their equilibrium and non-equilibrium parts. Moreover, it needs a rotation of the PDFs before the interpolation, which is a more complex process than the rotation of the macroscopic variables. Furthermore, it required interpolating the macroscopic stress tensor required to reconstruct the non-equilibrium PDFs.

Utilizing a uniform mesh in the LB-overset method or the standard

* Corresponding author.
E-mail address: sebastien.leclaire@polymtl.ca (S. Leclaire).

https://doi.org/10.1016/j.compfluid.2024.106522
Received 15 August 2024; Received in revised form 5 December 2024; Accepted 9 December 2024
Available online 11 December 2024

LBM imposes significant limitations on its applicability for complex boundaries, which requires boundary treatments based on grid refinement to increase the stability and accuracy of LBM and decrease the required computational resources. The boundary treatments based on grid refinement applied on LBM could be classified into body fitting and orthogonal mesh refinement methods.

First, the body-fitting methods modify the grid to conform to the boundaries, such as the body-conforming curvilinear mesh and the unstructured mesh methods. The body-conforming curvilinear mesh method [8,9] gives accurate results but only applies to regular curvilinear boundaries such as circular ones. If the boundary is not near-circular, the mesh may be intersected or not uniform enough to adapt to the LBM. The LBM unstructured meshing methods, such as finite-difference LBM [10–12], finite volume LBM [13], and finite element LBM [14] are based on the concept of independent space and time discretization LBM [15]. Furthermore, an unstructured grid was introduced based on the flux solver LBM [7], which is based on interpolating macroscopic variables around the flux calculation point to construct the equilibrium particle distribution functions (PDFs) the flux solver needs. Besides confirming the boundaries, the unstructured mesh methods also perform grid refinement around the boundaries, giving accurate results as the unstructured mesh conforms to the non-regular boundaries. However, it adds more complex data structures, complex grid generation, and increased computational resources due to additional calculations, contradicting LBM's simplicity advantages on regular grids. Furthermore, some unstructured methods, such as finite-difference LBM, could produce numerical instability at low relaxation time LBM, requiring a high-order discretization scheme [10].

Second, the orthogonal grid refinement methods aim to refine the grid around the boundaries based on orthogonal cells, which could be classified into uniform and non-uniform orthogonal methods. The uniform orthogonal grid refinement methods, such as local grid refinement [16] and multi-block grid [17,18], are based on performing the refinement with uniform refinement blocks.

The local grid refinement method introduces a grid refinement in locations of the most expected solution changes, which divides the mesh into two cells' blocks, fine and coarse cells. The fine-to-coarse displacement ratio n is not required to be two and can reach up to 50. As the grid size changes, the kinematic viscosity must be modified to retrieve the same viscosity and Reynolds number. Consequently, a relaxation parameter must be calculated in the fine mesh by knowing the coarse mesh relaxation parameter. However, the finer mesh performs the collision and streaming steps n times to maintain the same time step in both blocks. Moreover, special treatments on the PDFs transferred between coarse and fine cells must be performed.

Similarly, the multi-block grid method generates fine and coarse blocks in the grid and calculates new relaxation parameters to maintain the same viscosity. Nevertheless, the coarse block displacement is doubled as the finer one [18], maintaining the same dynamic viscosity. Consequently, these blocks could be repeated to reach the required refinement and result in multi-blocks, significantly decreasing the required computational resources [18]. However, similar to the local grid refinement method, it requires special treatments on the blocks' boundaries, and the collision and streaming steps in each fine block are implemented two times its coarser one.

Then, the non-uniform orthogonal grid refinement methods, such as directional refinement method [19], non-uniform staggered cartesian grid (NSCG) [20], quadtree LBM [21,22], and non-conforming orthogonal mesh [23] are based on performing the refinement with a non-uniform orthogonal mesh near the boundaries.

Cheng and Hung [19] introduced the directional refinement method, which applies refinements in specific directions, producing a non-uniform stretched grid containing rectangular cells with a uniform data structure. Moreover, they implemented the interpolated PDFs streaming in the non-uniform nodes using a spatial interpolation of nine surrounding nodes, which is performed using the multiplication of two 1-D Lagrange polynomials applied in the tangential and perpendicular directions.

Valero-Lara and Jansson [20] introduced the NSCG method, which presents a non-uniform refinement based on introducing rectangular cells and has four types of cells: coarse, fine, vertical-fine, and horizontal-fine, and implements scaling on the kinematic viscosity for each cell. Consequently, an arithmetic average relaxation parameter was presented and applied to corner nodes of the vertical-fine and horizontal-fine cells. This arithmetic average relaxation parameter allows all fluid points, whether coarse or fine, to be solved once each time step. However, it requires treatments based on the refinement cell in all lattice directions. Moreover, when applied to 1-D refinement, it reduces the nodes' number compared to local grid refinement. Conversely, it increases the necessary number of nodes when employed in 2D refinement, such as refinement around a cylinder.

The quadtree mesh is a data structure where the cells can be divided into four equivalent quadrants based on refinement requirements. These refinement requirements can be implemented by adding control points in the domain. If a cell contains a control point, it is split into four other cells till it reaches the maximum required level l_{max}. A cell with no splitting has a level equal to zero; with each split, the new cells have their level increased by one. The algorithm of the LB quadtree method in literature [21,22] is based on the local grid refinement method, where the calculation is performed between each quadtree's levels l and the finest cells perform collision and streaming $2^{l_{max}}$ times the most coarse ones. This could lead to an extremely high processing time for high-level quadtree mesh. For instance, if the quadtree has $l_{max} = 6$, so the maximum level cells perform the collision and streaming $2^6 = 64$ times the low level cell. Moreover, the local grid refinement special treatment must be applied between each two consecutive level cells, adding more complexity and processing time.

Finally, Pellerin et al. [23] introduced the non-conforming orthogonal mesh method that generates a quadtree-like mesh with large aspect ratio cells, as the x and y direction splitting are separated. The core of the non-conforming LBM is using bicubic interpolated post-collision PDFs in the streaming step. These interpolated PDFs' locations are offset from the PDF pulling node with the smallest cell size dimensions, which leads to using the same time loop and relaxation time for all nodes in the non-conforming orthogonal mesh. However, it adds more complexity due to the bicubic interpolation, which is difficult and computationally costly in 2D and even more in 3D.

Regarding the grid refinement on the LB-overset method, only two studies perform grid refinement on the LB-overset method. Based on the standard LB-overset method, one study applies a multi-block refinement only on the fixed grid, while the moving grid is a uniform cartesian grid with the finest cells [7,24]. The other one, based on the direct coupling LB-overset, applies the local grid refinement concept as the moving grid is a uniform cartesian grid with fine cells, while the fixed grid has coarse cells [6].

The present study aims to solve the flow around moving boundaries by integrating the LB-overset method with the non-conforming quadtree mesh based on the combination of spatial and lagrangian-link interpolated streaming technique. Our main goal of the study is to verify and validate this approach. The quadtree grid is selected to reduce the total number of nodes, consequently decreasing the computational resources, as the quadtree mesh always has a lower number of nodes than the multi-block refinement for the same refinement level. Moreover, the interpolated PDF streaming technique is used instead of the methods used in the quadtree studies [21,22] to avoid repeating the collision and streaming in the fine cells and the complex implementation on the different level boundaries.

The interpolated PDF streaming technique could be classified based on spatial (Eulerian) interpolation and link (Lagrangian) interpolation. The spatial interpolation methods [19,23] can be adapted to the non-uniform grids. However, they are complex and require high

computational resources and time due to the complex interpolation schemes. Moreover, the complexity and the computational resources will significantly increase for extending the methods to 3D. On the other hand, the Lagrangian-link interpolation scheme, such as Guo's boundary treatment [25], benefited from the LBM lagrangian nature and performed extrapolations on the non-equilibrium PDF on the link. However, it is not adapted to the non-uniform grids because the nodes are not aligned in the same link.

Thus, this study implements a combination of spatial and Lagrangian-link interpolation, which is straightforward, adapted to the non-uniform grids, and reduces the computational resources. Moreover, a streaming node splitting technique is developed to reduce the computational resources, splitting the streaming nodes into direct streaming, direct Lagrangian-link interpolation, and spatial Lagrangian-link interpolation nodes, which prevents the interpolation from being applied on all nodes. The direct streaming nodes are the most condensed area in the quadtree located in the cells with the highest level, where standard LBM streaming is applied and do not require interpolations. Moreover, the direct Lagrangian-link interpolation extends across much of the quadtree grid, affecting nodes surrounded by four square cells with spatial displacements identical and not equal to the minimum displacement. These nodes are situated in all level layers except the maximum level one. Furthermore, the spatial Lagrangian-link interpolation nodes are a few nodes located on the boundaries between the different levels.

The LB-overset method based on the recursive regularized BGK collision model is described in Section 2. Then, the non-conforming quadtree mesh based on the spatial Lagrangian-link interpolated PDF streaming technique is discussed in Section 3. Next, the results are discussed in Section 4 by implementing four test cases: the uniform flow, Poiseuille flow, Taylor-Couette flow, and flow around a rotating cylinder to verify and validate the method. Finally, the conclusion is articulated in Section 5, and an appendix is attached at the end of the paper containing the equations related to the recursive regularized LBM.

2. The recursive regularized LB-overset method

This section briefly introduces the recursive regularized LBM and the LB-overset method. For more information, the reader can revise Latt and Chopard [26], Malaspina [27], and Yoo et al. [7] works. The recursive regularized LB-overset method performs the moving boundary treatments for solving flow around moving objects.

2.1. Recursive regularized LBM

The lattice Boltzmann method is an emerging matured CFD solver based on the principles of kinetic theory. It involves discretizing the lattice Boltzmann equation (LBE) in time and space, utilizing a discretized velocity space denoted as (ξ_i). This discretization is achieved through Gauss-Hermite quadrature, enabling the evaluation of the PDF $f_i(\boldsymbol{x},t)$ at a specific location ($\boldsymbol{x}$), time (t), and lattice direction (i). The LBM's core operation revolves around two primary steps: collision and streaming.

Based on the Bhatnagar–Gross–Krook (BGK) model [28], the collision step is a relaxation process of the PDF $f_i(\boldsymbol{x},t)$ towards the equilibrium PDF $f_i^{eq}(\boldsymbol{x},t)$ after a specific time called relaxation time (τ). In other words, the collision step describes the collision of the PDF $f_i(\boldsymbol{x},t)$ in a specific location, then redefining them with new values based on their equilibrium PDF $f_i^{eq}(\boldsymbol{x},t)$. The streaming step represents the propagation of the post-collision PDFs to the surrounding nodes, each in its direction (i), which obtain the post-streaming PDFs on these nodes. In the following time step Δt, the previous post-streaming PDFs are the current pre-collision ones. The collision and streaming can be represented as

$$f_i(\boldsymbol{x}+\boldsymbol{\xi}_i\,\Delta t, t+\Delta t) - f_i(\boldsymbol{x},t) = -\frac{\Delta t}{\tau}\left(f_i(\boldsymbol{x},t) - f_i^{eq}(\boldsymbol{x},t)\right) \tag{1}$$

The equilibrium PDF f_i^{eq} is typically defined using the Maxwell-Boltzmann distribution, and its formula change depends on the lattice model. For instance, the D2Q9 model is applied for two dimensions flows with population discretized into nine directions ($i = 0:8$). Moreover, the equilibrium PDF is calculated using the weighting function w_i and the local microscopic velocity (ξ_i) associated with the lattice direction i, and the macroscopic variables presented in the flow density (ρ) and velocity ($\boldsymbol{u}$) as

$$f_i^{eq}(\boldsymbol{x},t) = w_i\,\rho\left(1 + \frac{\boldsymbol{\xi}_i\cdot\boldsymbol{u}}{c_s^2} + \frac{1}{2\,c_s^4}\,\mathscr{H}_i : \boldsymbol{uu}\right) \tag{2}$$

Where the Hermite polynomials $\mathscr{H}_{i\alpha\beta} = \xi_{i\alpha}\xi_{i\beta} - c_s^2\,\delta_{\alpha\beta}$, the lattice weights is w_i, the lattice speed of sound is c_s, and $\delta_{\alpha\beta}$ is the Kronecker delta. Moreover, the flow density (ρ) and velocity ($\boldsymbol{u}$) are retrieved from the PDF zeroth and first moments for the D2Q9 model as

$$\rho = \sum_{i=0}^{8} f_i \tag{3}$$

$$\rho u = \sum_{i=0}^{8} \boldsymbol{\xi}_i f_i \tag{4}$$

The regularized BGK model is based on maintaining the second-order Knudsen number ϵ of the Chapman–Enskog expansion [29], which allows the definition of the regularized PDF f_i^{Reg} based on the second-order non-equilibrium PDF f_i^{Neq} as

$$f_i^{Reg} = f_i^{Neq} + f_i^{eq} \tag{5}$$

The second-order non-equilibrium PDF f_i^{Neq} is reconstructed using the non-equilibrium momentum flux tensor $\boldsymbol{\Pi}^{Neq}$ [26] as

$$f_i^{Neq} \approx f_i^{(1)} = \frac{w_i}{2\,c_s^4}\,\mathscr{H}_i\cdot\boldsymbol{\Pi}^{Neq} \tag{6}$$

The non-equilibrium momentum flux tensor $\boldsymbol{\Pi}^{Neq}$ is calculated through the momentum flux tensor $\boldsymbol{\Pi}$ and its equilibrium part $\boldsymbol{\Pi}^{Neq}$ as

$$\boldsymbol{\Pi}^{Neq} = \boldsymbol{\Pi} - \boldsymbol{\Pi}^{eq} \tag{7}$$

$$\Pi_{\alpha\beta} = \sum_{i=0}^{8} \xi_{i\alpha}\,\xi_{i\beta}\,f_i \tag{8}$$

$$\Pi_{\alpha\beta}^{eq} = \sum_{i=0}^{8} \xi_{i\alpha}\,\xi_{i\beta} f_i^{eq} \tag{9}$$

Where α,β covers all possible combinations of the two directions x,y as xx, xy, yx, yy. The regularized LBM significantly improves stability and accuracy by truncating only up to the second-order non-equilibrium part. This approach is computationally efficient, requiring only a few additional steps to calculate the momentum flux tensor and the non-equilibrium distribution function [26].

