mathematics

MDPI

Article

A New Multi-Level Grid Multiple-Relaxation-Time Lattice Boltzmann Method with Spatial Interpolation

Zhixiang Liu [1,2], Shengyong Li [1], Jun Ruan [1], Wenbo Zhang [1,*], Liping Zhou [3,*], Dongmei Huang [4] and Jingxiang Xu [5]

[1] College of Information Technology, Shanghai Ocean University, Shanghai 201306, China
[2] East China Sea Forecast Center of State Oceanic Administration, Shanghai 200136, China
[3] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[4] College of Electrical Engineering, Shanghai University of Electric Power, Shanghai 200090, China
[5] College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China
* Correspondence: wbzhang@shou.edu.cn (W.Z.); lpzhou@shu.edu.cn (L.Z.)

Abstract: The traditional multi-level grid multiple-relaxation-time lattice Boltzmann method (MRT-LBM) requires interpolation calculations in time and space. It is a complex and computationally intensive process. By using the buffer technique, this paper proposes a new multi-level grid MRT-LBM which requires only spatial interpolation calculations. The proposed method uses a center point format to store multi-level grid information. The grid type determination in the flow field calculation domain is done using the axis aligned bounding box (AABB) triangle overlap test. According to the calculation characteristics of MRT-LBM, the buffer grid is proposed for the first time at the interface of different levels of grids, which is used to remove the temporal interpolation calculation and simplify the spatial interpolation calculation. The corresponding multi-level grid MRT-LBM algorithm is also presented for two-dimensional and three-dimensional flow field calculation problems. For the two-dimensional problem of flow around a circular cylinder, the simulation results show that a four-level grid MRT-LBM proposed in this paper can accurately obtain the aerodynamic coefficients and Strouhal number at different Reynolds numbers, and it has about 1/9 of the total number of grids as a single-level grid MRT-LBM and is 6.76 times faster. For the three-dimensional flow calculation problem, the numerical experiments of flow past a sphere are simulated to verify the numerical precision of the presented method at Reynolds numbers = 100, 200, 250, 300, and 1000. With the streamlines and velocity contours, it is demonstrated that the multi-level grid MRT-LBM can be calculated accurately even at the interface of different size grids.

Keywords: multiple-relaxation-time lattice Boltzmann method; multi-level; buffer technique; spatial interpolation

MSC: 76M25

Citation: Liu, Z.; Li, S.; Ruan, J.; Zhang, W.; Zhou, L.; Huang, D.; Xu, J. A New Multi-Level Grid Multiple-Relaxation-Time Lattice Boltzmann Method with Spatial Interpolation. *Mathematics* **2023**, *11*, 1089. https://doi.org/10.3390/math11051089

Academic Editor: Vasily Novozhilov

Received: 24 December 2022
Revised: 8 February 2023
Accepted: 15 February 2023
Published: 22 February 2023

1. Introduction

Computational fluid dynamics (CFD) is an interdisciplinary discipline between mathematics, fluid dynamics, and computers that emerged with the development of computers. CFD applies numerical methods to solve a set of partial differential equations that govern fluid flow. Numerical simulations of complex problems in fluid mechanics facilitate the qualitative and quantitative analysis of practical engineering problems. CFD is used in a variety of industries, including aerospace [1], chemical engineering [2], energy [3], and thermodynamics [4].

The lattice Boltzmann method (LBM) is an important numerical method in computational fluid dynamics. The LBM has many unique advantages, such as clear physical interpretations, being suitable for parallel operation, simple boundary treatment, and easy programming [5]. The Bhatnagar-Gross-Krook (BGK) model [6] is a widely used lattice

Boltzmann model. It uses a single relaxation coefficient instead of a matrix pattern of collision terms to regulate the velocity of particles near the equilibrium state, and this approximate treatment greatly simplifies the solution of LBM. Due to its simplicity and computational efficiency, the BGK model has been widely used in many fields, such as engineering [7,8], physics [9,10], and mechanics [11,12]. The BGK model does not take into account the anisotropy in the actual flow, and the calculation results may be numerically unstable. The generalized LBM (GLBM) [13] uses multiple relaxation times in the collision process and improves the numerical stability of the LBM by increasing the number of free parameters. So, the GLBM is also called the multi-relaxation time lattice Boltzmann equation (MRT-LBM). MRT-LBM effectively solves the BGK model's tendency to diverge in some specific situations. Lallemand and Luo [14] have presented a detailed theoretical analysis of the MRT-LBM model, and the results show that the MRT-LBM model has great advantages in terms of physical principles, parameter selection, and numerical stability. In addition, MRT-LBM transforms the collision process into moment space by linear transformation, which has better numerical stability and accuracy, and has been used in acoustics [15], thermodynamics [16,17], geology [18], and materials science [19].

The application of LBM in many fields is limited by the large computational effort and non-adjustable computational accuracy due to the uniform grid used in LBM. However, the LBM can be regarded as a special discrete form of the continuous Boltzmann equation, which allows the use of a non-uniform grid. Moreover, there are usually some regions with more drastic changes in the flow field simulation. In order to capture the flow details in these regions, the LBM needs to refine all the computational grid regions, which will increase the computational effort. Therefore, a local grid refinement method can be used to deal with these regions. Filippova and Hänel [20] proposed a local grid refinement BGK model (FH method). The FH method stores the grid information through a vertex point format scheme, and the evolution between coarse and fine grids takes a bidirectional coupling approach. Furthermore, the vertex point format has been widely adopted and further investigated in several studies [21–23]. Two formats are usually available for grid information storage: vertex point format and center point format, respectively. At the interface of coarse and fine grids, the vertices of the coarse grids and the fine grids in the vertex point format are overlapped. The method of storing grid information in center point format was proposed by Rohde et al. [24], where the information of the coarse grids and the fine grids are stored separately at the interface of coarse and fine grids. Based on the center point format structure, the local grid refinement of LBM uses temporal and spatial interpolation methods to ensure uniform information propagation between coarse and fine grids, which improves the computational accuracy [25–27]. Guo et al. [28] introduced the domain decomposition method into the LBM and constructed a domain decomposition-based LBM (DDLBM). The DDLBM divides the flow field into several non-overlapping sub-regions. Each sub-region is calculated using the conventional LBM, and the results of the entire flow field are determined by coupling the calculations of these sub-regions. The multi-block method divides the flow field into different regions and uses separate grids of different granularity for each region. At the interface of the coarse and fine grids, the continuity of physical quantities in different regions can be ensured by using temporal and spatial interpolation processes [29–31]. Adaptive mesh refinement (AMR) can effectively reduce the computational effort while decreasing the accuracy loss. A multi-level AMR method was applied to LBM [32]. The method relies on temporal and spatial interpolation and solving a constrained least squares problem to ensure conservation. Based on the octree AMR method, Liu et al. [33] proposed an immersion boundary lattice Boltzmann method (IB-LBM) solver. The solver exchanges the distribution functions between the coarse and fine grids interface through a four-point Lagrangian interpolation method. The effectiveness of the solver was verified by two-dimensional problems and three-dimensional problems.

The encryption of uniform grids using local grid refinement, domain decomposition, and the AMR method can improve the efficiency of BGK calculations and accurately capture the flow information in the drastically changing region, but it cannot change the problem

of BGK model divergence in specific cases. Since MRT-LBM has better computational stability, the design of MRT-LBM with uniform grid encryption can effectively reduce the computational effort and guarantee the computational accuracy. An early example of applying mesh refinement processing to MRT-LBM is studied by Peng et al. [34]. They used three-sample interpolation and three-point Lagrangian formulation for temporal and spatial interpolation at different levels of grid interfaces, respectively, and used a multi-block method with a multi-relaxation time collision scheme to improve efficiency and accuracy. Good results were obtained by numerical simulations of steady and unsteady flows in cylinders and wings. Arora et al. [35] constructed a discontinuous grid block MRT-LBM around a multi-block grid interface using a cubic spline interpolation scheme. By simulating three standard moving boundary problems, the results showed that the discontinuous grid block MRT-LBM has better stability than the BGK model at high Reynolds numbers. Based on this study, the correlation method for all non-equilibrium moments across the subdomain boundaries and a new method for handling buffer grids at the interface of coarse and fine grids in the MRT-LBM are presented [36]. A simulation of lid-driven cavity flow and Couette flow with moving cylinders was performed to validate the proposed method. Recently, Wu et al. [37] proposed a continuous refinement scheme for MRT-LBM, which is used to simulate rapidly changing wall regions of flow properties. Data transfer between coarse and fine grids is performed using a cubic polynomial interpolation. Turbulent channel flows at different Reynolds numbers are simulated, and the results show that the proposed continuous refinement scheme is effective in predicting the turbulent flow volumes.

The existing grid encryption processing methods [34–37] in the evolution of MRT-LBM usually require temporal and spatial interpolation calculations at different levels of the grid interface, and these temporal and spatial interpolation formulas are complex. In order to simplify the complexity of temporal and spatial interpolation formulas and reduce the computational effort. In this paper, we propose a multi-level grid MRT-LBM algorithm that requires only spatial interpolation. By setting buffers to remove the temporal interpolation step, this process effectively reduces the data transfer at different levels of the grid interfaces and simplifies the computational complexity. Meanwhile, we give a new spatial interpolation formula at different levels of the grid interfaces, which ensures the conservation of physical quantities in the two-dimensional and three-dimensional cases. In this paper, we also propose a grid generation technique applicable to the multi-level grid MRT-LBM algorithm, including grid storage structure, grid type determination, and multi-level grid generation, which makes the grid encryption of MRT-LBM more flexible.

The structure of this paper is as follows. In Section 2, the basic theory of MRT-LBM is described, as well as the MRT collision model. The multi-level generation technique suitable for MRT-LBM and the corresponding multi-level grid MRT-LBM evolution model are given in Section 3. Numerical simulation and analysis are provided in Section 4. A conclusion is presented in Section 5.

2. MRT-LBM

The MRT-LBM evolution equation [13] is:

$$f_i(\boldsymbol{x} + \boldsymbol{e}_i\delta_t, t + \delta_t) - f_i(\boldsymbol{x}, t) = \Omega_i \tag{1}$$

where f_i is the distribution function, $\boldsymbol{x}$ is the spatial position, $\boldsymbol{e} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m)^T$ is the discrete velocity, $i = 1, 2, \ldots, m$ is the speed direction of the discrete velocity model, δ_t represents the time step, t is the time, Ω_i is collision item. The vector form of the distribution function f_i is $\boldsymbol{f}(\boldsymbol{x}, t) = (f_1(\boldsymbol{x}, t), f_2(\boldsymbol{x}, t) \ldots f_m(\boldsymbol{x}, t))^T$. D_nQ_m is the discrete velocity model. n and m are denoted as the space dimension and discrete velocity numbers, respectively.

The equilibrium distribution function f_i^{eq} can be expressed as:

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

where ω_i is the weight coefficient, c_s is the lattice sound velocity, and ρ and $\boldsymbol{u}$ are the macro-density and macro-velocity, respectively.

In the D_nQ_m model, the value of the collision term Ω_i is as follows:

$$\Omega_i = [\boldsymbol{\Omega}]_i = \left\{ -\boldsymbol{M}^{-1} \cdot \boldsymbol{S} \cdot \boldsymbol{M} [\boldsymbol{f}(\boldsymbol{x}, t) - \boldsymbol{f}^{eq}(\boldsymbol{x}, t)] \right\}_i, \tag{3}$$

where $\boldsymbol{M}$ is the transformation matrix and $\boldsymbol{S} = diag(s_1, s_2, \ldots, s_m)$ is the non-negative relaxation diagonal matrix.

The evolution Equation (1) consists of two parts:

$$\text{Collision}: \ f_i^*(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) + \Omega_{\boldsymbol{i}}, \tag{4}$$

$$\text{Streaming}: \ f_i(\boldsymbol{x} + \boldsymbol{e}_i \delta_t, t + \delta_t) = f_i^*(\boldsymbol{x}, t), \tag{5}$$

where f_i^* represents the distribution function obtained after the collision.

After the streaming calculation, the macro-density ρ and macro-velocity $\boldsymbol{u}$ on the grids need to be updated:

$$\begin{array}{c} \sum_i f_i = \rho, \\ \sum_i \boldsymbol{e}_i f_i = \rho \boldsymbol{u}. \end{array} \tag{6}$$

Here, take the D3Q19 model as an example. The 19 velocity values of the discrete velocity $\boldsymbol{e}_i$ are:

$$\boldsymbol{e}_i = \begin{cases} (0,0,0), & i = 0 \\ (\pm s, 0, 0), (0, \pm s, 0), (0, 0, \pm s), & i = 1, \ldots, 6 \\ (\pm s, \pm s, 0), (\pm s, 0, \pm s), (0, \pm s, \pm s), & i = 7, \ldots, 18 \end{cases}$$

where $s = \delta_x / \delta_t$. δ_x represents the grid spacing step.

The transformation matrix $\boldsymbol{M}$ can be expressed as follows:

$$\boldsymbol{M} = \begin{Bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -30 & -11 & -11 & -11 & -11 & -11 & -11 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & -4 & -1 & -4 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & -4 & 4 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -4 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 0 & 2 & 2 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\ 0 & -4 & -4 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\ 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4 & -4 & 2 & 2 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{Bmatrix}$$

The relaxation factor $\boldsymbol{S}$ is

$$\boldsymbol{S} = diag\left(0, s_e, s_\varepsilon, 0, s_q, 0, s_q, 0, s_q, s_\nu, s_\pi, s_\nu, s_\pi, s_\nu, s_\nu, s_\nu, s_t, s_t, s_t\right)$$

The kinematic viscosity ν and ς are defined as

$$\nu = c_s^2\left(\frac{1}{s_v} - \frac{1}{2}\right)\delta_t,$$

$$\varsigma = \frac{2c_s^2}{3}\left(\frac{1}{s_v} - \frac{1}{2}\right)\delta_t,$$

the lattice sound velocity $c_s = \sqrt{1/3}$.

3. Multi-Level Grid MRT-LBM

Based on the basic concept of MRT-LBM, this section gives the multi-level grid generation technique applicable to MRT-LBM, including grid storage structure, type judgment, and a multi-level grid generation algorithm. Meanwhile, the evolutionary model of multi-level grid MRT-LBM is re-analyzed by setting buffers at the intersection of different levels of grids. Finally, we combine the multi-level grid generation technique with the MRT-LBM evolution model and propose the multi-level grid MRT-LBM algorithm capable of handling complex flows and complex geometries.

3.1. Multi-Level Grid Generation Technique

In this paper, the center point format is used as the storage format for multi-level grid information; see Figure 1. The dot in the center of each grid in Figure 1 represents the center point of that grid and is used to record the grid storage information. The blue dots represent the coarse grid center points and the green dots represent the fine grid center points. The bigger dots represent the larger size of the grid.

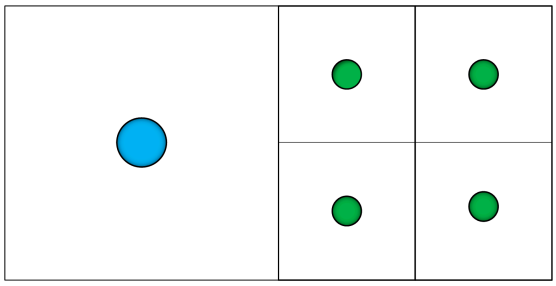

Figure 1. Center point format.

In 2D and 3D multi-level grid, the center point format will generate a quadtree and octree structure, respectively; see Figures 2 and 3.

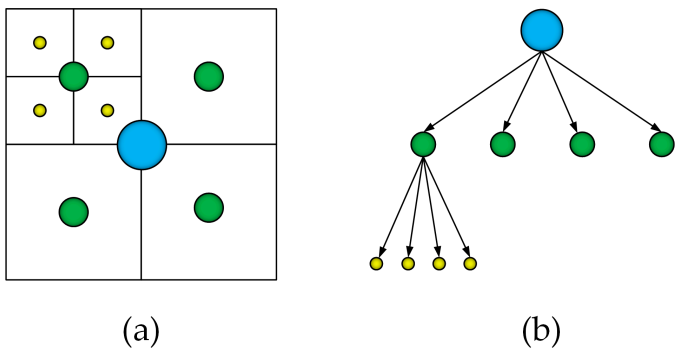

Figure 2. 2D multi-level grid storage format: (**a**) 2D multi-level grid; (**b**) quadtree storage format.

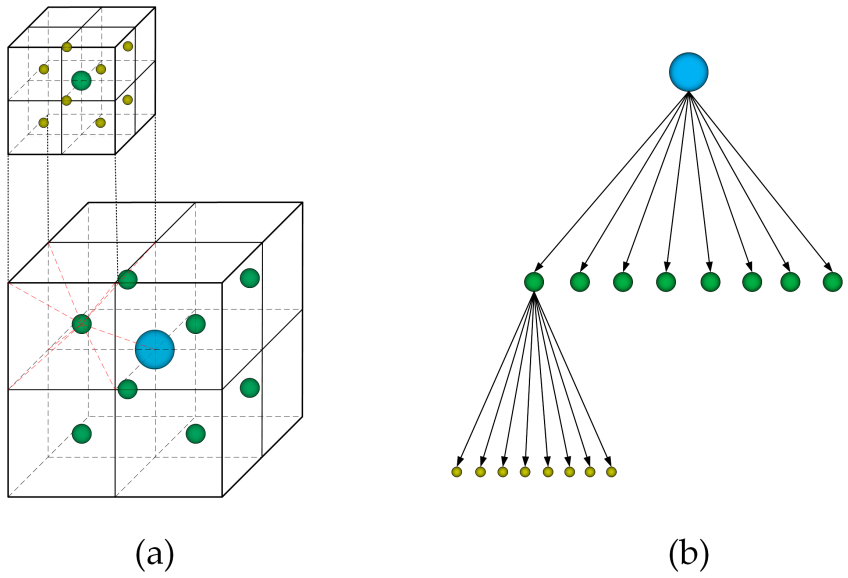

Figure 3. 3D multi-level grid storage format: (**a**) 3D multi-level grid; (**b**) octree storage format.

In a 2D multi-level grid, coarse grid point encryption will lead to four fine grid points. The fine grid size is half of the coarse grid size. As a result, a quadtree structure is formed for recording the 2D multi-level grid. Similarly, in a 3D multi-level grid, coarse grid encryption will lead to eight grid points, which eventually form an octree structure for recording the 3D multi-level grid. The quadtree or octree data structure facilitates the implementation of subsequent multi-level grid generation techniques and the storage of grid information. The multi-level grid MRT-LBM based on this data structure is also easier to design and implement.

Based on the MRT-LBM evolution details and the storage format of the center point grid, this paper designs a lattice point class to record the following information:

(I) The coordinates of the center point grid;
(II) Neighborhood number for grid points at the same size;
(III) Types of grid points;
(IV) Size of grid points;
(V) Level of grid points in a quadtree or octree structure;
(VI) Set the marker for grid point encryption. If the grid point is encrypted, the information of the corresponding sub-grid (four or eight) is recorded;
(VII) If a parent grid point exists for the current grid point, the information of the parent grid is recorded;
(VIII) If it is a boundary grid point, the distance information from the actual boundary to the grid point is recorded.

Here, the grid point types can be divided into flow field points, boundary points, and solid points; see Figure 4. The black line in Figure 4 is the curved boundary of the geometry. The boundary grid points (yellow grid in Figure 4) are the interface of the grid with the geometry. For the evolutionary calculation of MRT-LBM, the boundary grid points additionally need to record the distance from the grid center point along the direction of the discrete velocity model to the actual boundary. The solid grid points (brown grid in Figure 4) are the grid points inside the geometry. Since the solid grid points are not involved in the evolution calculation of MRT-LBM, the solid grid point information can also not be recorded. The remaining type of grid points are the flow field grid points, i.e., the internal grid points of the flow field (blue grid in Figure 4).

Based on the design of the lattice point class, the following design solutions are given for judgment on the types of 3D grid points. In the 3D case, we usually load the surface mesh file (STL file) of the geometry. A STL file is made up of triangles, and each triangle records its three vertices and normal. In the whole process for determining the

grid point type, we can first determine whether the grid point intersects with the triangle in space, i.e., the intersecting grid point is determined to be the type of boundary grid. The Fast 3D AABB-triangle-box overlap testing method [38] is used to judge whether the triangle intersects with the computational grid. By judging all computational grids, all boundary grid points can be marked and stored in the lattice point class of the boundary grid. Afterwards, by selecting one internal grid point of the geometry, we can directly divide the unmarked grid into solid grid points or flow field grid points.

The multi-level grid generation technique is presented in detail below. First, the enclosing box range and grid size for the flow field calculation are set. This enclosing box is the initial grid. The size of the initial grid is chosen appropriately, which will affect the quality of the adaptive grid and multi-level grid, and also the accuracy of the calculation of the whole flow field. After that, the initial grid is generated by isometric division according to the grid size. The initial grids generated in this way are cubic (3D) or square (2D) grids and are stored in center point format. The Algorithm 1 for the initial grid generation is shown as follows (for the 3D case):

Algorithm 1 Initial grid generation algorithm.

Input: the minimum coordinate point and the maximum coordinate point of the calculated region, size of the initial grid ($\delta_x = \delta_y = \delta_z$).
Output: flow field calculation region grid

1. Set the initial cube enclosing box based on the maximum and minimum coordinate points;
2. Uniformly divide the cube grids according to the initial grid size for the cube enclosing box;
3. The initial variable information in the lattice point class is stored in all cubic grids in the order of x-direction, then y-direction, and then z-direction:
   (a) The information of (I), (III), (IV), and (VIII) in the lattice point class can be obtained by direct calculation;
   (b) The information of (V), (VI), and (VII) in the lattice point class can be set to the corresponding initial values since the grids have not been encrypted yet;
   (c) The information of (II) in the lattice point class involves information about the neighbors of the grid points. Since the grid is in a three-dimensional space, each grid has 6 surface grid neighbors, 12 edge grid neighbors, and a grid with 8 vertex grid neighbors. The label of the grid has been obtained during initialization, and the 6 surface grid neighbors of a certain grid can be found directly by the number. If the grid is located at the boundary, there may not be a certain area grid neighbor, so the area grid neighbors of the grid point to empty. Suppose the grid number is q, and the number of grids divided on each coordinate axis is k. Then the upper, lower, left, right, front, and back neighbor grids of the grid can be determined according to the following grid numbers (when a neighbor of the grid does not exist, this indicates that the grid is located at the boundary of the calculation area):
      (1) upper neighbor grid number: $q + 1$, it does not exist when $((q + 1) mod\ k) = 0$;
      (2) lower neighbor grid number: $q - 1$, it does not exist when $((q - 1) mod\ k) = 0$;
      (3) left neighbor grid number: $q - k$, it does not exist when $0 \leq \left(q\ mod\ k^2\right) \leq k - 1\ or\ q < k$;
      (4) right neighbor grid number: $q + k$ it does not exist when $k^2 - k \leq \left(q \bmod k^2\right) \leq k^2 - 1$ or $q < k$;
      (5) front neighbor grid number: $q + k^2$ it does not exist when $i + k^2 \geq k^3$;
      (6) back neighbor grid number: $q - k^2$ it does not exist when $q < k^2$.

The LBM uses a discrete velocity model with 15 or 19 directions in 3D space. Considering the space size of the storage, only six faces (see Figure 5) of neighbor information in the lattice class are recorded here. Neighborhood information of other directions can be quickly obtained from these six faces.

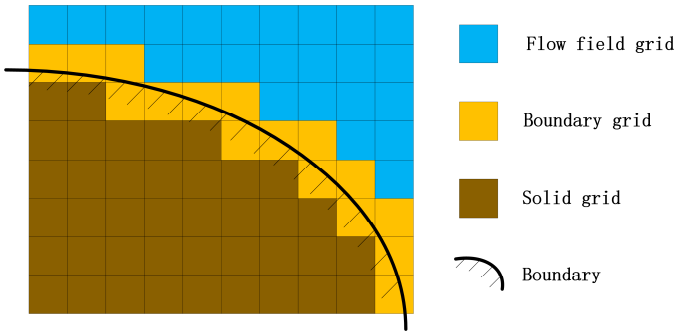

Figure 4. Schematic diagram of grid types.

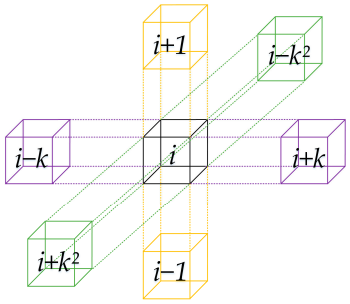

Figure 5. Schematic diagram of the neighboring grid numbers of the six faces of the grid in 3D case.

In addition to specifying the calculation area of the entire flow field, it is often necessary to formulate some areas to set smaller grids, as shown in Figure 6. At this time, on the basis of the initial grid (the first level grid), the lattice of the second level grid area needs to be refined and recorded in the lattice class. Following the generation of the second level grid, the third level grid is generated using the similar process. And so on, until the grid refinement of all specified calculation areas is completed. The solution of problems such as boundary levels in flow calculations requires very fine grids near the geometry. Therefore, in addition to meshing different areas, it is also necessary to further refine the boundary surface of the geometry for complex geometry. Pushing a certain number of flow field grids outward from the boundary grids is also refined. The grid refinement in this part is still performed by using 1/2 the size of the surrounding grids, and multiple refinements can be performed to meet the computational accuracy requirements. The specific multi-level grid generation in 3D can be described as Algorithm 2.

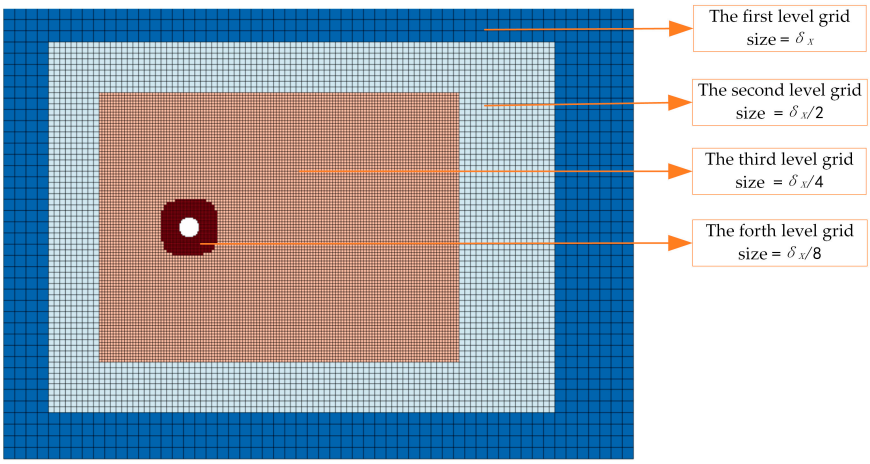

Figure 6. Schematic diagram of grid in different areas (flow around a circular cylinder).