In contrast to the regularized LBM, which utilizes second-order terms in the equilibrium and non-equilibrium PDFs, the recursive regularized LBM incorporates higher-order terms. This enhancement results in solutions that are both more stable and accurate [27,30]. Furthermore, the higher-order non-equilibrium distribution functions are derived from a recursive formula based on the second-order non-equilibrium tensor $\boldsymbol{a}_1^{(2)}$, resulting in a velocity space discretization error of approximately order $\varnothing\left(Ma^{n+1}\right)$ [27], which effectively reduces the compressibility error that arises due to the Mach number Ma. The equilibrium and non-equilibrium PDFs up to fourth-order are shown in Eqs. (11) and (12), and the Hermite polynomials and high-order non-equilibrium distributions are articulated in Appendix Section I.

$$\boldsymbol{a}_1^{(2)} = \boldsymbol{\Pi}^{Neq} = \sum_{i=0}^{8} \mathscr{H}_i^{(2)} \left(f_i - f_i^{eq}\right) \tag{10}$$

$$f_i^{eq} = w_i\, \rho \Bigg(1 + \frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{c_s^2} + \frac{1}{2\, c_s^4}\, \mathscr{H}_i^{(2)} : \boldsymbol{uu} + \frac{1}{2\, c_s^6} \left(\mathscr{H}_{ixxy}^{(3)} u_x^2 u_y + \mathscr{H}_{ixyy}^{(3)} u_y^2 u_x\right) + \frac{1}{4\, c_s^8} \mathscr{H}_{ixxyy}^{(4)} u_x^2 u_y^2 \Bigg) \tag{11}$$

$$f_i^{Neq} = w_i \Bigg(\frac{1}{2\, c_s^4}\, \mathscr{H}_i^{(2)} : \boldsymbol{a}_1^{(2)} + \frac{1}{2\, c_s^6} \left(\mathscr{H}_{ixxy}^{(3)} a_{1,xxy}^{(3)} + \mathscr{H}_{ixyy}^{(3)} a_{1,xyy}^{(3)}\right) + \frac{1}{4\, c_s^8} \mathscr{H}_{ixxyy}^{(4)} a_{1,xxyy}^{(4)} \Bigg) \tag{12}$$

2.2. LB-overset method

The LB-overset method presented in this section is based on the work of Yoo et al. [7]. The LB-overset approach involves employing two grids with distinct reference frames. The fixed grid adopts the inertial frame, whereas the moving boundary is affixed to the moving grid. Moreover, the fixed grid is addressed through the LBE in Eq. (1). In contrast, the LBE for the moving grid incorporates an extra discrete forcing term, denoted as $F_i(\boldsymbol{x}, t)$ or the Guo forcing term [21] as shown in Eq. (13).

$$f_i(\boldsymbol{x} + \boldsymbol{\xi}_i\, \Delta t, t + \Delta t) - f_i(\boldsymbol{x}, t) = -\frac{\Delta t}{\tau} \left(f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)\right) + \left(1 - 0.5\, \frac{\Delta t}{\tau}\right) F_i(\boldsymbol{x}, t) \tag{13}$$

The discrete force term $F_i(\boldsymbol{x}, t)$ introduces the influence of the inertial density force $\boldsymbol{F}(\boldsymbol{x}, t)$ derived from relative motion. This inertial force comprises the linear acceleration $-\dot{\boldsymbol{V}}$, centripetal acceleration $-2\boldsymbol{\omega} \times \boldsymbol{u}^c$, Coriolis acceleration $-2\boldsymbol{\omega} \times \boldsymbol{u}^c$, and Euler acceleration $-\dot{\boldsymbol{\omega}} \times \boldsymbol{r}$, as shown in Eqs. (14) and (15). Moreover, calculating the discrete force term and the inertial density force are based on the corrected velocity field $\boldsymbol{u}^c$ to retrieve a second-order accuracy. The corrected velocity is calculated using the half-force correction [6] shown in Eq. (16).

$$F_i(\boldsymbol{x}, t) = w_i \left(\frac{(\boldsymbol{\xi}_i - \boldsymbol{u}^c) \cdot \boldsymbol{F}}{c_s^2} + \frac{(\boldsymbol{\xi}_i \cdot \boldsymbol{u}^c)\ (\boldsymbol{\xi}_i \cdot \boldsymbol{F})}{c_s^4}\right) \tag{14}$$

$$\boldsymbol{F} = -\rho(\dot{\boldsymbol{V}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) + 2\boldsymbol{\omega} \times \boldsymbol{u}^c + \dot{\boldsymbol{\omega}} \times \boldsymbol{r}) \tag{15}$$

$$\rho \boldsymbol{u}^c(\boldsymbol{x}, t) = \sum_i \boldsymbol{\xi}_i f_i(\boldsymbol{x}, t) + \boldsymbol{F}(\boldsymbol{x}, t)\ \Delta t/2 \tag{16}$$

The corrected velocity is applied to calculate the equilibrium PDF $f_i^{eq}(\rho, \boldsymbol{u}^c)$ in Eqs. (2) and (11), which leads to an implicit problem as it is also a function of the inertial density force $\boldsymbol{F}(\boldsymbol{x}, t)$. This issue can be resolved by applying the accurate solution for the Coriolis acceleration $\frac{d\boldsymbol{u}}{dt} = -2\boldsymbol{\omega} \times \boldsymbol{u}$ [31], as:

$$\boldsymbol{u}^c = \begin{bmatrix} u_x \\ u_y \end{bmatrix}_{t+\frac{\Delta t}{2}} = \begin{bmatrix} \cos(\omega \Delta t/2) & \sin(\omega \Delta t/2) \\ -\sin(\omega \Delta t/2) & \cos(\omega \Delta t/2) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}_t \tag{17}$$

The fixed and moving grid communicates the macroscopic variables such as density, velocity, and flux tensor through two communication borders: fixed-to-moving and moving-to-fixed. These boundaries serve as boundary conditions for both grids, where the PDFs $f_i(\boldsymbol{x}, t)$ are reconstructed. This reconstruction relies on interpolating the macroscopic variables at each node along these boundaries, using data from the neighboring nodes of the other grid.

The fixed-to-moving treatment seeks to reconstruct the PDFs at the boundaries of the moving grid. Initially, on the fixed grid, the macroscopic variables (ρ, $\boldsymbol{u}$, $\boldsymbol{\Pi}^{Neq}$) are interpolated onto the projected borders of the moving grid. This interpolation process utilizes the neighboring nodes for implementation, using Eqs. (18) to (20), where the interpolated function (ϕ) is required to transfer the macroscopic variables between the two grids on the fixed-to-moving and moving-to-fixed borders, and the superscript (f, m) indicates the variable in the fixed and moving grids, respectively. Subsequently, the geometric adjustment of the velocity and flux tensor is applied to accommodate the disparity in coordinate systems, utilizing the rotation matrix $\mathscr{R}$, as articulated in Eqs. (21) to (23).

$$\rho^f(\boldsymbol{x}^m) = \phi\left(\rho^f\left(\boldsymbol{x}^f\right)\right) \tag{18}$$

$$\boldsymbol{u}^f(\boldsymbol{x}^m) = \phi\left(\boldsymbol{u}^f\left(\boldsymbol{x}^f\right)\right) \tag{19}$$

$$\boldsymbol{\Pi}^{Neq,f}(\boldsymbol{x}^m) = \phi\left(\boldsymbol{\Pi}^{Neq,f}\left(\boldsymbol{x}^f\right)\right) \tag{20}$$

$$\boldsymbol{u}_R^f(\boldsymbol{x}^m) = \mathscr{R} \cdot \boldsymbol{u}^f(\boldsymbol{x}^m) \tag{21}$$

$$\boldsymbol{\Pi}_R^{Neq,\,f}(\boldsymbol{x}^m) = \mathscr{R} \cdot \boldsymbol{\Pi}^{Neq,f}(\boldsymbol{x}^m) \cdot \mathscr{R}^T \tag{22}$$

$$\mathscr{R} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{23}$$

Next, the velocity is adjusted by incorporating the influence of non-inertial acceleration through the subtraction of the relative velocity ($\boldsymbol{v}_r$), expressed as $\boldsymbol{u}_{R,A}^f(\boldsymbol{x}^m) = \boldsymbol{u}_R^f(\boldsymbol{x}^m) - \boldsymbol{v}_r$, where $\boldsymbol{v}_r = \boldsymbol{\omega} \times \boldsymbol{r}$. Here, the subscripts (R, A) denote the adjustments due to the geometric rotation and non-inertial acceleration, respectively. Moreover, the non-equilibrium momentum tensor remains unchanged, as the grid rotation does not affect the shear stresses [7]. Following this, the macroscopic variables at the moving grid border $\boldsymbol{x}^m$ are redefined based on the modified values from the fixed grid, as $\boldsymbol{u}^m(\boldsymbol{x}^m) = \boldsymbol{u}_{R,A}^f(\boldsymbol{x}^m)$, $\boldsymbol{\Pi}^{Neq,m}(\boldsymbol{x}^m) = \boldsymbol{\Pi}^{Neq,f}(\boldsymbol{x}^m)$, and $\rho^m(\boldsymbol{x}^m) = \rho^f(\boldsymbol{x}^m)$. Finally, the PDFs at the border f_i^m are reconstructed by employing the redefined macroscopic variables at the borders. This reconstruction involves reconstituting both equilibrium and non-equilibrium PDFs using Eqs. (11) and (12), along with the incorporation of the half-force correction force term F_i in Eq. (14) as:

$$f_i^m(\boldsymbol{x}^m, t) = f_i^{eq}(\rho^m(\boldsymbol{x}^m), \boldsymbol{u}^m(\boldsymbol{x}^m)) - 0.5\, \boldsymbol{F}_i(\rho^m(\boldsymbol{x}^m), \boldsymbol{u}^m(\boldsymbol{x}^m)) + f_i^{Neq}\left(\boldsymbol{u}^m(\boldsymbol{x}^m), \boldsymbol{\Pi}^{Neq,m}(\boldsymbol{x}^m)\right) \tag{24}$$

The moving-to-fixed treatment is carried out following the same procedures as the fixed-to-moving one, with the notable difference of adding the relative velocity to the total velocity, expressed as $\boldsymbol{u}_A^m\left(\boldsymbol{x}^f\right) = \boldsymbol{u}^m\left(\boldsymbol{x}^f\right) + \boldsymbol{v}_r$, instead of subtracting it. This adjustment is made before the interpolation and the grid rotation [7]. Additionally, there is no need for the half-force correction when reconstructing the PDFs on this border, as the fixed frame serves as the inertial reference frame [6]. Algorithm 1 illustrates the procedures of the LB-overset method.