Algorithm 2 Multi-level grid generation algorithm.

Input: total number of grid levels: N, initial number of grid level $Ni = 2$, minimum and maximum coordinate points of each grid level, size of each grid level (1/2 the size of the previous grid level), the number of grid points to be encrypted by extrapolation from the boundary grid: M.
Output: flow field calculation region grid

1. Determine the Ni level grid range based on the maximum and minimum coordinate points of the Ni level grid;
2. Encrypt the Ni-−1 level grid within the Ni level grid range according to the Ni level grid size and the $Ni-1$ level grid lattice class in the 5th information: refine = 1, while recording 8 sub-grid information;
3. For the Ni level grid, the information in the lattice class is recorded in the Ni level cubic grid in the order of x-direction, then y-direction, and then z-direction:
   (a) The information of (I), (III), (IV), and (VIII) in the lattice point class can be obtained by direct calculation;
   (b) The information of (II) in the lattice class is calculated in the same way as in Algorithm 1;
   (c) The information of (V) and (VII) in the lattice point class is recorded separately for each Ni level grid and the parent grid number according to the actual situation of encryption;
   (d) The information of (VI) in the lattice point class is set to the corresponding initial value since the Ni level grids have not been encrypted yet.
4. If $Ni < N$, $Ni = Ni + 1$, repeat steps 1, 2, 3. Else, if $Ni = N$, execute step 6;
5. Encrypt the boundary grids and M grids outward from the boundary grids according to the grid size equal to 1/2 of the current grid size;
6. Record the information in the lattice class in the N level cube grid obtained after the encryption process of the boundary grids and M grids outward from the boundary grids:
   (a) The information of (I), (III), (IV), and (VIII) in the lattice point class can be obtained by direct calculation;
   (b) The information of (II) in the lattice class is calculated in the same way as in Algorithm 1;
   (c) The information of (V) and (VIII) in the lattice point class is recorded separately for each N level grid and the parent grid number according to the actual situation of encryption;
   (d) The information of (VI) in the lattice point class is set to the corresponding initial value since the N level grids have not been encrypted yet.

3.2. Multi-Level Grid MRT-LBM Evolution Model

In this section, the evolution model of multi-level grid MRT-LBM is presented based on the proposed multi-level grid generation technique. The discrete velocity model adopts D2Q9 in 2D and D3Q19 in 3D, respectively. The MRT-LBM evolution process of multi-level grid is actually the superposition of two-level grid evolution process. In order to facilitate the presentation, we describe the two-level grid MRT-LBM evolution process in 2D case. The multi-level grid MRT-LBM evolution process in 3D case is similar. Here, the two-level grids are marked as coarse and fine grids, represented by C and F, respectively. The distribution function on the coarse grid is denoted as: f_i^C, and the distribution function on the fine grid is denoted as: f_i^F. The corresponding MRT-LBM evolution equations are:

Coarse grid:

$$f_i^{*C} = f_i^{C,eq} + \left[-\boldsymbol{M}^{-1}\cdot\left(\boldsymbol{I}-\boldsymbol{S}^C\right)\cdot\boldsymbol{M}f^{C,\boldsymbol{neq}}\right]_i. \tag{7}$$

where $\boldsymbol{I}$ represents the identity matrix.

Fine grid:

$$f_i^{*F} = f_i^{Ceq} + \left[-\boldsymbol{M}^{-1}\cdot\left(\boldsymbol{I}-\boldsymbol{S}^C\right)\cdot\boldsymbol{M}\boldsymbol{f}^{\boldsymbol{F,neq}}\right]_i. \tag{8}$$

The size of the coarse grid is denoted as δ_x^C, the size of the fine grid is denoted as δ_x^F, and the ratio of the coarse and fine grid sizes is assumed to be r (r is an integer), so it can be derived:

$$\delta_t^F = \frac{1}{r}\delta_t^C. \tag{9}$$

To ensure the continuity of density and velocity at the interface of coarse grids and fine grids, it is necessary to satisfy:

$$\boldsymbol{f}^{eq} = \boldsymbol{f}^{C,eq} = \boldsymbol{f}^{F,eq}. \tag{10}$$

According to the equal shear stresses at the coarse and fine meshes, it can be obtained:

$$\frac{\boldsymbol{M}\boldsymbol{f}^{C,neq}\boldsymbol{S}^C}{\delta_t^C} = \frac{\boldsymbol{M}\boldsymbol{f}^{F,neq}\boldsymbol{S}^F}{\delta_t^F}. \tag{11}$$

Combining Equations (7)–(11), the streaming equation between the coarse and fine grid distribution functions can be computed:

$$f_i^{*C} = f_i^{eq} + r\left[\frac{\left(\boldsymbol{I}-\boldsymbol{S}^C\right)\boldsymbol{S}^F}{\left(\boldsymbol{I}-\boldsymbol{S}^F\right)\boldsymbol{S}^C}\boldsymbol{M}\left(\widetilde{\boldsymbol{f}}^C - \boldsymbol{f}^{eq}\right)\right]_i, \tag{12}$$

$$f_i^{*F} = f_i^{eq} + \frac{1}{r}\left[\frac{\left(\boldsymbol{I}-\boldsymbol{S}^C\right)\boldsymbol{S}^F}{\left(\boldsymbol{I}-\boldsymbol{S}^F\right)\boldsymbol{S}^C}\boldsymbol{M}\left(\widetilde{\boldsymbol{f}}^F - \boldsymbol{f}^{eq}\right)\right]_i, \tag{13}$$

where $\widetilde{\boldsymbol{f}}^C = \left(\widetilde{f}_1^C, \widetilde{f}_2^C \ldots \widetilde{f}_m^C\right)^T$ represents the distribution function that is spatially interpolated from the fine grid to the coarse grid, $\widetilde{\boldsymbol{f}}^F = \left(\widetilde{f}_1^F, \widetilde{f}_2^F \ldots \widetilde{f}_m^F\right)^T$ represents the distribution function that is spatially interpolated from the coarse grid to the fine grid.

A schematic diagram of the interface of the coarse and fine grids is given in Figure 7. The blue dots represent the coarse grids, the green dots represent the fine grids, and the interface of the coarse and fine grids is in the dashed box. In the interface area, the distribution function on the coarse grids need to be assigned to the fine grids by interpolation format. Similarly, the distribution functions on the fine grid need to be assigned to the coarse grid by means of a central interpolation format.

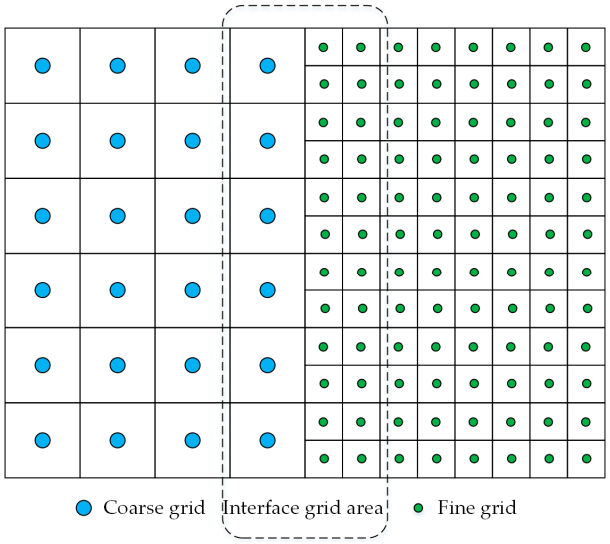

Figure 7. Schematic diagram of the interface of the coarse and fine grids in 2D.

The following is the detailed description of how to solve distribution functions $\widetilde{f}_i^C$ on coarse grid and distribution functions $\widetilde{f}_i^F$ on fine grid in 2D and 3D cases by spatial interpolation.

(I) Spatial interpolation format of coarse and fine grids in 2D.

In Figure 8, $C_i(i = 0, 1, \ldots, 8)$ is the coarse grid and $F_i(i = 0, 1, \ldots, 3)$ is the fine grid. The distribution function of the coarse grid C_0 is obtained from the four fine grids F_0, F_1, F_2, and F_3 through the center interpolation format:

$$\widetilde{f}_i^{C_0} = \frac{f_i^{F_0} + f_i^{F_1} + f_i^{F_2} + f_i^{F_3}}{4}, \tag{14}$$

where $i = 0, 1, \ldots, 8$ are the discrete velocity directions of the D2Q9.

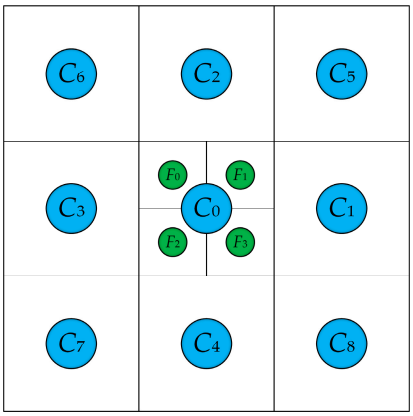

Figure 8. Schematic diagram of the coarse and fine grids in 2D.

The distribution functions of the fine grids F_0, F_1, F_2, and F_3 are obtained by interpolating the distribution functions on the nine coarse grids $C_i(i = 0, 1, \ldots, 8)$. The specific format is as follows:

$$\widetilde{f}_i^{F_0} = \frac{32f_i^{C_0} + f_i^{C_1} + 11f_i^{C_2} + 11f_i^{C_3} + f_i^{C_4} + f_i^{C_5} + 6f_i^{C_6} + f_i^{C_7}}{64}, \tag{15}$$

$$\widetilde{f}_i^{F_1} = \frac{32f_i^{C_0} + 11f_i^{C_1} + 11f_i^{C_2} + f_i^{C_3} + f_i^{C_4} + 6f_i^{C_5} + f_i^{C_6} + f_i^{C_8}}{64}, \tag{16}$$

$$\widetilde{f}_i^{F_2} = \frac{32f_i^{C_0} + f_i^{C_1} + f_i^{C_2} + 11f_i^{C_3} + 11f_i^{C_4} + f_i^{C_6} + 6f_i^{C_7} + f_i^{C_8}}{64}, \tag{17}$$

$$\widetilde{f}_i^{F_3} = \frac{32f_i^{C_0} + 11f_i^{C_1} + f_i^{C_2} + f_i^{C_3} + 11f_i^{C_4} + f_i^{C_5} + f_i^{C_7} + 6f_i^{C_8}}{64}, \tag{18}$$

where $i = 0, 1, \ldots, 8$ are the discrete velocity directions of the D2Q9 model.

(II) Spatial interpolation format of coarse and fine grids in 3D.

In Figure 9, the coarse grids are C_i, $(i = 0, 2, \ldots, 26)$, and $F_i(i = 0, 1, \ldots, 7)$ is the eight grid points refined by the coarse grid point C_0. The distribution function of the coarse grid C_0 is obtained from the eight grids $F_i(i = 0, 1, \ldots, 7)$ through the center interpolation format:

$$\widetilde{f}_i^{C_0} = \frac{f_i^{F_0} + f_i^{F_1} + f_i^{F_2} + f_i^{F_3} + f_i^{F_4} + f_i^{F_5} + f_i^{F_6} + f_i^{F_7}}{8}, \tag{19}$$

where $i = 0, 1, \ldots, 18$ are the discrete velocity directions of the D3Q19.

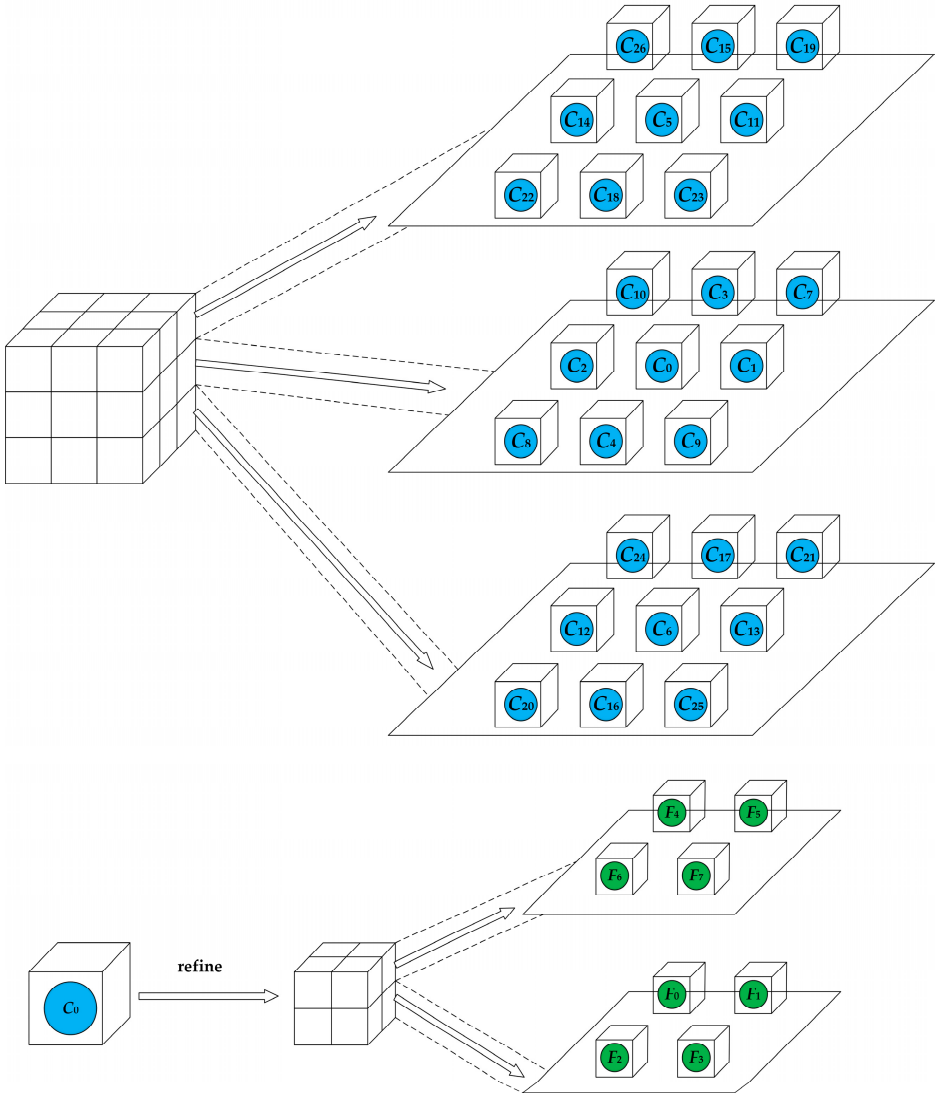

Figure 9. Schematic diagram of the coarse and fine grids in 3D.

The distribution functions of eight grids $F_i(i = 0, 1, \ldots, 7)$ are obtained by interpolating the distribution functions on the 27 coarse grids C_i (i = 0, 1, 2, . . . , 26). The specific format is as follows:

$$\begin{aligned}\widetilde{f}_i^{F_0} = \tfrac{1}{512}\Big(&168f_i^{C_0} + 6f_i^{C_1} + 58f_i^{C_2} + 58f_i^{C_3} + 6f_i^{C_4} + 6f_i^{C_5} + 58f_i^{C_6} + 7f_i^{C_7} + 5f_i^{C_8} + f_i^{C_9} + 29f_i^{C_{10}} + f_i^{C_{11}} + 29f_i^{C_{12}} + \\ &5f_i^{C_{13}} + 5f_i^{C_{14}} + 5f_i^{C_{15}} + 5f_i^{C_{16}} + 29f_i^{C_{17}} + f_i^{C_{18}} + f_i^{C_{19}} + 4f_i^{C_{20}} + 4f_i^{C_{21}} + f_i^{C_{22}} + 17f_i^{C_{24}} + f_i^{C_{25}} + 4f_i^{C_{26}}\Big),\end{aligned} \tag{20}$$

$$\begin{aligned}\widetilde{f}_i^{F_1} = \tfrac{1}{512}\Big(&168f_i^{C_0} + 58f_i^{C_1} + 6f_i^{C_2} + 58f_i^{C_3} + 6f_i^{C_4} + 6f_i^{C_5} + 58f_i^{C_6} + 29f_i^{C_7} + f_i^{C_8} + 5f_i^{C_9} + 5f_i^{C_{10}} + 5f_i^{C_{11}} + 5f_i^{C_{12}} + \\ &29f_i^{C_{13}} + f_i^{C_{14}} + 5f_i^{C_{15}} + 5f_i^{C_{16}} + 29f_i^{C_{17}} + f_i^{C_{18}} + 4f_i^{C_{19}} + f_i^{C_{20}} + 17f_i^{C_{21}} + f_i^{C_{28}} + 4f_i^{C_{24}} + 4f_i^{C_{25}} + f_i^{C_{26}}\Big),\end{aligned} \tag{21}$$

$$\begin{aligned}\widetilde{f}_i^{F_2} = \tfrac{1}{512}\Big(&168f_i^{C_0} + 6f_i^{C_1} + 58f_i^{C_2} + 6f_i^{C_3} + 58f_i^{C_4} + 6f_i^{C_5} + 58f_i^{C_6} + f_i^{C_7} + 29f_i^{C_8} + 5f_i^{C_9} + 5f_i^{C_{10}} + f_i^{C_{11}} + 29f_i^{C_{12}} + \\ &5f_i^{C_{13}} + 5f_i^{C_{14}} + f_i^{C_{15}} + 29f_i^{C_{16}} + 5f_i^{C_{17}} + 5f_i^{C_{18}} + 17f_i^{C_{20}} + f_i^{C_{21}} + 4f_i^{C_{22}} + f_i^{C_{23}} + 4f_i^{C_{24}} + 4f_i^{C_{25}} + f_i^{C_{26}}\Big),\end{aligned} \tag{22}$$

$$\begin{aligned}\widetilde{f}_i^{F_3} = \tfrac{1}{512}\Big(&168f_i^{C_0} + 58f_i^{C_1} + 6f_i^{C_2} + 6f_i^{C_3} + 58f_i^{C_4} + 6f_i^{C_5} + 58f_i^{C_6} + 5f_i^{C_7} + 5f_i^{C_8} + 29f_i^{C_9} + f_i^{C_{10}} + 5f_i^{C_{11}} + 5f_i^{C_{12}} + \\ &29f_i^{C_{13}} + f_i^{C_{14}} + f_i^{C_{15}} + 29f_i^{C_{16}} + 5f_i^{C_{17}} + 5f_i^{C_{18}} + f_i^{C_{19}} + 4f_i^{C_{20}} + 4f_i^{C_{21}} + f_i^{C_{22}} + 4f_i^{C_{23}} + f_i^{C_{24}} + 17f_i^{C_{25}}\Big),\end{aligned} \tag{23}$$

$$\tilde{f}_i^{F_4} = \frac{1}{512}\Big(168f_i^{C_0} + 6f_i^{C_1} + 58f_i^{C_2} + 58f_i^{C_3} + 6f_i^{C_4} + 58f_i^{C_5} + 6f_i^{C_6} + 5f_i^{C_7} + 5f_i^{C_8} + f_i^{C_9} + 29f_i^{C_{10}} + 5f_i^{C_{11}} + 5f_i^{C_{12}} + f_i^{C_{13}} + 29f_i^{C_{14}} + 29f_i^{C_{15}} + f_i^{C_{16}} + 5f_i^{C_{17}} + 5f_i^{C_{18}} + 4f_i^{C_{19}} + f_i^{C_{20}} + f_i^{C_{21}} + 4f_i^{C_{22}} + f_i^{C_{23}} + 4f_i^{C_{24}} + 17f_i^{C_{26}}\Big), \tag{24}$$

$$\tilde{f}_i^{F_5} = \frac{1}{512}\Big(168f_i^{C_0} + 58f_i^{C_1} + 6f_i^{C_2} + 58f_i^{C_3} + 6f_i^{C_4} + 58f_i^{C_5} + 6f_i^{C_6} + 29f_i^{C_7} + f_i^{C_8} + 5f_i^{C_9} + 5f_i^{C_{10}} + 29f_i^{C_{11}} + f_i^{C_{12}} + 5f_i^{C_{13}} + 5f_i^{C_{14}} + 29f_i^{C_{15}} + f_i^{C_{16}} + 5f_i^{C_{17}} + 5f_i^{C_{18}} + 17f_i^{C_{19}} + 4f_i^{C_{21}} + f_i^{C_{22}} + 4f_i^{C_{23}} + f_i^{C_{24}} + f_i^{C_{25}} + 4f_i^{C_{26}}\Big), \tag{25}$$

$$\tilde{f}_i^{F_6} = \frac{1}{512}\Big(168f_i^{C_0} + 6f_i^{C_1} + 58f_i^{C_2} + 6f_i^{C_3} + 58f_i^{C_4} + 58f_i^{C_5} + 6f_i^{C_6} + f_i^{C_7} + 29f_i^{C_8} + 5f_i^{C_9} + 5f_i^{C_{10}} + 5f_i^{C_{11}} + 5f_i^{C_{12}} + f_i^{C_{13}} + 29f_i^{C_{14}} + 5f_i^{C_{15}} + 5f_i^{C_{16}} + f_i^{C_{17}} + 29f_i^{C_{18}} + f_i^{C_{19}} + 4f_i^{C_{20}} + 17f_i^{C_{22}} + 4f_i^{C_{23}} + f_i^{C_{24}} + f_i^{C_{25}} + 4f_i^{C_{26}}\Big), \tag{26}$$

$$\tilde{f}_i^{F_7} = \frac{1}{512}\Big(168f_i^{C_0} + 58f_i^{C_1} + 6f_i^{C_2} + 6f_i^{C_3} + 58f_i^{C_4} + 58f_i^{C_5} + 6f_i^{C_6} + 5f_i^{C_7} + 5f_i^{C_8} + 29f_i^{C_9} + f_i^{C_{10}} + 29f_i^{C_{11}} + f_i^{C_{12}} + 5f_i^{C_{13}} + 5f_i^{C_{14}} + 5f_i^{C_{15}} + 5f_i^{C_{16}} + f_i^{C_{17}} + 29f_i^{C_{18}} + 4f_i^{C_{19}} + f_i^{C_{20}} + f_i^{C_{21}} + 4f_i^{C_{22}} + 17f_i^{C_{23}} + 4f_i^{C_{25}} + f_i^{C_{26}}\Big), \tag{27}$$

where $i = 0, 1, \ldots, 18$ are the discrete velocity directions of the D3Q19.

3.3. Multi-Level Grid MRT-LBM Algorithm

In this paper, a new multi-level grid MRT-LBM algorithm is constructed. The algorithm adopts the setting of the buffer, which removes the temporal interpolation and simplifies the data transfer on the interface of different size grids. It provides a convenient implementation of the algorithm. Considering the 2D case (Figure 7), the following setting of the buffer is introduced, as shown in Figure 10.

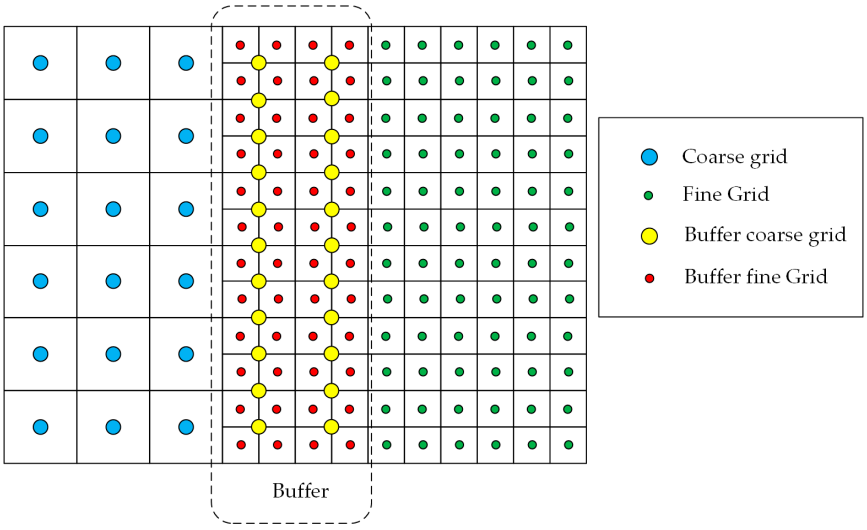

Figure 10. Buffer area for coarse and fine grid.

The blue and green dots in Figure 10 are the center points of the coarse grids and the fine grids, respectively. The grid points in the dotted lines are the grid points of the buffer. The yellow dots are the center points of the coarse grids in the buffer and the red dots are the center points of the fine grids in the buffer. In Figure 10, the grid points in the buffer are recorded in two columns of coarse grid points (yellow) and four columns of fine grid points (red) at the interface of coarse grids and fine grids.

Assuming one collision and streaming process of MRT-LBM on the coarse grid ($t^C = \delta_t$), in order to ensure that the moments of the coarse and fine grids are the same at the time of calculation, then the fine grid needs to perform r times collision and streaming process of MRT-LBM before the moment value is $t^F = \delta_t$. Consider the case of $r = 2$; thus, the coarse grid performs one collision and streaming process, and the fine grid needs to perform two collision and streaming processes to make $t^C = t^F$.

The following is a detailed description of the MRT-LBM calculation process for the coarse and fine grids in the buffer. After the grid initialization is completed, the first collision and streaming ($t^C = \delta_t$) is performed on the coarse grid by Equation (7). In Figure 10, the values of the rightmost column in the buffer with coarse grids will be incorrect after the collision and streaming process, because there are no coarse grids at the right side of the column to support the calculation. Additionally, the values of the leftmost two columns in the buffer with fine grids will be incorrect after performing twice collision and streaming ($t^F = \delta_t$) by Equation (8), because there are no fine grids at the left side of the column. In Figure 11, the result for column c1 of the coarse grid in the buffer is correct, while the result for column c2 is incorrect. The results for columns d1 and d2 of the fine grid in the buffer are incorrect, and the results for columns d3 and d4 are correct.