Algorithm 1
The LB-overset algorithm:

Define/calculate Initial conditions, ω, $\theta = \omega\ \Delta t$, $\dot{\omega} = d\omega/dt$
While $t < t_f$ **Do**
Do boundary conditions
Calculate $(\boldsymbol{F})$, $(\boldsymbol{u}^c)$, $(F_i(\boldsymbol{x}, t))$ using Eqs. (14) to (17) for both fixed and moving grids
Calculate f_i^{eq} using Eq. (11) for both grids, **use** $(\boldsymbol{u}^c)$ for moving grid.
Calculate $\boldsymbol{\Pi}^{Neq}$ using Eq. (10) for both grids.
Calculate f_i^{Neq} using Eq. (12) for both grids.
Calculate f_i using Eq. (5) for fixed grids, **use** Eq. (24) for moving grid.
Interpolate ρ, $\boldsymbol{u}$, $\boldsymbol{\Pi}^{Neq}$ on fixed-to-moving boundary using Eqs. (18) to (20).
Calculate $\boldsymbol{u}_R^f(\boldsymbol{x}^m)$, $\boldsymbol{\Pi}_R^{Neq,\,f}(\boldsymbol{x}^m)$ using Eqs. (21) to (23).
Calculate $\boldsymbol{u}_{R,A}^f(\boldsymbol{x}^m) = \boldsymbol{u}_R^f(\boldsymbol{x}^m) - \boldsymbol{v}_r$.
Reconstruct f_i^{eq}, f_i^{Neq}, f_i on fixed-to-moving boundary.
Calculate $\boldsymbol{u}_A^m\left(\boldsymbol{x}^f\right) = \boldsymbol{u}^m\left(\boldsymbol{x}^f\right) + \boldsymbol{v}_r$
Interpolate ρ, $\boldsymbol{u}$, $\boldsymbol{\Pi}^{Neq}$ in moving-to-fixed boundary.

(*continued on next page*)

Algorithm 1 (*continued*)

Calculate $\boldsymbol{u}$, $\boldsymbol{\Pi}^{Neq}$ in moving-to-fixed boundary similar to Eqs. (21) to (23).
Reconstruct f_i^{eq}, f_i^{Neq}, f_i on fixed-to-moving boundary.
Do collision and streaming using Eq. (1) in both grids.
Update Macroscopic variables ρ, $\boldsymbol{u}$ **in both grids**
Do geometric rotation
End while

3. Non-conforming quadtree mesh based on interpolated PDF streaming technique

In this section, the quadtree mesh generation is described in Section 3.1, with the illustration of the streaming nodes splitting. Then, the spatial Lagrangian-link interpolated PDF streaming technique is discussed in Section 3.2. Finally, Section 3.3 articulated the spatial Lagrangian-link technique.

3.1. Quadtree mesh and streaming nodes' splitting

The non-conforming quadtree mesh is a refinement mesh technique where the mesh nodes do not necessarily lay on the boundaries. It also reduces the total number of nodes, reducing the required computational resources. The quadtree mesh generation core is splitting the cells into four equivalent quadrants by specified refinement criteria. This refinement criteria can be implemented by defining control points in the domain. Each cell that contains a control point is divided into four new cells: northeast NE, northwest NW, southeast SE, and southwest SW, and five new nodes located on half of the spatial spacing, as shown in Fig. 3.1. Moreover, the newly generated cell's level increased by one, and the refinements continued until they reached the maximum

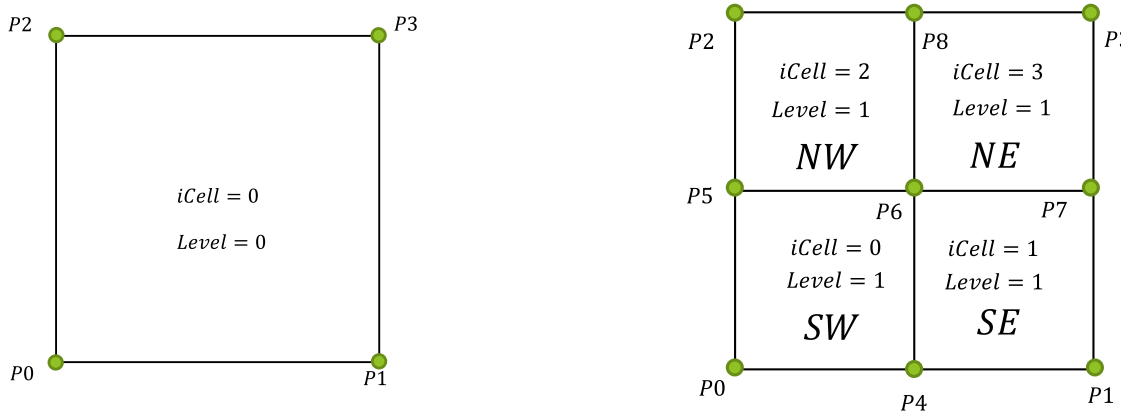


Fig. 3.1. Schematic shows the quadtree mesh splitting.

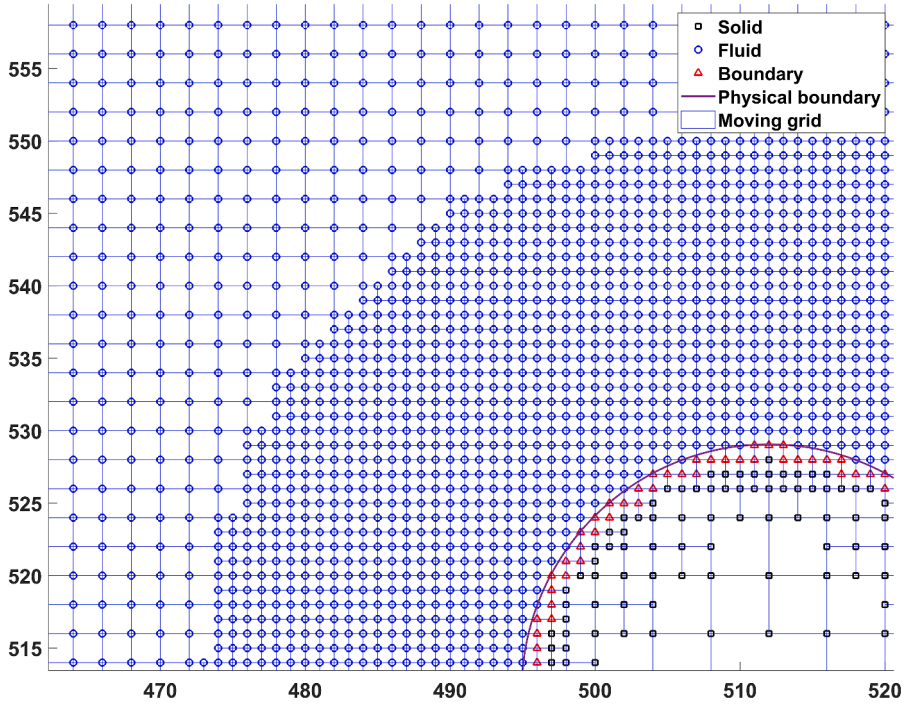


Fig. 3.2. An example of a circular boundary shows the grid nodes' classification as solid, fluid, and boundary nodes. The solid node is the unfilled black circle, the fluid node is the unfilled blue diamonds, the boundary nodes are the unfilled red triangles, and the physical boundary is the continuous purple line.

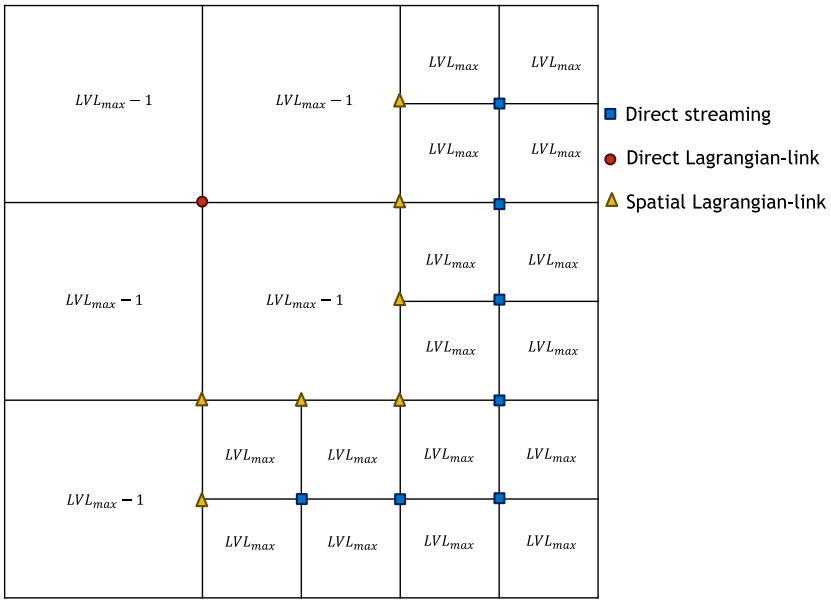


Fig. 3.3. Schematic shows the three classifications of the streaming nodes. The blue squares are the direct streaming, the green circles are the direct Lagrangian-link, and the yellow triangles are the spatial Lagrangian-link nodes.

required level.

Furthermore, a boundary node detection algorithm has been created to categorize the grid into solid, boundary, and fluid nodes using the winding number algorithm [32], as shown in Fig. 3.2. This classification allows us to conduct collision and streaming operations for the fluid nodes, implement suitable boundary treatments for the boundary nodes, and effectively exclude any solid nodes from the simulations, reducing the required computational resources.

As discussed in the introduction, the spatial Lagrangian-link interpolation is not required to be implemented on all nodes, and it is implemented only on the nodes that are not located on the link. Thus, the quadtree domain is further classified into three node categories: direct streaming, direct Lagrangian-link, and spatial Lagrangian-link nodes, as shown in Fig. 3.3.

The "direct streaming nodes" are nodes where all surrounding cells share the same minimal spatial displacement within the domain. As a result, these nodes possess the accurate local relaxation time and viscosity, obviating the need for any interpolation. Thus, they directly undergo a streaming step similar to the standard LBM. Furthermore, these direct streaming nodes are concentrated in the domain's most compact region, eliminating the interpolation requirement and reducing computational effort. These nodes can be identified by employing a condition based on the cells surrounding them, which share the same level, and this level is the highest grid level.

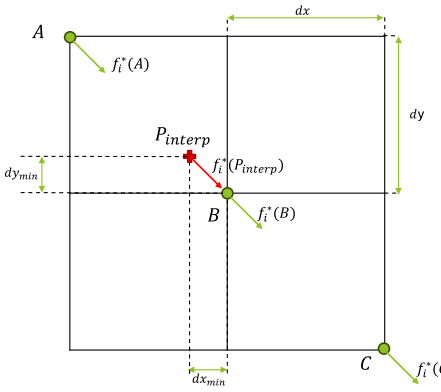


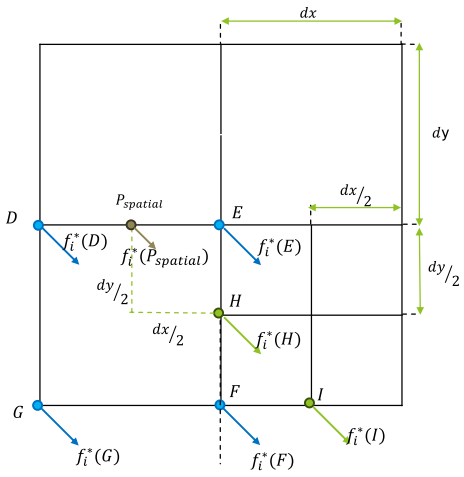
Fig. 3.4. The interpolated post-collision PDFs. The green circles and arrows are the lagrangian-link nodes and their corresponding post-collision PDFs, and the red cross and arrow are the interpolation location and the interpolated post-collision PDF.

Then, the "direct Lagrangian-link nodes" are the majority of the quadtree mesh. These nodes employ a straightforward Lagrangian-link interpolation method, avoiding the necessity to perform spatial interpolation on all nodes, resulting in significant reductions in computational resources and effort. These nodes can be identified by employing a condition based on the cells surrounding them, which share the same level, and this level does not equal the highest grid level.

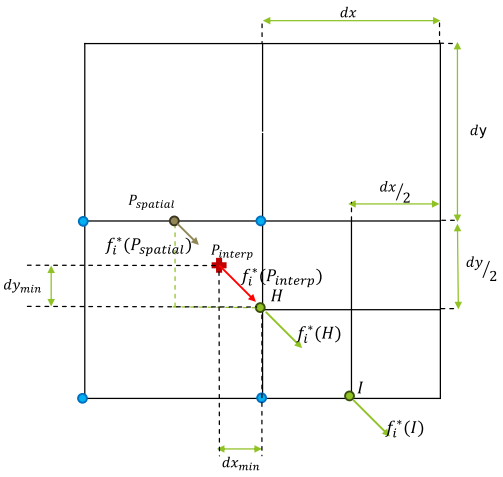
Finally, the "spatial Lagrangian-link nodes" represent the nodes where the spatial Lagrangian-link interpolation is carried out where the spatial displacements of the surrounding cells are not equal. These nodes are exclusively present along the boundaries that separate different levels of layers, constituting a minority among all grid nodes. Consequently, the need for spatial Lagrangian-link interpolation arises only for a few nodes, resulting in significant computational resource savings. These nodes can be identified by employing a condition based on the cells surrounding them, which do not have the same level. Furthermore, this condition could be further enhanced by considering the link direction and removing the collinear nodes; however, it is not considered in this study because the quantity of such nodes is relatively limited, which almost does not affect the computational resources. More details are provided about the Lagrangian-link interpolation in Section 3.2 and the spatial Lagrangian-link interpolation in Section 3.3.