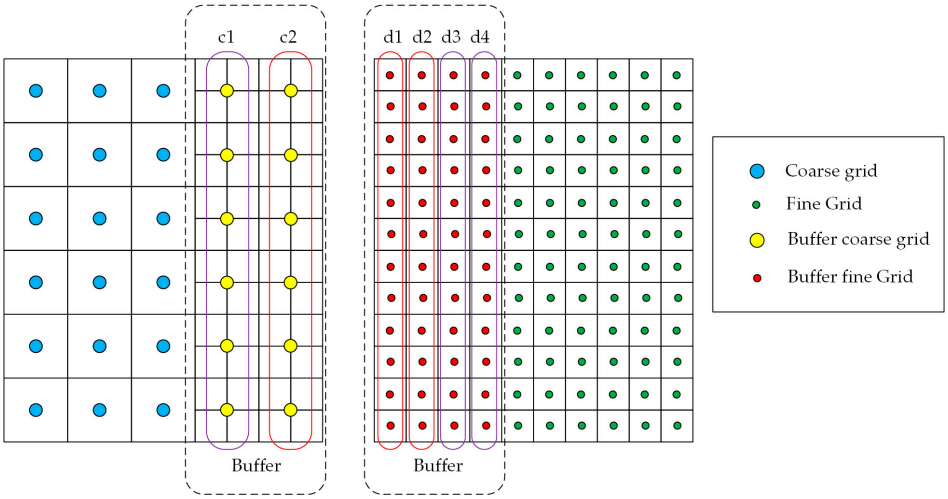

Figure 11. The description of the calculation process of the coarse grid and fine grid in the buffer.

After that, the correct distribution function values in column c2 of the coarse grid and the correct distribution function values in columns f1 and f2 of the fine grid can be calculated by spatial interpolation since the current coarse and fine grids are in the same time. The specific data transfer schematic is shown in Figure 12. The yellow triangle in Figure 12 represents the distribution function values on the coarse grid points corresponding to the fine grid region of the buffer calculated by the central interpolation Equation (14), and then the correct distribution function values are transferred to the coarse grid distribution function values represented by the yellow dots in column c2. In this way, the values of coarse grid distribution function in columns c1 and c2 are the correct distribution function values. Similarly, the red triangle in column c1 of the coarse grid in the buffer represents the distribution function values corresponding to the fine grid calculated by the interpolation Equations (15)–(18), which are then transferred to the fine grid distribution function values represented by the red dots in columns d1 and d2. In this way, all the coarse and fine grids are at time δ_t, and the distribution functions on all grid points are correct. After that, the calculation of the flow field information can be performed, and then the next time step can be started.

Based on the above analysis, setting the buffer can remove the temporal interpolation calculation and effectively reduce the spatial interpolation calculation, and also reduce the number of data transfers between the coarse and fine grid interfaces. In order to clearly describe the evolution of the multi-level grid MRT-LBM, a schematic diagram of the two-level grid MRT-LBM calculation is given in Figure 13.

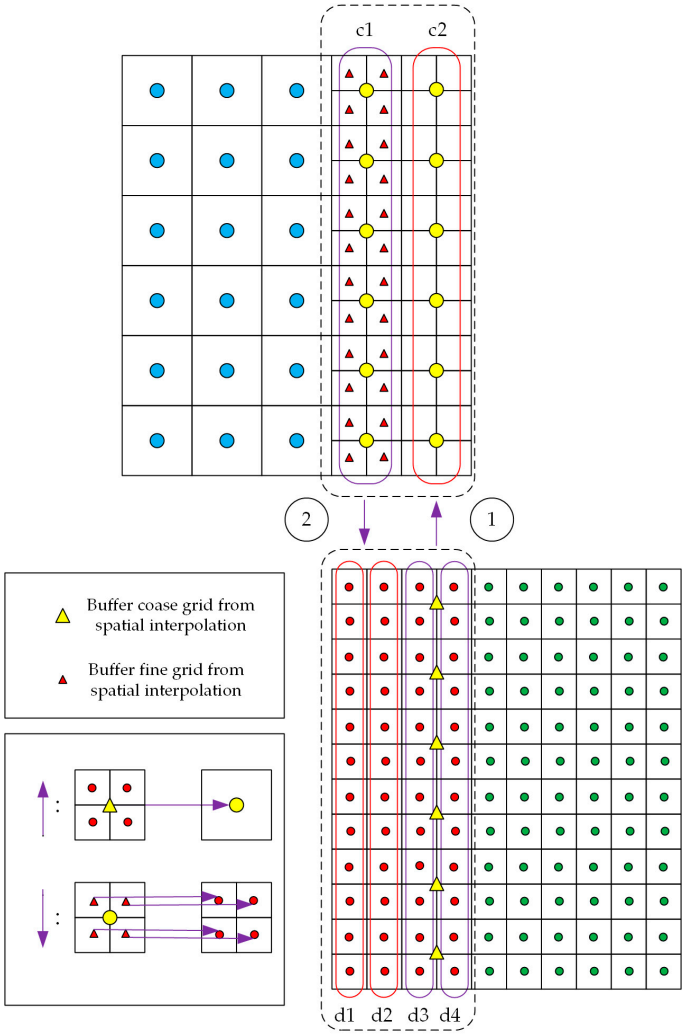

Figure 12. Data transfer between coarse and fine grids in the buffer.

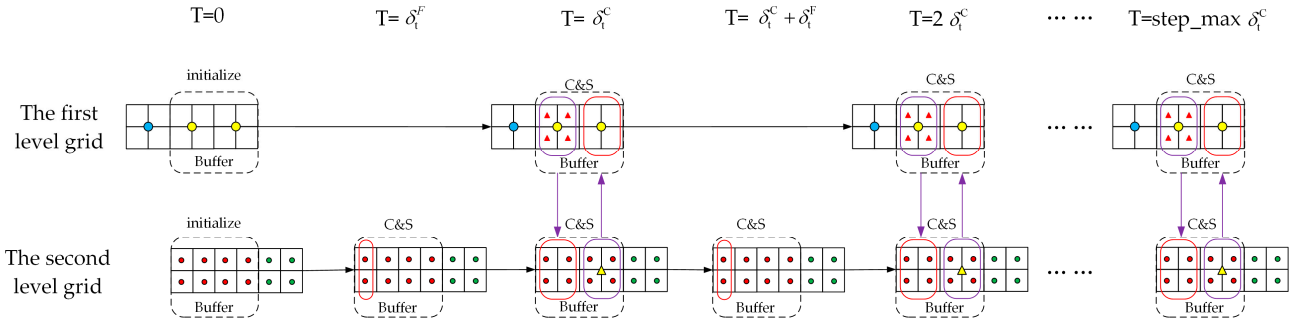

Figure 13. Schematic diagram of MRT-LBM calculation of two-level grid.

In Figure 13, "C&S" indicates the collision and streaming, "T" indicates the evolution time, and step_max indicates the total number of "C&S" in the first level of the grid. The red solid boxes indicate incorrect values, and the purple solid boxes indicate correct values obtained by spatial interpolation. The downward arrows indicate the distribution function values of the corresponding grid points on the fine grid obtained from the coarse grid by spatial interpolation, and the upward arrows indicate the distribution function values of the corresponding grid points on the coarse grid obtained from the fine grid by spatial interpolation.

According to the analysis of the two-level grid MRT-LBM calculation in Figure 13, the computational process of the multi-level grid MRT-LBM proposed in this paper is clear and easy to be implemented. Since the evolution process on each grid point is the same, the multi-level grid MRT-LBM recursive evolution process is as follows, as shown in Algorithm 3.

Algorithm 3 Multi-level grid MRT-LBM algorithm.

Input: The level of the grid is at: level L
Output: The results of evolution

```
Do step = 1, step_max
EvolutionStepofGrid (level L)
EndDo
Function EvolutionStepofGrid (level L)
  Collision (L)                                   //MRT-LBM collision process
  If (L ≠ last level)
    EvolutionStepofGrid (L + 1);                  //recursive call EvolutionStepofGrid
  end if
  Streaming (L);                                  //MRT-LBM streaming process
  If (L = first level)
    return;                                       //end function EvolutionStepofGridPoint
  end if
  Collision (L)                                   //MRT-LBM collision process
  If (L ≠ last level)
  EvolutionStepofGrid (L + 1);                    //recursive call EvolutionStepofGrid
  Endif
  Streaming (L);                                  //MRT-LBM streaming process
  Interpolation (L, L + 1);                       //transfer buffer data between L and L+1
end
```

In Algorithm 3, all grids in the first level execute the recursive function EvolutionStepofGrid() once to complete one time step from t to $t + \delta_t$ for all level grids. Until step = step_max, the overall evolution of Multi-level MRT-LBM is finished. A similar calculation of multi-level grid MRT-LBM for the 3D case can also be performed by setting the buffer. The steps are the same as for the 2D case, the difference is the choice of the spatial interpolation equations. The interpolation equations for the 3D calculation are given in Equations (19)–(27).

3.4. Analysis of the Computational Complexity

The spatial interpolation equation proposed in this paper is shown in Equations (14)–(27). The influence of surrounding points on the current grid points can be taken into account in the two-dimensional and three-dimensional case, respectively. The application of linear interpolation schemes has a high degree of localization compared to quadratic or cubic interpolation [34,37].

The multi-level grid MRT-LBM algorithm based on spatial interpolation proposed in this paper can also further improve the computational efficiency compared with the conventional temporal and spatial interpolation algorithm. The computational complexity analysis of the proposed algorithm at the intersection of different levels of the grid is given below.

For the convenience of description, we define some variables as follows. T_{sp} is the computational amount of spatial interpolation on each grid. T_{ti} is the computational amount of temporal interpolation on each grid. N_{buf} is the number of grids at the intersection of different levels of grids (i.e., the number of buffer grids).

In the two-dimensional case, according to Figure 10, the computational volume of the conventional temporal and spatial interpolation method can be expressed as follows:

$$T_{con} = 5T_{sp}N_{buf} + 3T_{ti}N_{buf} \tag{28}$$

The computational amount of the proposed method with spatial interpolation in this paper can be expressed as follows:

$$T_{new} = 5T_{sp}N_{buf} \tag{29}$$

From the analysis of computational amount, the proposed method in this paper removes the temporal interpolation at the intersection of multi-level grids and reduces the computational amount. The ratio R of the computational volume of the proposed method with spatial interpolation to that of the conventional temporal and spatial interpolation method is as follows:

$$R = \frac{T_{new}}{T_{con}} = \frac{5T_{sp}}{5T_{sp} + 3T_{ti}} \tag{30}$$

Generally, the computational amount of spatial interpolation and temporal interpolation is approximated, i.e., $T_{sp} = T_{ti}$.The value of the ratio R is 62.5%.

The multi-level grid MRT-LBM algorithm proposed in this paper is applicable in parallel. In the parallel algorithm design, it is usually necessary to set buffers at the grid interface of each level to achieve MPI communication. Therefore, the parallel algorithm design can be based on our algorithm.

4. Numerical Experiments

To further verify the accuracy and feasibility of the multi-level grid MRT-LBM proposed in this paper. In this section, numerical simulations are performed for the flow around a circular cylinder in the 2D case and the flow past a sphere in the 3D case.

4.1. Flow around a Circular Cylinder in 2D Case

The condition of the flow around a circular cylinder problem is: $28D \times 20D$ rectangular area is used. The coordinates of the center of the cylinder are $(8D, 10D)$. The boundary condition adopts the YMS format [39]. Reynolds number $Re = DU/V$, where U is the flow velocity ($U = 0.1$), is the diameter of cylinder, and V is the coefficient of viscosity. As shown in Figure 14, a four-level grid is generated based on the multi-level grid generation technique proposed in this paper. The first three levels of the grid are the enclosing boxes, and the range of these three levels of enclosing boxes is determined by the maximum and minimum coordinate points of the rectangular enclosing boxes at each level of input. The last level of the grid is obtained by extrapolating 6 grids from the boundary grid. The size of the first level grid is $1/4D$, and the grid size of the other levels is half the size of the previous level.

The accuracy of the multi-level grid MRT-LBM is verified by drag coefficient C_D, lift coefficient C_L and Strouhal number St.

The drag coefficient C_D can be expressed as:

$$C_D = \frac{F_D}{\frac{1}{2}\rho U^2 D}, \tag{31}$$

where ρ is the density of the fluid and F_D is the drag force,

The lift coefficient C_L can be expressed as:

$$C_L = \frac{F_L}{\frac{1}{2}\rho U^2 D}, \tag{32}$$

where F_L is the drag force.

The Strouhal number is usually calculated for unsteady flows. The parameters St are expressed as:

$$St = \frac{f_q D}{U}, \tag{33}$$

where f_q is the vortex shedding frequency.

The vortex structure within the wake is stable and symmetrical at Re < 47. Otherwise, the vortex begins to shed off the cylinder periodically. The results of the numerical simulation are analyzed below. The streamline for Reynolds numbers 10, 20, and 40 are shown in Figure 15. The streamlines of Reynolds numbers 100 and 200 are shown in Figure 16, and it can be seen that vortex shedding has occurred.

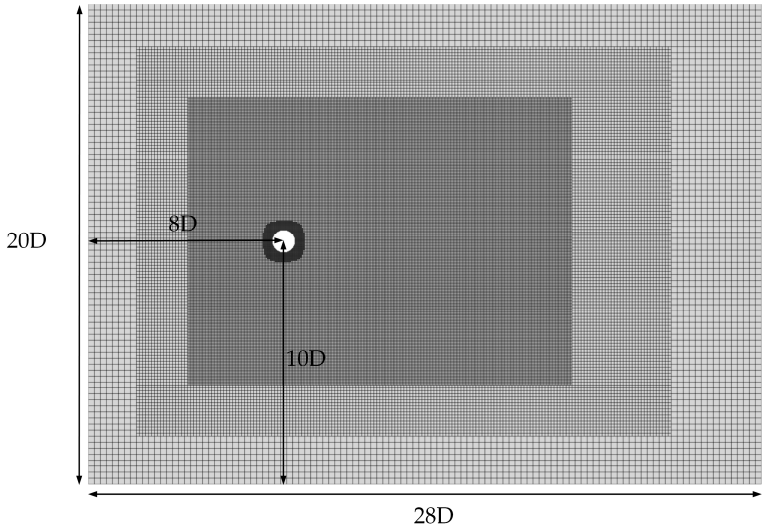

Figure 14. Schematic diagram of the four-level grid around the cylinder in 2D case.

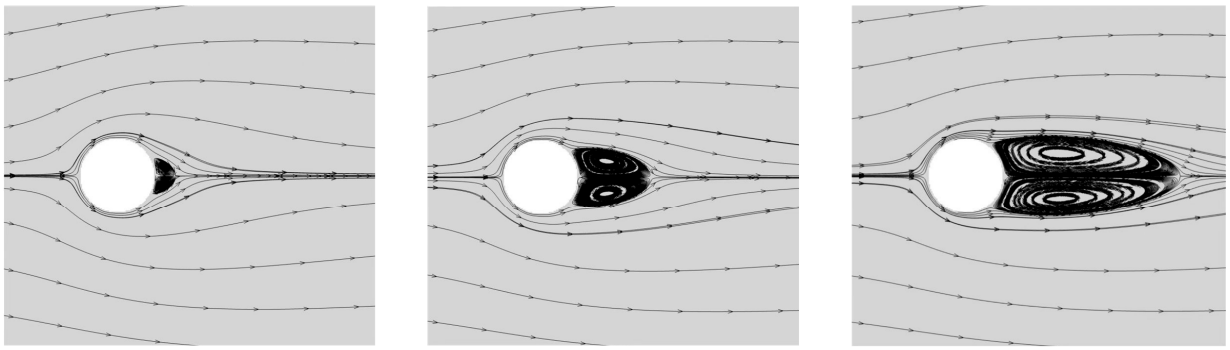

Figure 15. Streamline of the flow around circular cylinder at different Reynolds numbers.

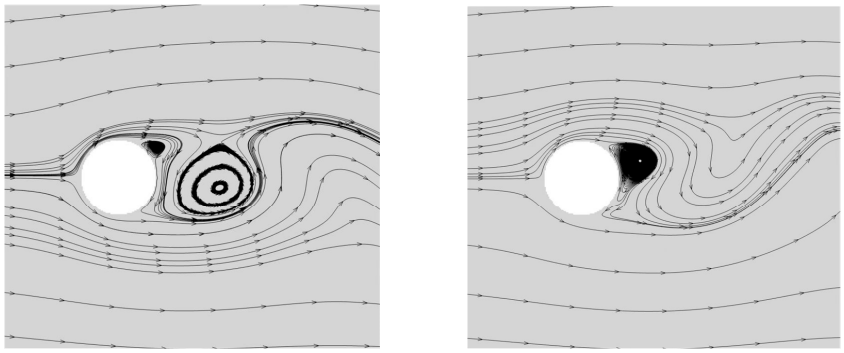

Figure 16. Streamline of the flow around circular cylinder at different Reynolds numbers.

The results of C_D at different Reynolds numbers are shown in Table 1. The C_D was compared with the corresponding results of Park et al. [40], Calhoun et al. [41], and Shi et al. [42]. It can be seen that the current experimental results using multi-level grid MRT-LBM are well matched.

Table 1. The drag coefficients C_D obtained on the cylinder and compared with the other literature [40–42] at different Reynolds numbers.

	Re = 10	**Re = 20**	**Re = 40**
	C_D	C_D	C_D
Park et al. [40]	2.78	2.01	1.51
Calhoun et al. [41]	—	2.19	1.62
Shi et al. [42]	2.79	2.05	1.55
Present	2.61	2.01	1.66

For Re < 47, the flow characteristics around the cylinder reach a steady state after a while and do not change anymore. This phenomenon is quantified with two parameters, separation angle θ_s, and wake length L/R_0 ($R_0 = D/2$). Table 2 lists the current θ_s and L/R_0 obtained at Re = 10, 20, and 40. It can be seen that the present results are similar to the results of He et al. [43] and Tuann et al. [44].

Table 2. Separation angle θ_s and wake length L/R_0 of the flow around cylinder at different Reynolds numbers.

	Re = 10		**Re = 20**		**Re = 40**	
	L/R_0	θ_s	L/R_0	θ_s	L/R_0	θ_s
He et al. [43]	0.474	26.89	1.80	44.1	4.49	52.84
Tuann et al. [44]	0.50	29.7	1.842	42.96	4.20	54.8
Present	0.423	23.9	1.84	39.5	4.611	49.8

For Re $\geq$ 47, the drag coefficient C_D and lift coefficient C_L are periodically changing. Figures 17 and 18 show the schematic diagrams of C_D and C_L periodically changing at Re = 100 and Re = 200, respectively. Table 3 shows the numerical results of the average drag coefficient $\overline{C_D}$, lift coefficient C_L, and Strouhal number St. of this method at Re = 100 and Re = 200. It can be seen that the numerical simulation results are accurate using the multi-level grid MRT-LBM.

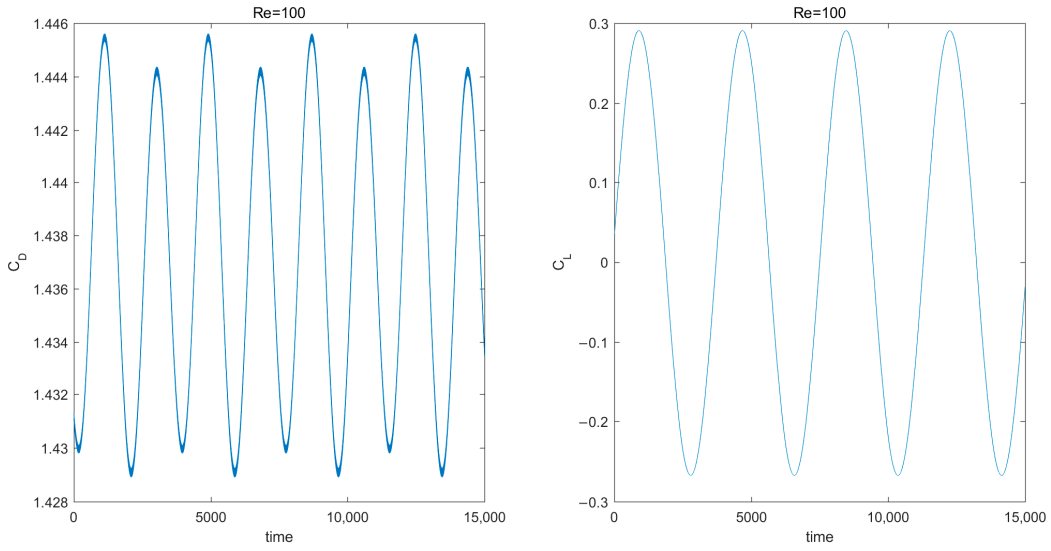

Figure 17. Schematic diagrams of C_D and C_L periodically changing at Re = 100.

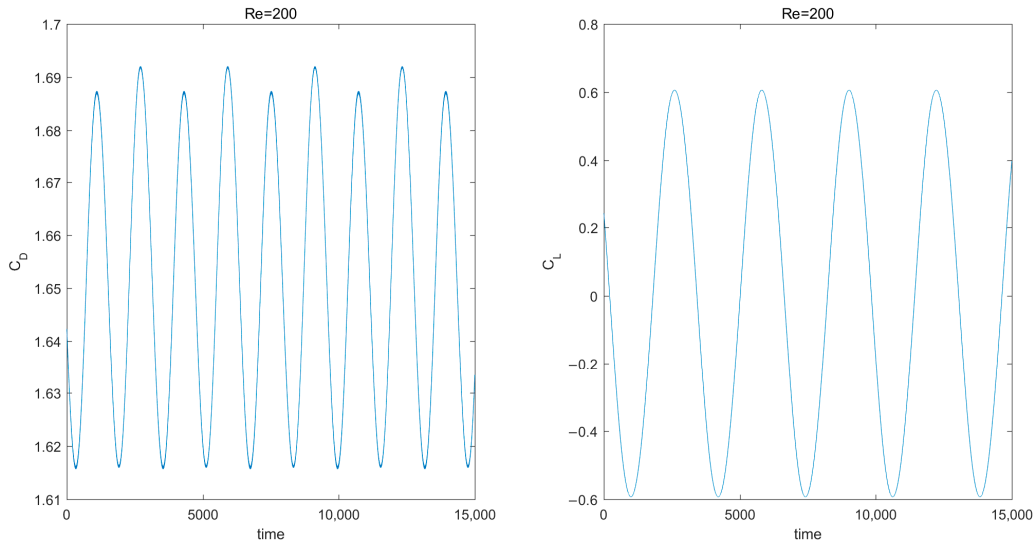

Figure 18. Schematic diagrams of C_D and C_L periodically changing at Re = 200.

Table 3. The parameter settings of averaged drag coefficient $\overline{C_D}$, lift coefficient C_L, and Strouhal numbers St at different Reynolds numbers, compared with those in the other literature.

	Re = 100			**Re = 200**		
	$\overline{C_D}$	C_L	St	$\overline{C_D}$	C_L	St
Shu et al. [45]	1.364	0.344	0.163	1.349	—	0.197
Le et al. [46]	1.39	0.346	0.160	1.38	0.676	0.192
Lecointe et al. [47]	—	—	—	1.58	0.5	0.194
Present	1.436	0.291	0.160	1.65	0.60	0.178

The single-level grid of flow around a circular cylinder is shown in Figure 19. The single-level grid resolution is equal to the resolution (D/32) of the fourth level of the four-level grid, and the total number of grids is 573,438. At Re = 100, we give the corresponding runtime overheads for the single-level grid MRT-LBM algorithm and the four-level grid MRT-LBM algorithm. The comparison of the number of grids and CPU time is shown in Figure 20. The total number of four-level grids is about 1/9 of the total number of single-level grids, and the computational efficiency can be improved by 6.76 times. The

numerical simulation results in Table 3, which match well with the other literature [45–47], show that the proposed multi-level grid MRT-LBM algorithm has good computational efficiency and accuracy.

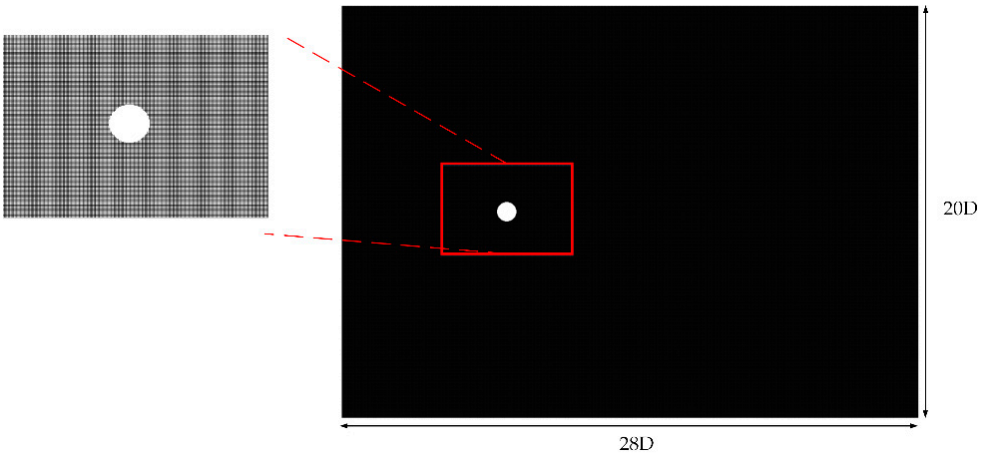

Figure 19. Schematic diagram of the single-level grid around the cylinder in 2D case.