3.2. Lagrangian-link interpolated PDF streaming technique

The interpolated PDF streaming is based on interpolating the post-collision PDFs f_i^* on specific locations, which enables the same relaxation time and time step for all nodes in the grid despite their levels. These interpolated PDFs' locations P_{interp} are offset from the pulling node with the smallest cell size in the quadtree mesh, as shown in Fig. 3.4. As LBM also has Lagrangian bases, interpolating the post-collision PDFs can be implemented on the Lagrangian path with the particles moving on the same link. Consequently, the interpolated post-collision PDFs are obtained using the post-collision PDFs of the nodes on the link, revealing the direct lagrangian-link interpolation. This interpolation is a straightforward 1-D interpolation that consumes low computational resources compared to spatial interpolators such as bicubic interpolator.



a) Schematic of the predictor step of spatial interpolation as node D are not colinear with $\overrightarrow{HI}$.



b) Schematic of the second step of the spatial Lagrangian-link interpolation, showing the Lagranigian-link interpolation using the predicted spatial PDF.

Fig. 3.5. Schematics showing the spatial Lagrangian-link interpolation. The blue circles and arrows are the surrounding nodes and PDFs used in spatial interpolation, the grey circle is the location of the spatial interpolation location, and the red cross is the interpolated post-collision PDF location.

In this study, the interpolation is performed using the Lagrangian polynomial interpolation. This interpolation is applied in all directions, resulting in all required post-streaming interpolated PDFs used in the streaming process. The interpolated PDF $f_i^*(P_{interp})$ can be reconstructed using the three points A, B, and C which have enough information $f_i^*(A)$, $f_i^*(B)$, and $f_i^*(C)$ to reconstruct a quadratic parabola as:

$$f_i^*(P_{interp}) = \phi_{Lag}(A, B, c, f_i^*(A), f_i^*(B), f_i^*(C)) \tag{25}$$

Where ϕ_{Lag} is the 1-D quadratic Lagrangian interpolation function.

3.3. Spatial Lagrangian-link interpolated PDF streaming technique

The quadtree mesh is a non-uniform grid, which does not guarantee the correct lagrangian path of the particles as the nodes may not be on the same link, as shown in Fig. 3.5.a). The node D is not collinear with $\overrightarrow{HI}$, so if we used directly the location of the node D for interpolation it will lead to numerical instability. Thus, the spatial Lagrangian-link interpolated scheme is introduced to implement at first a predictor step where the post-collision PDF is interpolated on the path's known location $P_{spatial}$ using a spatial interpolator, as shown in Fig. 3.5.a). The spatial interpolation implemented in this study is the bilinear interpolation, which uses the four nodes D, E, F, and G and their corresponding post-collision PDFs $f_i^*(D)$, $f_i^*(E)$, $f_i^*(F)$, and $f_i^*(G)$ as:

$$f_i^*(P_{spatial}) = \phi_{Spatial}(D,\ E,\ F,\ G, f_i^*(D), f_i^*(E), f_i^*(F),\ f_i^*(G)) \tag{26}$$

Then, the lagrangian-link interpolation is implemented as a corrector step to calculate $f_i^*(P_{interp})$ using the predicted interpolated post-collision PDF at $P_{spatial}$ and the other two existing nodes on the link H, and I, as shown in Fig. 3.5.b), using the following equation:

$$f_i^*(P_{interp}) = \phi_{Lag}(P_{spatial},\ H,\ i, f_i^*(P_{spatial}), f_i^*(H), f_i^*(I)) \tag{27}$$

Where the $\phi_{Spatial}$ is a bilinear interpolator acting as a predictor, while the ϕ_{Lag} is a quadratic 1-D interpolator acting as a corrector.

This approach significantly minimizes the computational resources required compared to utilizing a purely spatial interpolated streaming scheme like the bicubic method introduced by Pellerin et al. [23]. This reduction is achieved by diminishing the required nodes needed for interpolation. Moreover, it is a straightforward process that does not necessitate extensive computational effort, primarily because the spatial Lagrangian-link interpolation involves explicit equations. Moreover, it is applicable to any orthogonal type of grid, including grids with rectangular cells, and is more suitable for 3D extension due to lower computational resources and effort. Furthermore, it is important to note that the spatial interpolation is not entirely quadratic, as it utilizes bilinear interpolation. However, it is employed as a predictor step on a very small number of nodes, which is expected to have a negligible impact on the solution's accuracy. Additionally, using bilinear requires only four points and is good for parallelization as it keeps the scheme as local as possible.

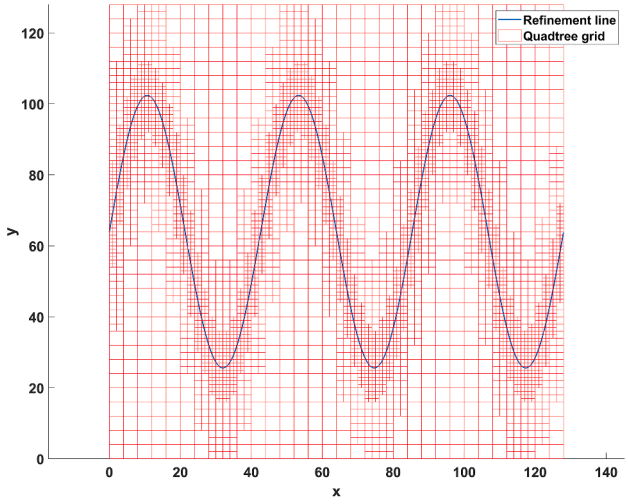
4. The result and discussion

This section aims to verify and validate the LB-overset with spatial Lagrangian-link interpolated PDF streaming method by discussing four test cases. In Section 4.1, the uniform flow is implemented to ensure no slight flow distortion due to the moving grid motion and the streaming interpolated method applied in both the fixed and moving grid. Then, Section 4.2 discussed the Poiseuille flow, which presents a more complex flow as the velocity gradient is introduced. Moreover, the second order of accuracy is verified for the LB-overset with spatial Lagrangian-link interpolated PDF streaming method. After that, the Taylor-Couette flow is illustrated in Section 4.3, where the method is applied to a curved boundary problem and verified by retrieving the analytical solution velocity profile. Finally, the flow around a rotating cylinder is implemented at different Reynolds numbers and rotational ratios to validate the present study in Section 4.4.

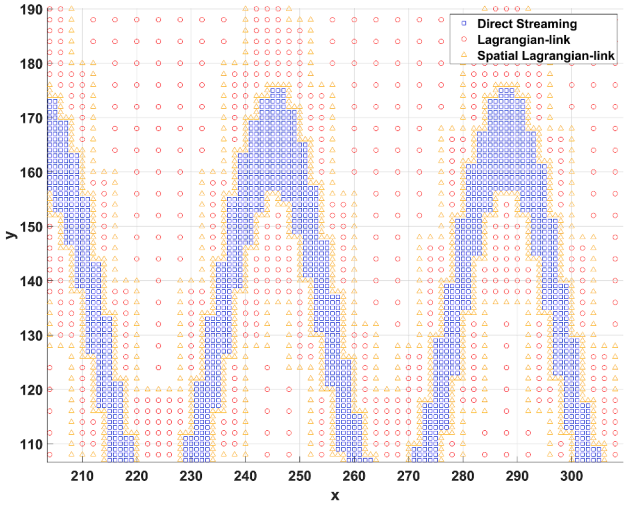
For all the test cases, the moving grid is the quadtree grid, while the fixed grid is a uniform grid with displacement equal to the largest moving grid's displacement. Moreover, All the test cases implemented with the lattice system of units, where the minimum spatial displacement, time step, and density are unity $\Delta x_{min} = \Delta y_{min} = \Delta t = \rho = 1$; thus, the fixed grid spatial displacement is multiplied by $2^{LVL_{max}}$ as $\Delta x_f = \Delta y_f = 2^{LVL_{max}}$, where LVL_{max} is the maximum quadtree grid level.

4.1. Uniform flow

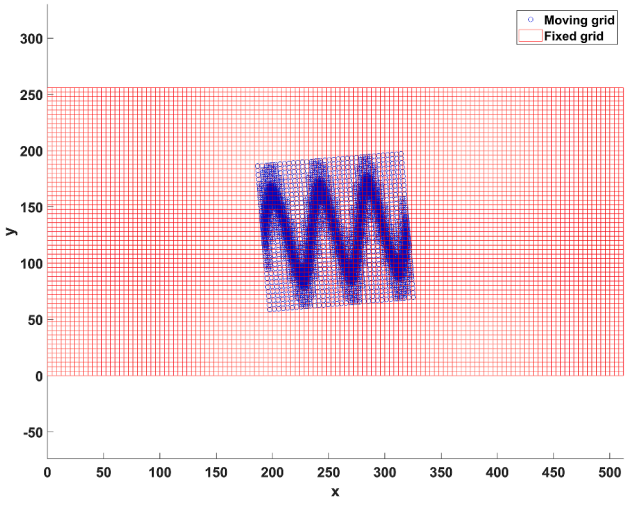
Uniform flow is employed to ensure no flow field distortion, even in the presence of a rotating grid and the utilization of a quadtree mesh.



a) Quadtree grid refinement.



b) Streaming nodes splitting of the moving grid. The blue squares are the direct streaming nodes, the red circles are the direct Lagrangian-link nodes, and the orange triangles are the spatial Lagrangian-link nodes.



c) The overlapped fixed and moving grids.

Fig. 4.1. The mesh refinement based on sinusoidal wave control points.

The quadtree mesh is designed with a sinusoidal wave to apply a non-uniform pattern on the moving grid, as shown in Fig. 4.1.a). Moreover, the moving grid nodes are separated into three nodes of streaming: direct streaming, direct Lagrangian-link, and spatial Lagrangian-link nodes, as shown in Fig. 4.1.b). Furthermore, the overlapped fixed and moving grids are shown in Fig. 4.1.c), where the moving grid is based on sinusoidal wave refinement, and the fixed grid is scaled with $2^{LVL_{max}}$ to maintian the minimum spatial displacments equals one.

The inlet and outlet boundary conditions are Zou and He velocity inlet and pressure outlet [33]. The upper and lower boundary conditions are straight velocity implemented according to the Inamuro boundary condition [34]. The simulation parameters are the fixed grid size of 128 $\times$ 64 cells, relaxation time $\tau = 0.7$, Reynolds number $Re = \frac{u_\infty L}{\upsilon} = 100$, reference length $L = N_y^f = 64$, kinematic viscosity $\upsilon = \frac{1}{3}(\tau - 0.5)$, u_∞ is the freestream velocity, Mach number $Ma = u_{max}/c_s = 0.1776$, rotational speed $\omega = 10^{-4}$, and the fixed-to-moving grid size ratio $N_y^f / N_y^m = 2$, where N_y^f, N_y^m are the number of nodes in the vertical direction for the fixed and moving grids, respectively.

The simulation starts with zero velocity and unity density in the overall domain, with the moving grid starting rotation from the first time step t_s. The error L_1^u is calculated as

$$L_1^u = \frac{1}{N} \sum_{k=0}^{N-1} \frac{\| \boldsymbol{u}(\boldsymbol{x_k}) - \boldsymbol{u}_{ref}(\boldsymbol{x_k}) \|}{\max(\boldsymbol{u}_{ref})} \tag{28}$$

Where the reference velocity $\boldsymbol{u}_{ref}$ is the velocity of the uniform flow field, the nodes' index is k, and N is the total number of nodes. The moving grid rotates for $t_f = 100,000$ time-step, which covers more than two rotations of the moving grid around its center.

The simulation is also tested regarding the error L_1^u, its time-average value $\widehat{L}_1^u$, the pressure error L_1^P, the pressure noise at the final time $\frac{\partial P}{\partial t}|_{t_f}$,

Table 1

The L_1^u, $\widehat{L}_1^u$, L_1^P and $\frac{\overline{\partial P}}{\partial t}$ after grid rotation at t_f.