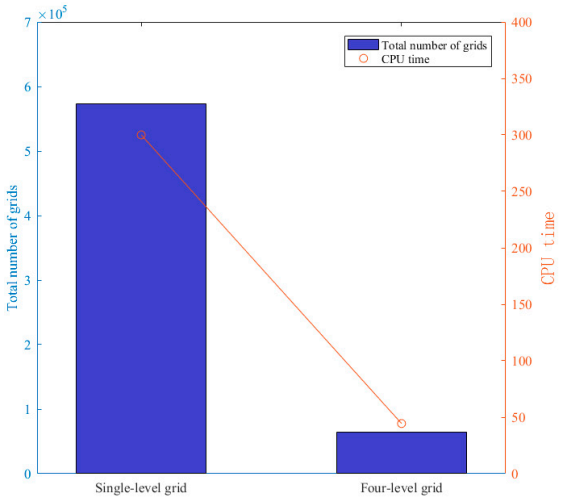

Figure 20. Comparing the total number of grids and CPU time for a single-level grid MRT-LBM and a four-level grid MRT-LBM at Re = 100.

4.2. Flow Past a Sphere in 3D Case

To verify the feasibility and accuracy of the multi-level grid MRT-LBM designed in this paper in a three-dimensional flow field, the benchmark experimental section simulates the flow state of flow past a sphere at a constant velocity for a customized iteration step time. The simulation results are appropriately compared with simulated phenomena and numerical results from the other literature. The radius of the sphere within the flow field of the enclosing box shape is R. The length, width, and height of the computational domain are set to 20R, 12R, and 12R, respectively. The flow velocity U = 0.1 is adopted experimentally. The position coordinates of the sphere are shown in Figure 21. A three-level multi-level grid is selected, with the size of the first-level grid being 1/4R and the size of the remaining levels being 1/2 of the previous level grid. A schematic diagram of the multilevel grid hierarchy is shown in Figure 22. The multi-level grid structure in the three-dimensional sphere flow experiment is set up with two levels of grids in the flow field area (the outer grids are partially refined to the second level of grid). The grids around the sphere are locally refined to obtain a third-level grid. Several grids are extrapolated from the sphere boundary grid to accurately simulate the rapidly changing region around the sphere.

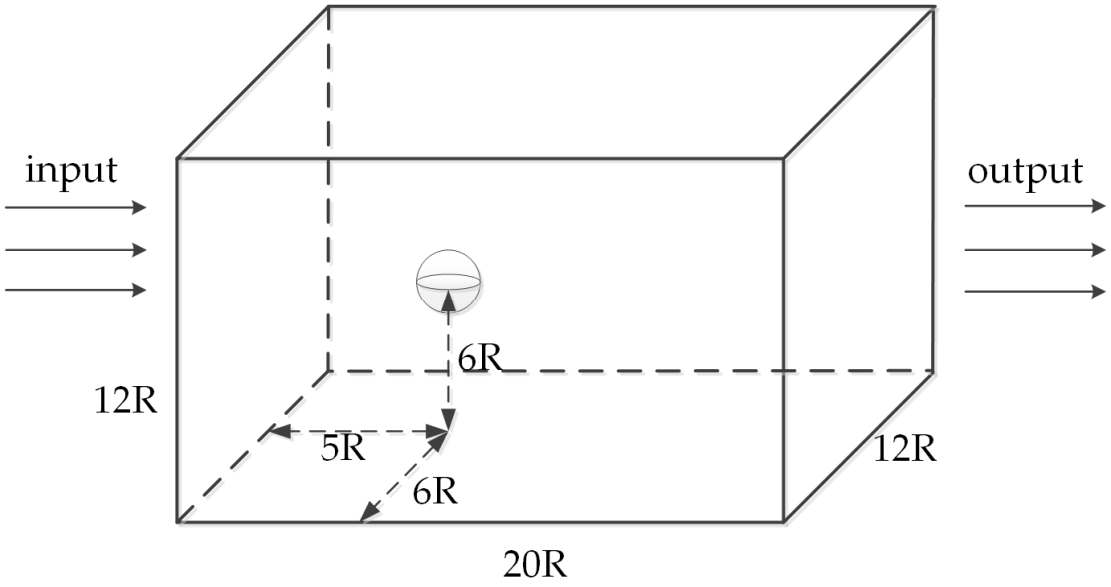

Figure 21. Schematic diagram of the computational domain of a flow past a sphere in 3D case.

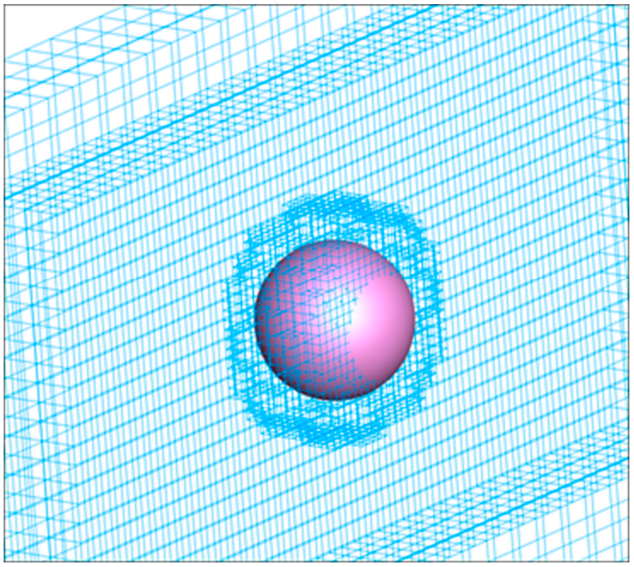

Figure 22. Schematic diagram of three-level grid of flow past a sphere in 3D case.

The flow past a sphere can be divided into three states: the flow is steady and axisymmetric, the flow is steady and non-axisymmetric, and the flow is unstable and non-axisymmetric. These states correspond to different Reynolds number intervals by Ref. [48], and it is known that the three states occur at Re < 200, 210, < Re < 270, and Re > 280, respectively. Therefore, Re = 100, 200, 250, and 300 are set to verify the proposed multi-level grid MRT-LBM in the flows past a sphere.

Streamlines and velocity contours on the XY plane at Re = 100 and 200 are shown in Figure 23. The area of the trailing vortices attached to the sphere becomes bigger with increasing Reynolds number. It can be seen that the flow in both cases is a stable symmetric flow. The drag coefficient C_D compared with the result of the other literature is shown in Table 4. The comparison of the data in the Table 4 shows that the multi-level grid MRT-LBM can correctly and effectively simulate the flow field changes.

The C_D drag coefficient can be formulated as:

$$C_D = \frac{8F_x}{\rho\pi U^2 D^2} \tag{34}$$

The C_L lift coefficient can be formulated as:

$$C_L = \frac{8\sqrt{F_y{}^2 + F_z{}^2}}{\rho\pi U^2 D^2}, \tag{35}$$

The streamlines and velocity contours for flow in XY plane and XZ plane at Re = 250 are shown in Figure 24. The XY plane flow field is still stable and symmetrical, while the XZ plane flow field is stable but no longer symmetrical.

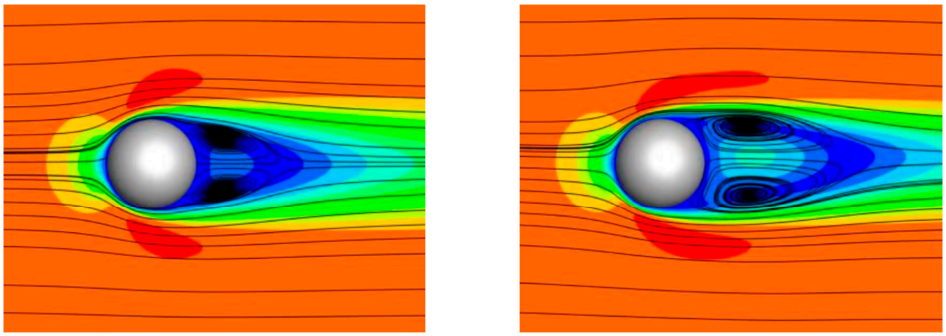

Figure 23. Streamlines and velocity contours for flow past a sphere on XY plane at different Reynolds numbers.

Table 4. Comparison of drag coefficient C_D at different Reynolds numbers.

	Re = 100	**Re = 200**
	C_D	C_D
Johnson et al. [48]	1.08	0.78
Hartmann et al. [49]	1.083	0.764
cheng et al. [50]	1.099	0.80
Present	1.067	0.774

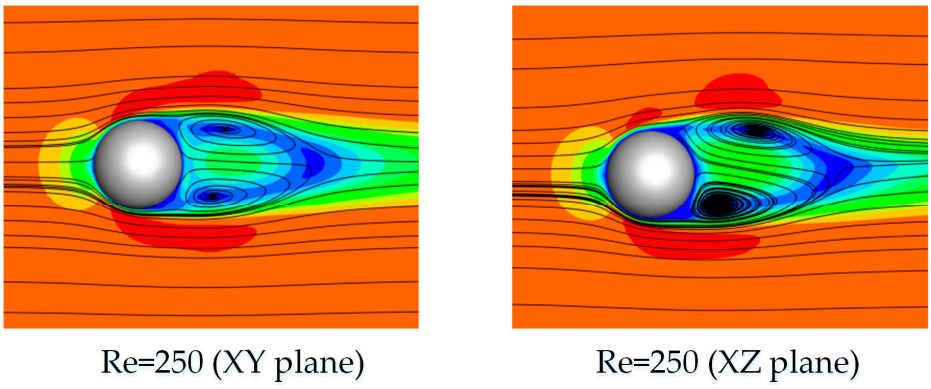

Figure 24. Streamlines and velocity contours for flow past a sphere at Re = 250.

The flow at Re = 300 is shown in Figure 25, and the streamline shows an unstable state of flow. In this case, periodic vortex shedding occurs. To quantify this unstable flow regime, the average drag coefficient C_D, average lift coefficient C_L, and Strouhal numbers were calculated in this experiment. Table 5 shows the comparison results with other physical quantities in the literature. The results indicate that the multi-level grid MRT-LBM proposed in this paper still has good stability and accuracy in the 3D case.

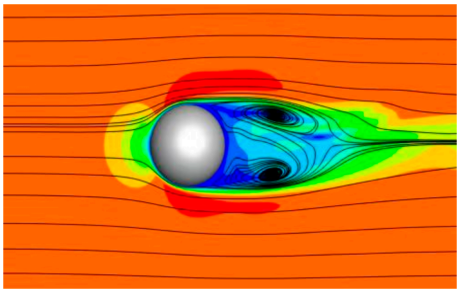

Figure 25. Streamlines and velocity contours for flow past a sphere at Re = 300.

Table 5. The drag coefficient C_D and the lift coefficient C_L are listed at Re = 250 and averaged drag coefficient $\overline{C_D}$, averaged lift coefficient $\overline{C_L}$, and Strouhal numbers are listed at Re = 300.

	Re = 250		**Re = 300**		
	C_D	C_L	$\overline{C_D}$	$\overline{C_L}$	**St**
Johnson et al. [48]	—	0.060	0.656	−0.069	0.137
Hartmann et al. [49]	0.689	0.065	0.657	−0.069	0.135
Kim et al. [51]	0.701	0.059	0.657	−0.067	0.134
Present	0.699	0.059	0.649	−0.066	0.133

In the following, we analyze the numerical simulation for the spherical disturbance flow with Re = 1000. At Re = 1000, a series of asymmetric hairpin vortexes appear in the wake of the sphere and the planar symmetry of the flow is lost. The complexity of the flow field in the wake region is seen in the vortex structure in Figure 26. Our simulation correctly captures this feature of the flow. Figure 27 shows the numerical results of the drag coefficient C_D obtained from simulations using the multi-level grid MRT-LBM algorithm proposed in this paper at Re = 1000, which can be seen to be very close to the theoretical value [52].

Figure 26. Vortex structure of the flow past a sphere at Re = 1000.

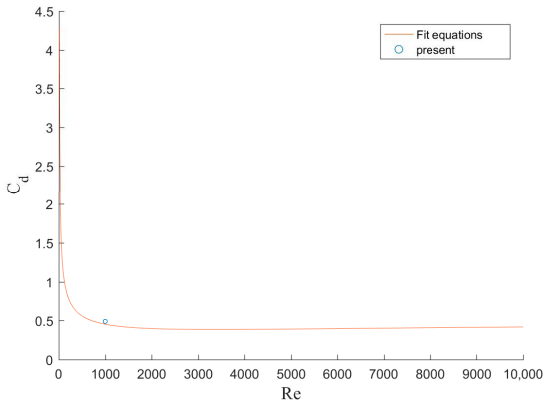

Figure 27. Comparing the drag coefficient C_D with the fitting equation [52] at Re = 1000.

5. Conclusions

The standard MRT-LBM usually uses a grid with a regular shape, such as a Cartesian grid. During numerical simulations, it is often necessary to obtain fine flow conditions in localized areas. If all grids are set to a small enough size, this will increase the total number of computational grids and the required storage space. Therefore, this paper proposes a multi-level grid generation algorithm, which is able to set different grid sizes according to the needs of the computational problem and is able to increase the fineness of the surface of complex geometry in order to better simulate the flow near the geometry. A detailed multi-level grid MRT-LBM evolution model and algorithm are given based on the collision migration process of MRT-LBM and the characteristics of multi-level grids.