Variable	Velocity		Pressure	
	L_1^u	$\widehat{L}_1^u$	L_1^P	$\frac{\overline{\partial P}}{\partial t}$
Present study	1.4687e-09	1.1572e-09	3.7188e-10	−3.3019e-14

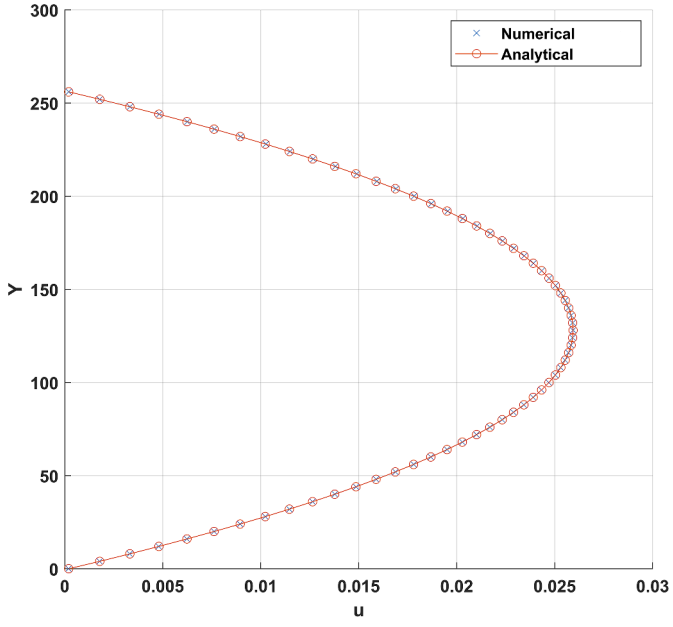


Fig. 4.2. Velocity profile after the moving grid at 75 % of the channel length for the test case of 128×64 at t_f.

and its space-average value $\overline{\frac{\partial P}{\partial t}}$ as

$$\widehat{L}_1^u = \frac{1}{(t_f - t_s)} \sum_{t=t_s}^{t_f} L_1^u(t) \tag{29}$$

$$L_1^P = \frac{1}{N} \sum_{k=0}^{N-1} \frac{\| P(\boldsymbol{x}_k) - P_{ref}(\boldsymbol{x}_k) \|}{\max(P_{ref})} \tag{30}$$

$$\left.\frac{\partial P}{\partial t}\right|_{t_f} = \frac{P_{t_f} - P_{t_f - \Delta t}}{\Delta t} \tag{31}$$

$$\overline{\frac{\partial P}{\partial t}} = \frac{1}{N} \sum_{k=0}^{N-1} \left.\frac{\partial P}{\partial t}\right|_{t_f,k} \tag{32}$$

The error L_1^u is calculated to evaluate the effect of the rotation and the streaming interpolation technique on a non-uniform grid refinement. The average-time value indicates the performance over time after the grid motion. The pressure noise indicates the noise accompanied by the particle distribution function reconstruction on the border during the rotation. Moreover, the particle distribution function directly affects the pressure calculation on each node. Thus, the pressure change with time indicates the accuracy of PDF reconstruction and the streaming interpolation effect.

Table 1 shows that the values of L_1^u, $\widehat{L}_1^u$, L_1^P and $\overline{\frac{\partial P}{\partial t}}$ at the final time step t_f, which reveal that the error is almost near the machine's accuracy and the simulation is not affected by the spatial Lagrangian-link streaming

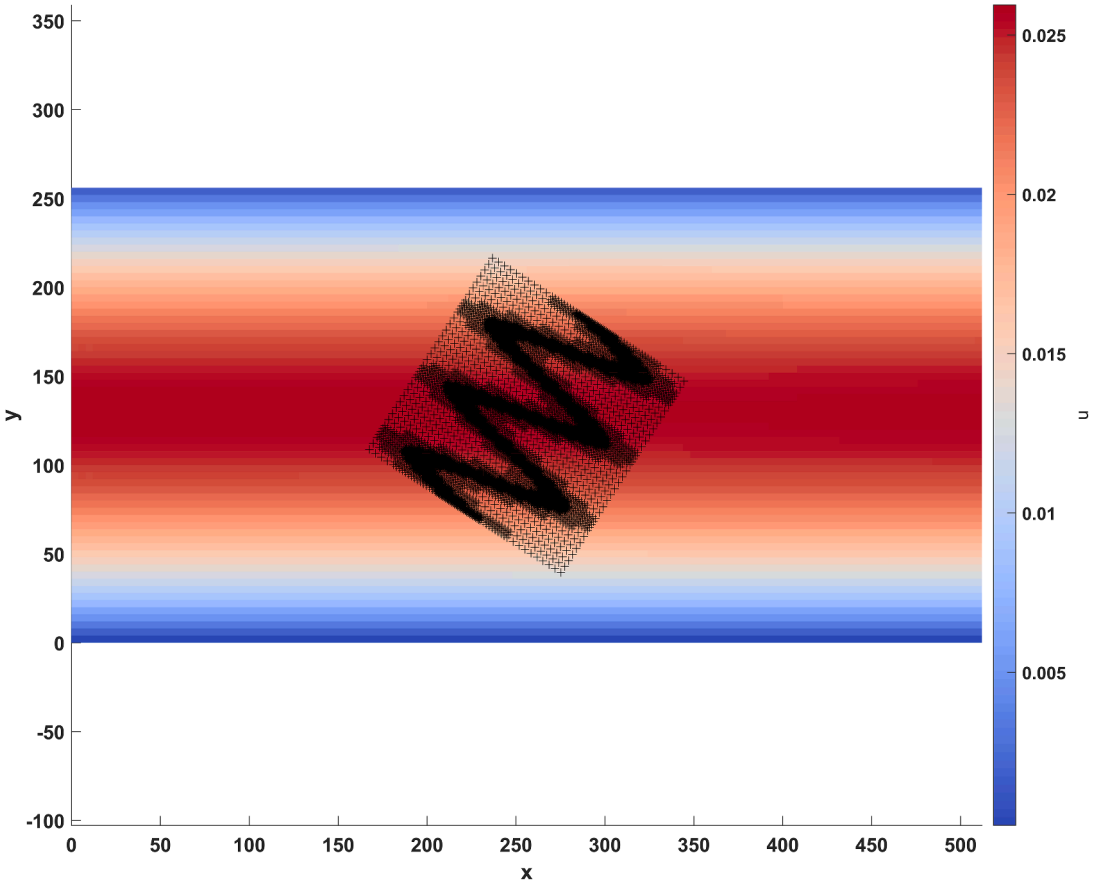
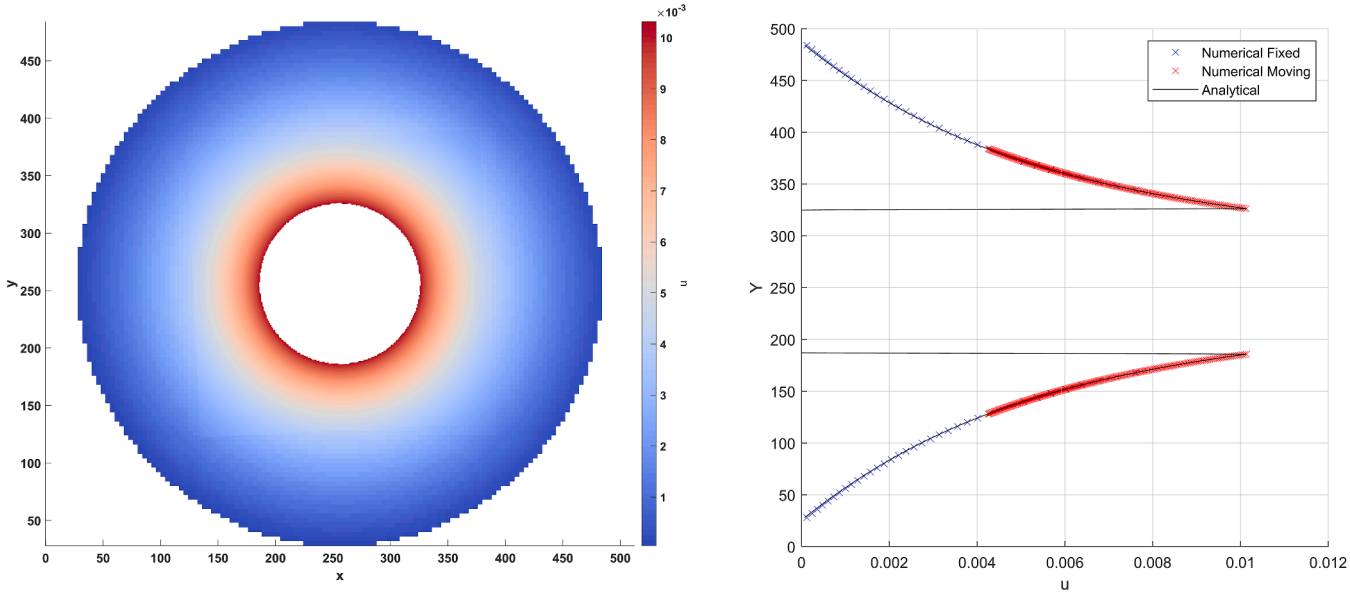


Fig. 4.3. Velocity field of the overlapped grids for the test case of 128×64 at t_f.

Table 2

The Poiseuille flow $\overline{\frac{\partial P}{\partial t}}$, L_2^P, L_2^u, and L_2^P and L_2^u order of accuracy.

No	$N_x^f \times N_y^f$	$\overline{\frac{\partial P}{\partial t}}$	L_2^P	L_2^P Order of accuracy	L_2^u	L_2^u Order of accuracy
1	32 × 16	6.0643e-10	6. 3314e-05	–	1.1105e-04	–
2	64 × 32	3.3999e-10	1.0585E-05	2.58	3.0251e-05	1.84
3	128 × 64	3.0966e-10	2.3135E-06	2.19	7.4543e-06	2.02
4	256 × 128	1.5488e-11	5.6510E-07	2.03	1.8601e-06	2.00



a) Overlapped grids' velocity field.

b) velocity profile of the numerical and analytical solution at the middle of the domain.

Fig. 4.4. Taylor-Couette flow velocity field of the overlapped grids at t_f.

technique and the rotation of the moving grid. This is expected since the flow is constant and the 1D quadratic interpolation and the 2D bilinear interpolation are used, so they should capture the constant profile accurately.

4.2. Poiseuille flow

Poiseuille flow is introduced to add flow complexity by including a velocity gradient, testing the impact of the presented method on the simulation, and testing the order of accuracy as the flow solution is parabolic in the continuum limit. The moving quadtree grid is based on a sinusoidal control points wave similar to Fig. 4.1. The inlet and outlet boundary conditions are velocity inlet and pressure outlet evaluated using the Zou and He [33], while the far field of the lower and upper surfaces are halfway bounceback [35].

The simulation is performed for four sizes of the fixed gird as 32 × 16, 64 × 32, 128 × 64, and 256 × 128, relaxation time $\tau = 1.0$, Reynolds number $Re = \frac{u_\infty L}{\upsilon} = 10$, reference length $L = N_y^f + 1$ to enable the halfway bounceback to be implemented on the half-displacement of the upper and lower surfaces, and the fixed-to-moving grid size ratio of $N_y^f / N_y^m = 2$. The analytical solution u_{Ana} and P_{Ana} of the velocity profile is

$$u_{Ana} = u_{x,\,max}\left(\frac{4y}{L} - \frac{4y^2}{L^2}\right) \tag{33}$$

$$P_{Ana} = P_{inlet} + \frac{\Delta P}{X}x_k = P_{inlet} + \frac{P_{inlet} - P_{outlet}}{X}x_k \tag{34}$$

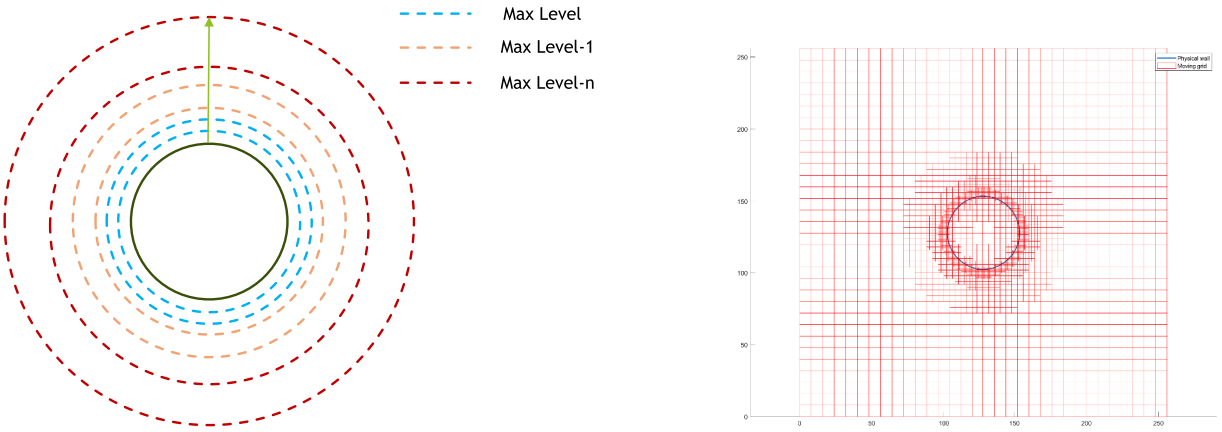
Where X and x_k is the channel length and the nodes location in longitudinal direction, P_{inlet} and P_{outlet} are the pressure at the inlet and outlet boundary condition, respectively. Similar to the uniform flow, the simulation starts with zero velocity and unity density in the overall domain, with the moving grid starting rotation from the first time step t_s to the final time step t_f. The final time step is implemented using stopping criteria ϵ or reaching 10^6 time step to ensure the steady-state solution for the different grid sizes as

$$\epsilon = \frac{|u(\boldsymbol{x}_k, t) - u(\boldsymbol{x}_k, t - \Delta t)|}{u(\boldsymbol{x}_k, t - \Delta t)} \leq 10^{-10} \tag{35}$$

The numerical and analytical solutions' velocity profiles are the same, as shown in Fig. 4.2. Moreover, the velocity field of the overlapped grids is shown in Fig. 4.3, as there is no deviation between the fixed and moving grids on the moving-to-fixed border, ensuring the correct velocity field and the implementation of the Poiseuille flow.