Moreover, the proposed multi-level grid MRT-LBM algorithm can effectively reduce the computational effort and complexity through setting buffers at different levels of the grid interfaces and eliminating temporal interpolation and reducing spatial interpolation when transferring information at different level of grid intersections. The accuracy of our proposed multi-level grid MRT-LBM algorithm has been verified by both 2D and 3D simulation experiments, i.e., flow around a circular cylinder using a four-level grid and flow past a sphere using a three-level grid, respectively. In the 2D simulation experiment, our algorithm achieved higher accuracy in the quantitative data such as separation angle θ_s, wake length L/R_0, and Strouhal numbers, compared to the results in the published literature. Meanwhile, the total number of grids of the four-level grid MRT-LBM is only 1/9 of the single-level grid MRT-LBM, which can improve the computational efficiency by 6.76 times while reducing the computational storage. The three-level grid MRT-LBM is used to simulate the flow past a sphere at Re = 100, 200, 250, 300, and 1000 in 3D case. The streamlines, velocity contours, and vortex structure diagrams are plotted at different Reynolds numbers. The hairpin vortex in the wave is successfully obtained at Re = 1000. The experimental quantitative data C_D match well with the results in the published literature, which shows that the proposed multi-level grid MRT-LBM in this paper can maintain numerical stability at higher Reynolds numbers. The multi-level grid MRT-LBM algorithm provides favorable conditions for subsequent parallel algorithm design and offers a solution to increasingly complex computational fluid dynamics problems.

Author Contributions: Conceptualization, Z.L., S.L. and W.Z.; methodology, Z.L. and S.L.; software, S.L. and J.R.; validation, Z.L., S.L. and L.Z.; investigation, L.Z. and W.Z.; resources, L.Z., J.X. and D.H.; data curation, Z.L. and S.L.; writing—original draft preparation, Z.L., S.L., J.R. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (62102243), Shanghai Sailing Program (21YF1417000), The Open Project of Shanghai Key Laboratory of Trustworthy Computing (OP202102), and Startup Foundation for Young Teachers of Shanghai Ocean University.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their gratitude for the support of the Fishery Engineering and Equipment Innovation Team of Shanghai High-level Local University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ma, L.; Barakos, G.N.; Zhao, Q. A 3D implicit structured multi-block grid finite volume method for computational structural dynamics. *Aerosp. Sci. Technol.* **2021**, *117*, 106980. [CrossRef]
2. Alhazmi, N.; Almutairi, G.; Alenazey, F.; AlOtaibi, B. Three-dimensional computational fluid dynamics modeling of button solid oxide fuel cell. *Electrochim. Acta* **2021**, *390*, 138838. [CrossRef]
3. Chowdhury, A.A.; Rasul, M.; Khan, M. Thermal performance assessment of a retrofitted building using an integrated energy and computational fluid dynamics (IE-CFD) approach. *Energy Rep.* **2022**, *8*, 709–717. [CrossRef]
4. Mousazadeh, S.; Shahmardan, M.; Tavangar, T.; Hosseinzadeh, K.; Ganji, D. Numerical investigation on convective heat transfer over two heated wall-mounted cubes in tandem and staggered arrangement. *Theor. Appl. Mech. Lett.* **2018**, *8*, 171–183. [CrossRef]
5. Succi, S. *The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond*; Oxford University Press: Oxford, UK, 2001.

6. Qian, Y.H.; Humières, D.D.; Lallemand, P. Lattice BGK Models for Navier-Stokes Equation. *Europhys. Lett.* **1992**, *17*, 479. [CrossRef]
7. Fei, F.; Liu, H.; Liu, Z.; Zhang, J. A Benchmark Study of Kinetic Models for Shock Waves. *AIAA J.* **2020**, *58*, 2596–2608. [CrossRef]
8. Zhou, D.; Lu, Z.; Guo, T. A Gas-Kinetic BGK Scheme for Natural Convection in a Rotating Annulus. *Appl. Sci.* **2018**, *8*, 733. [CrossRef]
9. Zhao, J. Discrete-Velocity Vector-BGK Models Based Numerical Methods for the Incompressible Navier-Stokes Equations. *Commun. Comput. Phys.* **2021**, *29*, 420–444. [CrossRef]
10. Lou, Q.; Meng, X.; Karniadakis, G.E. Physics-informed neural networks for solving forward and inverse flow problems via the Boltzmann-BGK formulation. *J. Comput. Phys.* **2021**, *447*, 110676. [CrossRef]
11. Baranger, C.; Dauvois, Y.; Marois, G.; Mathé, J.; Mathiaud, J.; Mieussens, L. A BGK model for high temperature rarefied gas flows. *Eur. J. Mech.—B Fluids* **2020**, *80*, 1–12. [CrossRef]
12. Zhang, J.; Yao, S.Q.; Fei, F.; Ghalambaz, M.; Wen, D.S. Competition of natural convection and thermal creep in a square enclosure. *Phys. Fluids* **2020**, *32*, 12. [CrossRef]
13. D'Humières, D. Generalized lattice-Boltzmann equations. *Rarefied Gas Dyn.* **1992**, *159*, 450–458.
14. Lallemand, P.; Luo, L.-S. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E* **2000**, *61*, 6546. [CrossRef] [PubMed]
15. Ezzatneshan, E.; Vaseghnia, H. Dynamics of an acoustically driven cavitation bubble cluster in the vicinity of a solid surface. *Phys. Fluids* **2021**, *33*, 123311. [CrossRef]
16. Cheng, X.; Su, R.; Shen, X.; Deng, T.; Zhang, D.; Chang, D.; Zhang, B.; Qiu, S. Modeling of indoor airflow around thermal manikins by multiple-relaxation-time lattice Boltzmann method with LES approaches. *Numer. Heat Transf. Part A Appl.* **2019**, *77*, 215–231. [CrossRef]
17. Yang, Y.; Shan, M.; Kan, X.; Shangguan, Y.; Han, Q. Thermodynamic of collapsing cavitation bubble investigated by pseudopotential and thermal MRT-LBM. *Ultrason. Sonochem.* **2020**, *62*, 104873. [CrossRef]
18. Jiang, C.; Zhou, H.; Xia, M.; Tang, J.; Jiang, S.; Zhang, M. Stability conditions of multiple-relaxation-time lattice Boltzmann model for seismic wavefield modeling. *J. Appl. Geophys.* **2022**, *204*, 104742. [CrossRef]
19. Huang, T.; Gu, H.; Zhang, J.; Li, B.; Sun, J.; Wu, W. An Improved Multi-Relaxation Time Lattice Boltzmann Method for the Non-Newtonian Influence of the Yielding Fluid Flow in Cement-3D Printing. *Materials* **2018**, *11*, 2342. [CrossRef]
20. Filippova, O.; Hänel, D. Grid refinement for lattice-BGK models. *J. Comput. Phys.* **1998**, *147*, 219–228. [CrossRef]
21. Dupuis, A.; Chopard, B. Theory and applications of an alternative lattice Boltzmann grid refinement algorithm. *Phys. Rev. E* **2003**, *67*, 066707. [CrossRef] [PubMed]
22. Stiebler, M.; Krafczyk, M.; Freudiger, S.; Geier, M. Lattice Boltzmann large eddy simulation of subcritical flows around a sphere on non-uniform grids. *Comput. Math. Appl.* **2011**, *61*, 3475–3484. [CrossRef]
23. Astoul, T.; Wissocq, G.; Boussuge, J.F.; Sengissen, A.; Sagaut, P. Analysis and reduction of spurious noise generated at grid refinement interfaces with the lattice Boltzmann method. *J. Comput. Phys.* **2020**, *418*, 31. [CrossRef]
24. Rohde, M.; Kandhai, D.; Derksen, J.; Van den Akker, H.E. A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes. *Int. J. Numer. Methods Fluids* **2006**, *51*, 439–468. [CrossRef]
25. Schornbaum, F.; Rude, U. Massively parallel algorithms for the lattice Boltzmann method on nonuniform grids. *SIAM J. Sci. Comput.* **2016**, *38*, C96–C126. [CrossRef]
26. Yu, Z.; Fan, L.-S. An interaction potential based lattice Boltzmann method with adaptive mesh refinement (AMR) for two-phase flow simulation. *J. Comput. Phys.* **2009**, *228*, 6456–6478. [CrossRef]
27. Eitel-Amor, G.; Meinke, M.; Schröder, W. A lattice-Boltzmann method with hierarchically refined meshes. *Comput. Fluids* **2013**, *75*, 127–139. [CrossRef]
28. Guo, Z.-L.; Shi, B.-C.; Wang, N.-C. A nonuniform lattice boltzmann method based on domain decomposition. *Chin. J. Comput. Phys.* **2001**, *18*, 181.
29. Ezzatneshan, E. Study of unsteady separated fluid flows using a multi-block lattice Boltzmann method. *Aircr. Eng. Aerosp. Technol.* **2021**, *93*, 139–149. [CrossRef]
30. Yu, D.; Mei, R.; Shyy, W. A multi-block lattice Boltzmann method for viscous fluid flows. *Int. J. Numer. Methods Fluids* **2002**, *39*, 99–120. [CrossRef]
31. Yu, D.; Girimaji, S.S. Multi-block lattice Boltzmann method: Extension to 3D and validation in turbulence. *Phys. A Stat. Mech. Its Appl.* **2006**, *362*, 118–124. [CrossRef]
32. Guzik, S.M.; Weisgraber, T.H.; Colella, P.; Alder, B.J. Interpolation methods and the accuracy of lattice-Boltzmann mesh refinement. *J. Comput. Phys.* **2014**, *259*, 461–487. [CrossRef]
33. Liu, Z.; Tian, F.-B.; Feng, X. An efficient geometry-adaptive mesh refinement framework and its application in the immersed boundary lattice Boltzmann method. *Comput. Methods Appl. Mech. Eng.* **2022**, *392*, 114662. [CrossRef]
34. Peng, Y.; Shu, C.; Chew, Y.-T.; Niu, X.; Lu, X.-Y. Application of multi-block approach in the immersed boundary–lattice Boltzmann method for viscous fluid flows. *J. Comput. Phys.* **2006**, *218*, 460–478. [CrossRef]
35. Arora, N.; Gupta, A.; Shyy, W. A shifting discontinuous-grid-block lattice Boltzmann method for moving boundary simulations. *Comput. Fluids* **2016**, *125*, 59–70. [CrossRef]

36. Chen, S.; Peng, C.; Teng, Y.; Wang, L.-P.; Zhang, K. Improving lattice Boltzmann simulation of moving particles in a viscous flow using local grid refinement. *Comput. Fluids* **2016**, *136*, 228–246. [CrossRef]
37. Wu, C.-M.; Zhou, Y.-S.; Lin, C.-A. Direct numerical simulations of turbulent channel flows with mesh-refinement lattice Boltzmann methods on GPU cluster. *Comput. Fluids* **2020**, *210*, 104647. [CrossRef]
38. Akenine-Möllser, T. Fast 3D Triangle-Box Overlap Testing. *J. Graph. Tools* **2001**, *6*, 29–33. [CrossRef]
39. Yu, D.; Mei, R.; Shyy, W. A unified boundary treatment in lattice boltzmann method. In Proceedings of the 41st Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 6–9 January 2003.
40. Park, J.; Kwon, K.; Choi, H. Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160. *KSME Int. J.* **1998**, *12*, 1200–1205. [CrossRef]
41. Calhoun, D. A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. *J. Comput. Phys.* **2002**, *176*, 231–275. [CrossRef]
42. Shi, X.; Huang, X.; Zheng, Y.; Ji, T. A hybrid algorithm of lattice Boltzmann method and finite difference–based lattice Boltzmann method for viscous flows. *Int. J. Numer. Methods Fluids* **2017**, *85*, 641–661. [CrossRef]
43. He, X.; Doolen, G. Lattice Boltzmann method on curvilinear coordinates system: Flow around a circular cylinder. *J. Comput. Phys.* **1997**, *134*, 306–315. [CrossRef]
44. Tuann, S.-Y.; Olson, M.D. Numerical studies of the flow around a circular cylinder by a finite element method. *Comput. Fluids* **1978**, *6*, 219–240. [CrossRef]
45. Wu, J.; Shu, C. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *J. Comput. Phys.* **2009**, *228*, 1963–1979. [CrossRef]
46. Le, D.; Khoo, B.; Lim, K. An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains. *Comput. Methods Appl. Mech. Eng.* **2008**, *197*, 2119–2130. [CrossRef]
47. Lecointe, Y.; Piquet, J. On the use of several compact methods for the study of unsteady incompressible viscous flow round a circular cylinder. *Comput. Fluids* **1984**, *12*, 255–280. [CrossRef]
48. Johnson, T.A.; Patel, V.C. Flow past a sphere up to a Reynolds number of 300. *J. Fluid Mech.* **1999**, *378*, 19–70. [CrossRef]
49. Hartmann, D.; Meinke, M.; Schröder, W. A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 1038–1052. [CrossRef]
50. Cheng, C.; Galindo-Torres, S.; Zhang, X.; Zhang, P.; Scheuermann, A.; Li, L. An improved immersed moving boundary for the coupled discrete element lattice Boltzmann method. *Comput. Fluids* **2018**, *177*, 12–19. [CrossRef]
51. Kim, J.; Kim, D.; Choi, H. An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. *J. Comput. Phys.* **2001**, *171*, 132–150. [CrossRef]
52. Turton, R.; Levenspiel, O. A short note on the drag correlation for spheres. *Powder Technol.* **1986**, *47*, 83–86. [CrossRef]