The simulations are also tested through the pressure-noise space-average value $\overline{\frac{\partial P}{\partial t}}$ as in Eqn. (32), the L2 pressure error norm L_2^P as in Eqn. (28) with $\boldsymbol{u}_{ref}$ is the analytical solution, and. Moreover, the L2 velocity and pressure error norms L_2^u, L_2^P, are calculated and the corresponding order of accuracy to ensure that the method did not change the order of accuracy of the standard LBM. The L_2^u and L_2^P, are calculated on the fixed grid and the empty nodes in the fixed grid are reconstructed using bicubic interpolation from the moving grid as:

$$L_2^u = \sqrt{\frac{\sum_{k=0}^{N-1}\left(\,\|\,\boldsymbol{u}(\boldsymbol{x}_k) - \boldsymbol{u}_{ref}(\boldsymbol{x}_k)\|\right)^2}{N_x^f\, N_y^f}} \tag{36}$$



a) Schematic of different layers of control points

b) A close-up view of a generated mesh showing different level layers based on the different layers of control points. The quadtree mesh is shown in red cells, and the physical boundary is shown in a blue line.

Fig. 4.5. Different layers of control points for flow around a cylinder. The presented figure is coarser than the one used in the simulation for visualization purposes.

$$L_2^P = \sqrt{\frac{\sum_{k=0}^{N-1} \left(\| P(\boldsymbol{x}_k) - P_{ref}(\boldsymbol{x}_k) \| \right)^2}{N_x^f \, N_y^f}} \tag{37}$$

Where N is the fixed grid nodes' number, $N_x^f$and N_y^f are the cell's number of the fixed grid in x and y directions, k are the fixed grid nodes, and $\boldsymbol{u}_{ref}$ and P_{ref} are the velocity and pressure analytical solution, respectively.

Table 2 shows that the L_2^u and L_2^P are decreasing with the increasing grid size, showing low error values even with low mesh size. Moreover, $\frac{\partial P}{\partial t}$ gives very low error values which are almost near the machine accuracy. Furthermore, the order of accuracy converges to 2, which verifies the second order of accuracy in space for the present study, and the order of accuracy did not deteriorate as the bilinear interpolation is used as a predictor for a very low number of nodes.

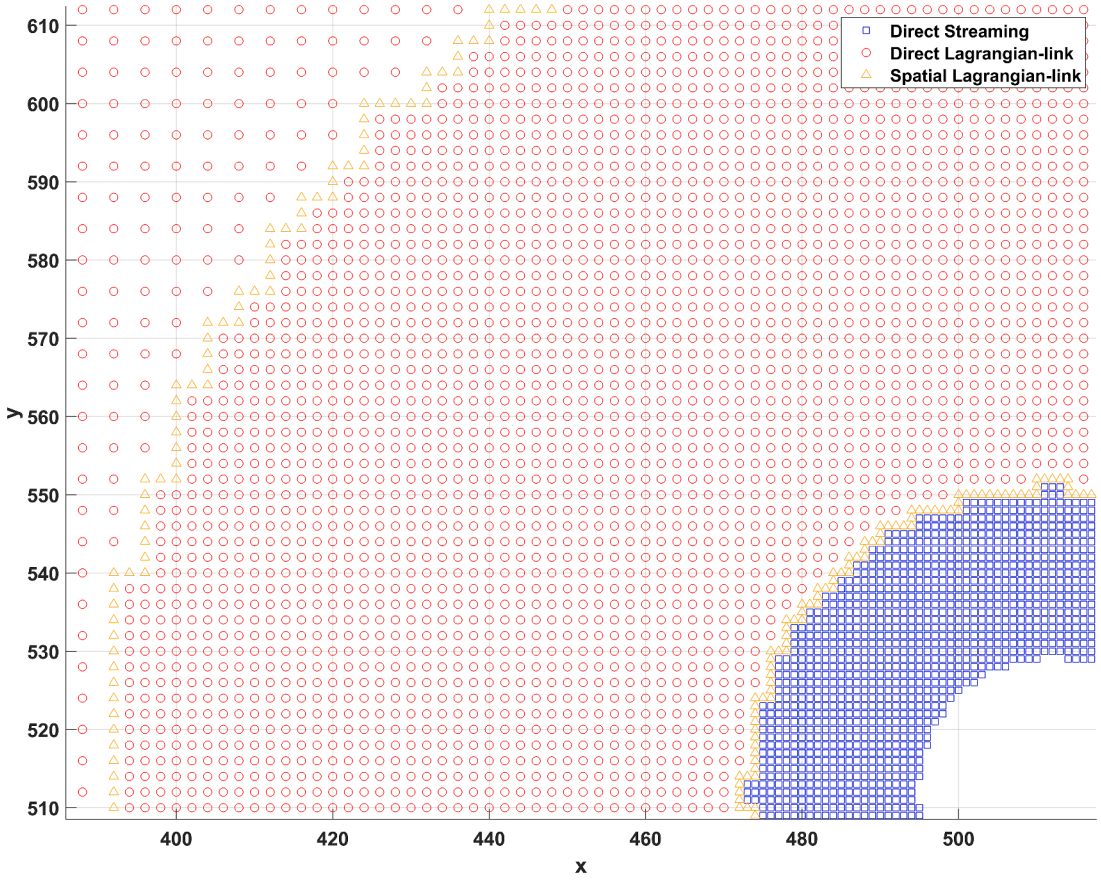


Fig. 4.6. An example of the moving quadtree grid streaming nodes splitting. The blue squares are the direct streaming nodes, the red circles are the direct Lagrangian-link nodes, and the orange triangles are the spatial Lagrangian-link nodes.

Table 3
The lift and drag coefficients and Strouhal number. The metric values of Kang et al. [38] are extracted from figures.

Case			a	$N_x^f \times N_y^f$	Ma	τ	$\overline{C}_L$	$\overline{C}_D$	$C_L^{'}$	St
1	Ours, present study	100	2	512 × 512	0.04	0.595	5.519	0.894	0	0
	Kang et al. [38]			–	–	–	5.499	0.87	0	0
2	Ours, present study	100	1	512 × 512	0.04	0.595	2.507	1.282	0.372	0.167
	Kang et al. [38]			–	–	–	2.48	1.31	0.366	0.167
	Tang and Ingham [39]			–	–	–	1.973	1.008	–	–
3	Ours, present study	60	2	512 × 512	0.04	0.658	5.649	0.711	0	0
	Kang et al. [38]			–	–	–	5.614	0.67	0	0
4	Ours, present study	60	1	256 × 256	0.08	0.657	2.593	1.269	0.145	0.1394
	Kang et al. [38]			–	–	–	2.55	1.167	0.123	0.139
	Tang and Ingham [39]			–	–	–	2.249	1.219	–	–
5	Ours, present study	20	1	256 × 256	0.04	0.736	2.794	1.944	0	0
	Kang et al. [38]			–	–	–	2.76	–	0	0
	Badr et al. [40]			–	–	–	2.79	1.91	0	0
6	Ours, present study	20	0.5	256 × 256	0.04	0.736	1.379	2.08	0	0
	Ours, LB-overset without quadtree			1200 × 1200	0.08	0.76	1.388	2.055	0	0
	Kang et al. [38]			–	–	–	1.345	–	0	0
	Badr et al. [40]			–	–	–	1.39	2.12	0	0

4.3. Taylor–Couette flow

The Taylor-Couette flow setup examines the method when incorporating boundary rotation and a more steep velocity profile. This test case is implemented to ensure that the method works correctly with the curved boundary treatment of Guo's extrapolation method by comparing the analytical and numerical solutions velocity profiles. In this configuration, the flow occurs between two cylinders. The outer cylinder is located in the fixed grid, while the inner cylinder is positioned within the moving grid, denoted as borders 1 and 2 in Fig. 4.4. a), respectively. Moreover, the inner cylinder's boundary enforces a zero velocity using the curved boundary treatment provided by the Guo extrapolation technique [25]. As a result, the influence of the inner cylinder's rotation is only attributable to the rotation of the moving grid.

The two cylinders' centers are in the center of a 128 × 128 square, and the outer radius $R_{out} = 0.45H$, where H is the square edge length. Moreover, the inner circle $R_{in} = \beta R_{out}$, where β is the radii ratio, the reference length is the thickness between the two cylinders $L = R_{out} - R_{in}$, relaxation time $\tau = 1.0$, Reynolds number $Re = 100$, the angular speed $\omega = u_{max}/R_{in}$, and the fixed-to-moving grid size ratio of $N_y^f/N_y^m = 2$. The domain is initialized with zero velocity and unity density, with moving grid rotation starting in the first time step till the final time step t_f based on the stopping criteria in Eq. (35).

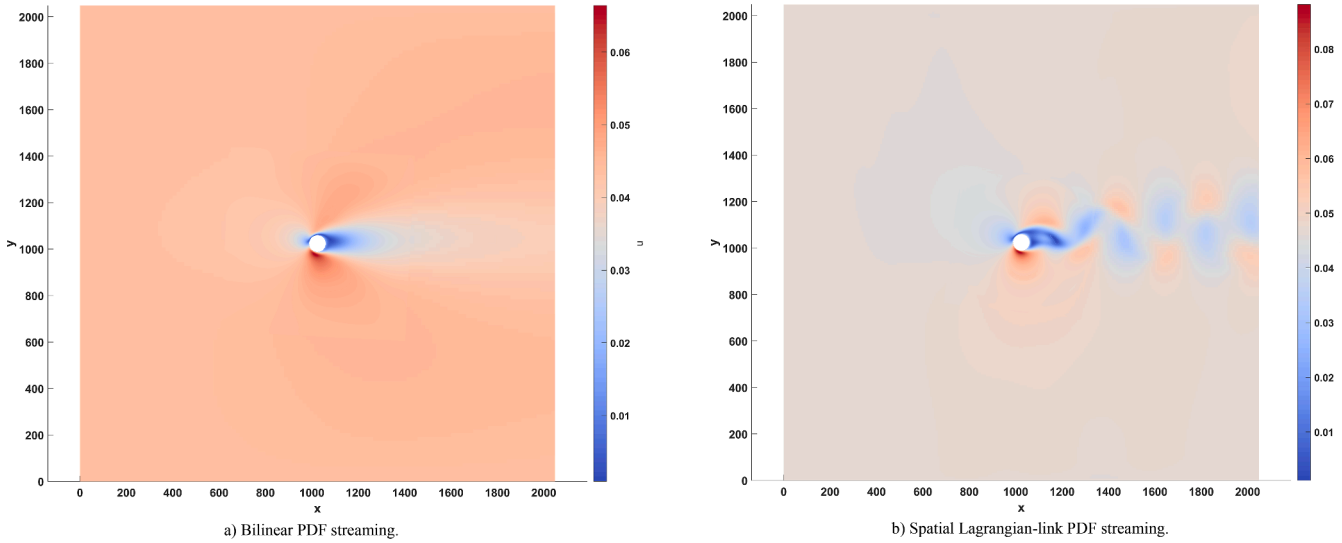
The analytical solution for the Taylo–Couette flow [36] is based on the inner and outer cylinders' radiuses, their corresponding velocities, and the radius value $r(x,y)$, as shown in Eq. (38). In this test case, the outer cylinder velocity is zero, while the inner has the maximum speed. Thus, the analytical solution could be presented as a function of the radii ratio β as shown in Eq. (39).

As described in reference [36], the analytical solution for the Taylor–Couette flow relies on the radii of the inner and outer cylinders and their respective velocities, as represented by Eq. (38). In the present study, the outer cylinder maintains a zero velocity while the inner cylinder attains the maximum speed. Consequently, the analytical solution can be expressed as a function of the radius ratio β, as in Eq. (39).

$$u_{Ana}(x,y) = u_{inner}\left(\frac{\frac{R_{out}}{r(x,y)} - \frac{r(x,y)}{R_{out}}}{\frac{R_{out}}{R_{in}} - \frac{R_{in}}{R_{out}}}\right) + u_{outer}\left(\frac{\frac{r(x,y)}{R_{in}} - \frac{R_{in}}{r(x,y)}}{\frac{R_{out}}{R_{in}} - \frac{R_{in}}{R_{out}}}\right) \tag{38}$$

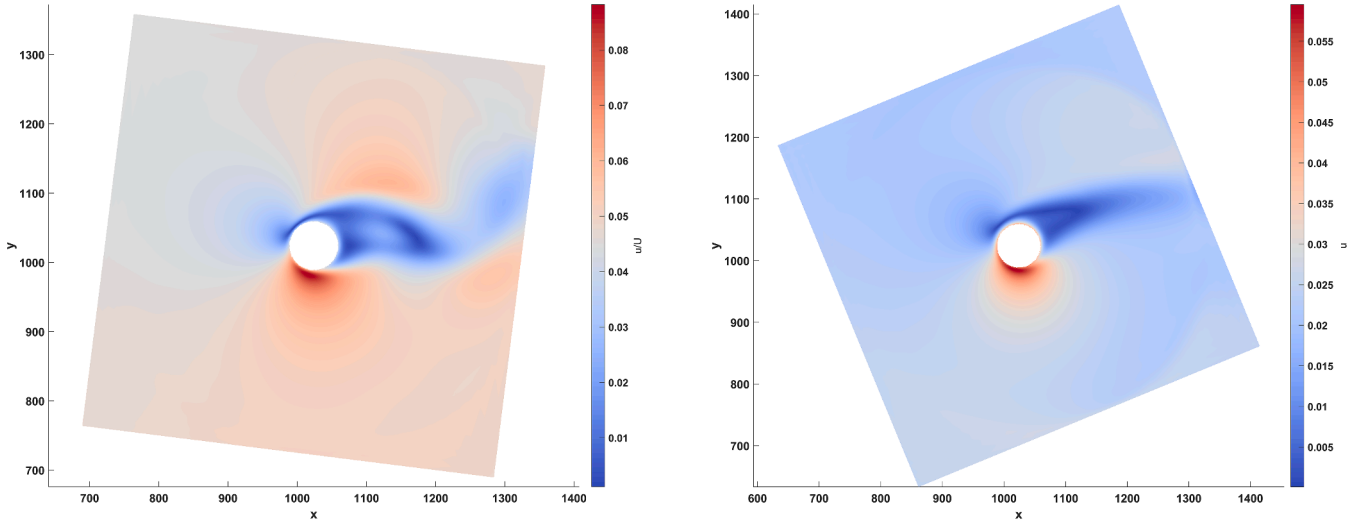
$$u_{Ana}(x,y) = u_{max}\left(\frac{\beta}{1-\beta^2}\right)\left(\frac{R_{out}}{r(x,y)} - \frac{r(x,y)}{R_{out}}\right) \tag{39}$$

The analytical solution shows that the lower the radii ratio β, the



a) Bilinear PDF streaming. b) Spatial Lagrangian-link PDF streaming.

Fig. 4.7. Comparison of the velocity profile using the Bilinear PDF streaming and the spatial Lagrangian-link PDF streaming for $Re = 100$ and $a = 1$. The colorbar is regarding to the velocity values .



a) $Re = 100$ and $a = 1$. b) $Re = 100$ and $a = 2$.

Fig. 4.8. Comparison of the velocity profile of the moving grid regarding the steady and the unsteady solution depending on the rotational ratio. The color bar is regarding the velocity values.

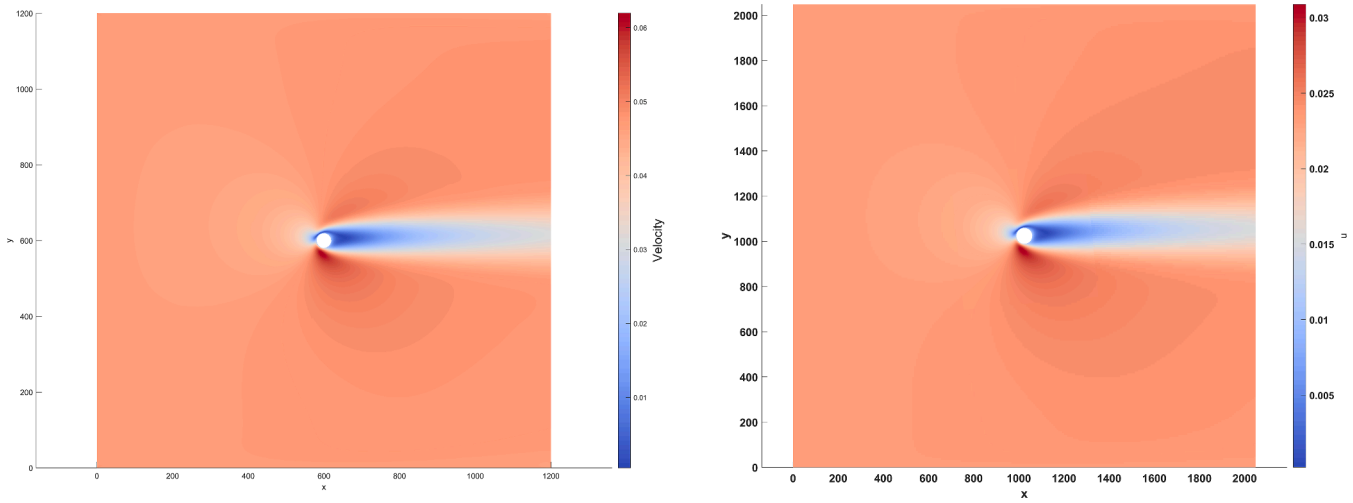


Fig. 4.9. Comparison of the velocity profile using the LB-overset method with and without the quadtree spatial Lagrangian-link PDF streaming for $Re = 20$ and $a = 0.5$. The colorbar is regarding the velocity values.

steeper the velocity profile; thus, the radii ratio $\beta = 0.3$ is selected. The numerical solution velocity field and velocity profile of the numerical and analytical solution are shown in Fig. 4.4. a) and b), respectively. The figures show a good velocity profile, as the numerical and analytical velocity profiles are identical. Moreover, the L1 velocity error $e_1^u = \frac{1}{N}\sum_{k=0}^{N-1} |u(\boldsymbol{x}_k) - u_{Ana}(\boldsymbol{x}_k)|$ is calculated in both fixed and moving grids and equals $e_1^u = 4.5798\text{E} - 05$ which reveals a low value of the velocity error.

4.4. Flow around a rotating cylinder

The flow around a rotating cylinder aims to validate the LB-overset spatial Lagrangian-link PDF streaming method by comparing the lift and drag coefficients with the literature at different Reynolds numbers and rotational speeds. The rotating cylinder is located in the moving quadtree grid, while the fixed grid is a uniform coarse grid. Moreover, the refinement control points are designated along the perimeters and adjusted at varying layer depths, each adhering to distinct level criteria to facilitate a gradual refinement, as shown in Fig. 4.5. Moreover, the moving grid streaming nodes splitting of the direct streaming, Lagrangian link, and spatial Lagrangian-link nodes is performed as shown in Fig. 4.6.

The Zou and He velocity input and pressure output boundary conditions, as described in reference [33], are applied to the inlet and outlet boundary conditions. For the lower and upper surfaces, straight velocity boundary conditions are enforced using the Inamuro boundary condition method [34]. A curved boundary treatment is also employed by utilizing the extrapolation Guo technique, as outlined in reference [25]. This technique involves constructing the post-collision particle distribution function on the boundaries by extrapolating key parameters such as density, velocity, and non-equilibrium distribution function from adjacent fluid nodes. The simulation starts with zero velocity and unity density in the overall domain and the total forces acting on the cylinder boundary $\boldsymbol{F_B}$ are calculated according to the momentum exchange theory [37] as

$$\boldsymbol{F_B} = \sum_{k\in B_n}\sum_{i=1}^{8} \boldsymbol{\xi}_i \left(f_i^*(\boldsymbol{X}_k, t) + f_{\bar{i}}^*(\boldsymbol{X}_k + \boldsymbol{\xi}_i\delta_t, t)\right)\cdot\delta_k \tag{40}$$

Here, B_n represents the set of boundary nodes and δ_k is equal to 1 when a node is fluid and 0 otherwise. Boundary nodes are those nodes surrounded by both fluid and solid nodes. The post-collision PDFs at these boundary nodes are denoted as $f_i^*(X_b,\ t)$, while $f_{\bar{i}}^*(X_b+\xi_i\delta_t,\ t)$ are the post-collision PDFs of neighboring nodes in the opposite direction. Furthermore, the lift and drag coefficients at zero angle of attack are determined from the total force components $\boldsymbol{F_B} = F_x\hat{i} + F_y\hat{j}$. However, this force should be transformed to be evaluated in the inertial fixed frame $\boldsymbol{F_B^f} = F_x^f\hat{i} + F_y^f\hat{j}$ as:

$$\boldsymbol{F_B^f} = \mathscr{R}^T\ \boldsymbol{F_B} \tag{41}$$

$$C_L = \frac{2\ F_y^f}{\rho\ u_\infty^2\ D} \tag{42}$$

$$C_D = \frac{2\ F_x^f}{\rho\ u_\infty^2\ D} \tag{43}$$

Where D is the cylinder diameter and u_∞ is the freestream velocity. The method validation is implemented by comparing the lift and drag coefficients (C_L, C_D) of the current study with previously published data at different rotation ratios

$a = \omega\ D/2\ u_\infty$ and fixed grid sizes $N_x^f \times N_y^f$. Table 3 shows the lift coefficient amplitude C_L', mean drag coefficient $\overline{C_D}$ and Strouhal number $St = f\ L/u_\infty$ for different Reynolds number and rotational ratios, where f is the vortex shedding frequency, which agrees with the published data.

The present study demonstrates strong agreement with the published data across all cases, and any disparities in these metrics were primarily attributed to the different methodologies and boundary conditions employed. For instance, Kang et al. employed an approach that addressed unsteady incompressible flow with boundary nodes conforming to the cylinder walls. In contrast, our study utilized an unsteady weakly compressible solver, and the boundary nodes were not aligned with the cylinder wall. Nevertheless, these variations did not significantly impact the results for each metric compared to the reference data.

Moreover, in examining cases 1, 2, 3, and 4, the fixed grid sizes were larger than the other cases. This increase in grid size is primarily because these cases exhibit a higher Reynolds number, necessitating an enlargement of the grid size to avoid a low relaxation time τ, near 0.5. In cases 1 and 3, there is a high rotation ratio a, which demands a reduction in the freestream velocity u_∞ to mitigate the effects of a high rotational speed $\omega = 2\ u_\infty\ a/D$. This rotational speed reduction can be achieved by increasing the grid size, thereby allowing for a decrease in the freestream velocity while keeping the relaxation time constant and increasing the cylinder diameter D, which is related to the fixed grid height. Furthermore, cases 1, 3, 5, and 6 have flows produced a steady solution similar to the published data where the Strouhal number is zero, while cases 2 and 4 produce an unsteady flow with vortex street with values matched with the published data. Additionally, in test case 6, a comparison was made between the results of the present study and our LB-overset simulation without quadtree (direct streaming for all nodes), which have excellent agreement across all metrics, revealing that the spatial Lagrangian-link streaming did not degrade the solution.

More investigation of the present study's effect is performed by comparing the velocity field of a bilinear PDF streaming and the spatial Lagrangian-link PDF streaming method shown in Fig. 4.7a) and b), respectively. Both simulations have $Re = 100$, rotational ratio $a = 1$, and fixed grid size 256×256 cells. So, the fixed grid height and length are scaled by the value of $2^{LVL_{max}}$. The bilinear PDF streaming is implemented by interpolating the post-collision PDF required for streaming using the bilinear interpolation only without any Lagrangian-link interpolation. In such case, the velocity profile must have a vortex street in the wake of the cylinder as the rotational ratio did not reach the steady solution rotational ratio a_s, in $Re = 100$ the $a_s \approx 2$. The bilinear PDF streaming failed to recover the accurate velocity field because it did not generate the vortex street phenomenon. This limitation can be explained by the fact that bilinear PDF streaming only captures the interpolated PDF value from a single cell, which is insufficient for accurately representing the velocity gradient. On the other hand, the spatial Lagrangian-link PDF streaming retrieved the correct velocity profile as the vortex street was developed.

Additionally, Fig. 4.8 adds more qualitative validation by showing the effect of the rotational speed on the flow field solution steadiness. Both simulations are implemented using the LB-overset quadtree spatial Lagrangian-link method. Fig. 4.8.a) shows the velocity profile for case 2, $Re = 100$ and $a = 1$, which resulted in an unsteady solution and developing a vortex street because the rotational ratio a is less than the steady solution rotational ratio $a_s \approx 2$. On the other hand, Fig. 4.8.b) shows velocity ptofile for case 1, $Re = 100$ and $a = 2$, where a steady solution was developed because the rotational ratio reaches its steadiness value $a_s \approx 2$.

Moreover, Fig. 4.9a) and b) show the velocity profiles of the standard LB-overset method (no quadtree mesh) and LB-overset with quadtree spatial Lagrangian-link PDF streaming technique, respectively, for $Re = 20$ and $a = 0.5$. The LB-overset method without quadtree applies direct streaming of standard LBM for all nodes in the fixed and moving grid. Both simulations retrieve a correct velocity field where the steady solution emerged. Moreover, their lift and drag coefficients, shown in Table 3, case 6, are quite similar, indicating that the proposed spatial Lagrangian-link technique did not negatively affect the solution compared with the LBM standard streaming. Moreover, the LB-overset method without quadtree required high computational resources as the fixed grid size $1200 \times 1200 = 1,440,000$ nodes and the moving grid size is $300 \times 300 = 90,000$, compared with the LB-overset with a quadtree is for fixed grid $256 \times 256 = 65536$ nodes and moving grid 35,500 nodes. This leads to a decrease in the total number of nodes by 1,428,964 nodes with a percentage of $1,428,964,/(6,553,6 + 3,550,0) = ,206,9.8,8$%. Furthermore, the LB-overset method without quadtree was not able to simulate high Reynolds number cases with high rotational speed, such as $Re = 100$ and $Re = 60$ with $a = 1.0$ on AMD Hawaii, 12,288 MB GPU device due to a lack of computational resources. This is achieved by using the quadtree spatial Lagrangian-link PDF streaming technique, as it decreases the required computational resources significantly.

5. Conclusion

In the present study, a spatial Lagrangian-link PDF interpolation streaming technique is introduced and integrated with the LB-overset method with quadtree mesh refinement. The method is verified and validated with the previously published numerical data and does not show any degradation in the results or the order of accuracy. First, the uniform flow test case shows the correct implementation of the proposed method as no flow distortion appeared, and the pressure noise reaches the machine's accuracy, revealing that the solution was not affected by the moving grid rotation or the reconstruction of the interpolated PDFs. Then, Poiseuille flow verifies the method by retrieving a second order of accuracy in space and the correct velocity profile, showing that using the spatial Lagrangian interpolation method did not degrade the order of accuracy. Next, the Taylor-Couette flow shows the correct implementation of the present method with the curved boundary treatment of Gou's extrapolation method by retrieving the correct velocity profile. After that, the flow around the rotating cylinder validates the proposed method by retrieving matched results with the published lift, drag, and Strouhal number data.

Moreover, using the bilinear interpolation alone for interpolating the post-collision PDFs is insufficient, as the interpolator is performed using one cell and could not capture the effect of the other surrounding cell,

consequently not capturing the gradient effect. This issue does not appear in the spatial Lagrangian-link method because the spatial interpolation is performed as a predictor step on a low number of points, while the Lagrangian-link interpolation acts as a corrector step of a quadratic interpolation.

Additionally, using the quadtree mesh significantly decreases the required computational resources. Also, the spatial Lagrangian-link interpolation decreases the number of nodes required for implementing the PDF streaming than the spatial interpolations, decreasing the required computational resources and increasing the simulation efficiency. Moreover, the node-splitting technique allows the interpolations or the direct streaming to be executed depending on the nodes' requirements, further increasing the simulation efficiency. Also, the current approach could be employed with non-conforming quadtree rectangular grids, such as Pellerin et al.'s work [23], and could be extended to 3D applications because of the Lagrangian nature of LBM. This extension would offer additional advantages, as it would demand significantly fewer nodes for interpolation compared to the extension of spatial interpolators like the bicubic method, which decreases the required computational resources and increases the simulation efficiency.

CRediT authorship contribution statement

Abdallah ElSherbiny: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Sébastien Leclaire:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The author, Abdallah ElSherbiny, would like to thank Professor Jean-Yves Trepanier, Department of Mechanical Engineering, Polytechnique Montréal, and Dr. Nicolas Pellerin, Aircraft Performance Engineer, National Research Council of Canada, Ottawa, for constructive criticism of the manuscript and the fruitful discussions about the present study. Moreover, the authors acknowledge the support of the Natural Sciences and Engineering Council of Canada (NSERC), [funding reference number RGPIN-2020-04512].

Appendix

The Appendix shows the Hermite polynomials and non-equilibrium high-order terms needed for implementing the equilibrium and non-equilibrium particle distribution function.

Hermite polynomials and higher-order non-equilibrium distributions

The Hermite polynomials in Eqs. (11) and (12) are presented in the equations from (44) to (46). For example, the term $\mathscr{H}_i^{(2)} : \boldsymbol{uu}$ in Eq. (11) is calculated using Eq. (44) as $\mathscr{H}_i^{(2)} : \boldsymbol{uu} = \left(\xi_{i,x}^2 - c_s^2\right)u_x^2 + \left(\xi_{i,y}^2 - c_s^2\right)u_y^2 + 2\left(\xi_{i,x}\xi_{i,y}\right)u_x u_y$. Moreover, the high-order nonequlibrium terms are shown in Eqs. (12), and (10) are illustrated in equations from (47) to (49).

$$\mathscr{H}_{i,xx}^{(2)} = \xi_{i,x}^2 - c_s^2, \mathscr{H}_{i,xy}^{(2)} = \xi_{i,x}\xi_{i,y} \tag{44}$$

$$\mathscr{H}_{i,xxy}^{(3)} = \left(\xi_{i,x}^2 - c_s^2\right)\xi_{i,y} \tag{45}$$

$$\mathscr{H}_{i,xxyy}^{(4)} = \left(\xi_{i,x}^2 - c_s^2\right)\left(\xi_{i,y}^2 - c_s^2\right) \tag{46}$$

$$a_{1,xx}^{(2)} = \sum_i \mathscr{H}_{i,xx}^{(2)}\left(f_i - f_i^{eq}\right), a_{1,xy}^{(2)} = \sum_i \mathscr{H}_{i,xy}^{(2)}\left(f_i - f_i^{eq}\right) \tag{47}$$

$$a_{1,xxy}^{(3)} = 2u_x\, a_{1,xy}^{(2)} + \, u_y a_{1,xx}^{(2)} \tag{48}$$

$$a_{1,xxyy}^{(4)} = 2\left(u_x\, a_{1,xyy}^{(3)} + \, u_y\, a_{1,xxy}^{(3)}\right) - \left(c_s^2 \, + u_x^2\right)a_{1,yy}^{(2)} - \left(c_s^2 \, + u_y^2\right)a_{1xx}^{(2)} - 4u_x u_y a_{1,xy}^{(2)} \tag{49}$$

Data availability

Data will be made available on request.

References

[1] Zhang R, et al. Lattice Boltzmann approach for local reference frames. Commun Comput Phys 2011;9(5):1193–205.
[2] Li, Y., *An improved volumetric LBM boundary approach and its extension for sliding mesh simulation.* 2011.
[3] Far EK, Geier M, Krafczyk M. Simulation of rotating objects in fluids with the cumulant lattice Boltzmann model on sliding meshes. Comput Math Appl 2020;79(1):3–16.
[4] Meldi M, Vergnault E, Sagaut P. An arbitrary Lagrangian–Eulerian approach for the simulation of immersed moving solids with Lattice Boltzmann Method. J Comput Phys 2013;235:182–98.
[5] Lallemand P, Luo L-S. Lattice Boltzmann equation with Overset method for moving objects in two-dimensional flows. J Comput Phys 2020;407:109223.
[6] Bahlali M, et al. A lattice Boltzmann direct coupling overset approach for the moving boundary problem. Phys Fluids 2021;33(5):053607.
[7] Yoo H, et al. A hybrid recursive regularized lattice Boltzmann model with overset grids for rotating geometries. Phys Fluids 2021;33(5):057113.

[8] He X, Doolen GD. Lattice Boltzmann method on a curvilinear coordinate system: vortex shedding behind a circular cylinder. Phys Rev E 1997;56(1):434.
[9] He X, Doolen G. Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder. J Comput Phys 1997;134(2):306–15.
[10] Hejranfar K, Saadat MH, Taheri S. High-order weighted essentially nonoscillatory finite-difference formulation of the lattice Boltzmann method in generalized curvilinear coordinates. Phys Rev E 2017;95(2):023314.
[11] Mei R, Shyy W. On the finite difference-based lattice Boltzmann method in curvilinear coordinates. J Comput Phys 1998;143(2):426–48.
[12] Hejranfar K, Ezzatneshan E. Implementation of a high-order compact finite-difference lattice Boltzmann method in generalized curvilinear coordinates. J Comput Phys 2014;267:28–49.
[13] Karbalaei A, Hejranfar K. A Central Difference Finite Volume Lattice Boltzmann Method for Simulation of 2D Inviscid Compressible Flows on Triangular Meshes. In: ASME International Mechanical Engineering Congress and Exposition. American Society of Mechanical Engineers; 2018.
[14] Li Y, LeBoeuf EJ, Basu P. Least-squares finite-element scheme for the lattice Boltzmann method on an unstructured mesh. Phys Rev E 2005;72(4):046711.
[15] He X, Luo L-S. Theory of the lattice Boltzmann method: from the Boltzmann equation to the lattice Boltzmann equation. Phys Rev E 1997;56(6):6811.
[16] Filippova O, Hänel D. Boundary-fitting and local grid refinement for lattice-BGK models. Int J Modern Phys. C 1998;9(08):1271–9.
[17] Peng Y, et al. Application of multi-block approach in the immersed boundary–lattice Boltzmann method for viscous fluid flows. J Comput Phys 2006;218(2):460–78.
[18] Inamuro T. Lattice Boltzmann methods for moving boundary flows. Fluid Dyn Res 2012;44(2):024001.
[19] Cheng M, Hung KC. Lattice Boltzmann method on nonuniform mesh. Int J Comput Eng Sci 2004;5(02):291–302.
[20] Valero-Lara P, Jansson J. A non-uniform Staggered Cartesian grid approach for Lattice-Boltzmann method. Procedia Comput Sci 2015;51:296–305.
[21] Mahmoudi S, et al. Investigation on reactive flow through porous media by quadtree Lattice Boltzmann. J Nat Gas Sci Eng 2022;104:104674.
[22] Crouse B, et al. A LB-based approach for adaptive flow simulations. Int J Modern Phys B 2003;17(01n02):109–12.
[23] Pellerin N, Leclaire S, Reggio M. An interpolation-based lattice Boltzmann method for non-conforming orthogonal meshes. Comput Math Appl 2021;100:152–66.
[24] Yoo H, et al. Compressible lattice Boltzmann method with rotating overset grids. Phys Rev E 2023;107(4):045306.
[25] Guo Z, Zheng C, Shi B. An extrapolation method for boundary conditions in lattice Boltzmann method. Phys fluids 2002;14(6):2007–10.
[26] Latt J, Chopard B. Lattice Boltzmann method with regularized pre-collision distribution functions. Math Comput Simul 2006;72(2–6):165–8.
[27] Malaspinas O. Increasing stability and accuracy of the lattice Boltzmann scheme: recursivity and regularization. arXiv preprint; 2015. arXiv:1505.06900.
[28] Bhatnagar PL, Gross EP, Krook M. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys Rev 1954;94(3):511.
[29] Chapman S, Cowling TG. The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases. Cambridge university press; 1990.
[30] Coreixas C, et al. Recursive regularization step for high-order lattice Boltzmann methods. Phys Rev E 2017;96(3):033306.
[31] Dellar PJ. An interpretation and derivation of the lattice Boltzmann method using Strang splitting. Comput Math Appl 2013;65(2):129–41.
[32] Alciatore, D. and R. Miranda, *A winding number and point-in-polygon algorithm*. Glaxo virtual anatomy project research report, Department of Mechanical Engineering, Colorado State University, 1995.
[33] Zou Q, He X. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. Phys Fluids 1997;9(6):1591–8.
[34] Inamuro T, Yoshino M, Ogino F. A non-slip boundary condition for lattice Boltzmann simulations. Phys Fluids 1995;7(12):2928–30.
[35] Krüger T, et al. The lattice boltzmann method, 10. Springer International Publishing; 2017. p. 4–15.
[36] Taylor GI. Stability of a viscous liquid contained between two rotating cylinders. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character 1923;223(605–615):289–343.
[37] Mei R, et al. Force evaluation in the lattice Boltzmann method involving curved geometry. Phys Rev E 2002;65(4):041203.
[38] Kang S, Choi H, Lee S. Laminar flow past a rotating circular cylinder. Phys Fluids 1999;11(11):3312–21.
[39] Tang T, Ingham D. On steady flow past a rotating circular cylinder at Reynolds numbers 60 and 100. Comput Fluids 1991;19(2):217–30.
[40] Badr H, Dennis S, Young P. Steady and unsteady flow past a rotating circular cylinder at low Reynolds numbers. Comput Fluids 1989;17(4):579–609.