

Volumetric method for calculating the flow around moving objects in lattice-Boltzmann schemes

M. Rohde, J. J. Derksen, and H. E. A. Van den Akker

Kramers Laboratorium voor Fysische Technologie, Delft University of Technology, Prins Bernhardlaan 6, 2628 BW Delft, The Netherlands

(Received 24 August 2001; published 23 April 2002)

A method for calculating the fluid flow around moving objects is presented, based on a volumetric representation of the lattice-Boltzmann scheme and surfaces defined by facets. It enables us to move objects of arbitrary shape and orientation independent of the position of the grid nodes. To represent the motion of the object, additional momentum is added to the reflected particles from each facet in the propagation step. These particles are redistributed on nodes in the vicinity of the surface, depending on the position and orientation of the facet. Because the surface is considered to be closed, additional techniques need to be used to guarantee the conservation of mass. The flow field of a moving periodic cubic array of cubes at two Re numbers ($Re = 0.5$ and $Re = 50$) is compared with that of a fixed array. For $Re = 0.5$, no significant deviations are found for the velocity field, pressure field, and the drag force. For $Re = 50$, the drag and pressure field exhibit small fluctuations that relate to the position of the surface relative to the position of the grid. However, the influence of the pressure fluctuations on the velocity field is very small. Results on the velocity for a moving array of cubes show second-order accuracy in the lattice spacing. For physical consistency, the drag force on a periodic cubic array of moving spheres at $Re = 0.5$ is compared with Hasimoto's analytical solution. The dependence on the grid spacing, the resolution of the surface of the object, and the viscosity have been studied. The discrepancies between simulations and the analytical results are smaller than 1.5%. For $Re = 50$, the drag force, the streamline pattern, and the pressure field around a moving sphere in a large periodic domain showed good agreement with data from literature on a single sphere in an infinitely large flow field.

DOI: 10.1103/PhysRevE.65.056701

PACS number(s): 02.70.-c, 05.20.Dd, 45.50.-j, 47.11.+j

I. INTRODUCTION

Objects moving in a fluid can be found in many situations, such as rotating impellers in stirred vessels or particles moving in suspensions. Although the physics of the flow around a moving surface is well understood, setting up a numerical scheme for simulating it requires careful modeling. In conventional computational fluid dynamics (CFD), moving or deforming grids and, in some special cases, sliding meshes are often employed. In the case of moving or deforming grids, the grid geometry is constantly adjusted to the changing position of the surface (see, e.g., Refs. [1,2]).

Just as in conventional CFD, lattice-Boltzmann simulations require specific techniques if the moving surfaces are a part of the flow problem. Although the lattice-Boltzmann methods are significantly different from the conventional CFD techniques, similar issues need to be studied in order to impose the appropriate boundary condition at the surface. They relate to the adaptation of the grid geometry to the position of the object and the conservation of mass and momentum.

As far as imposing the no-slip boundary condition for moving surfaces in lattice-Boltzmann schemes is concerned, a lot of pioneering work has been carried out by Ladd [3,4]. In his technique, the no-slip boundary condition is recovered by transferring an extra amount of mass across the surface, which is proportional to the velocity of the particle. This addition or subtraction of mass must be balanced in some way in order to conserve mass. For staircase shaped objects, grid nodes inside the moving object (i.e., internal fluid) were used to establish mass conservation [3]. A drawback is that this may cause stability problems in the case of freely mov-

ing low-density objects [4]. Several extensions to the method were proposed, such that the arbitrary density ratios could be simulated [5,6]. Aidun *et al.* [5] excluded the internal fluid by considering the surface to be closed. Although the addition of mass on the fluid node cannot be balanced by the subtraction of mass on a node inside the object, simulations showed no significant global mass change in the computational domain. Aidun *et al.* also proposed a technique to determine the mass distributions on new nodes that were formerly covered by the object.

As in these methods the surface is located exactly between the grid nodes, inclined and curved surfaces are discretized into staircase shapes. This implies that the surface of the object cannot be positioned accurately at every time step, because it is directly related to the position of the grid nodes. Especially for curved and inclined surfaces, this causes its shape to change during its motion, which might cause significant fluctuations in the hydrodynamics. In addition, it may inherit the accuracy problems of the bounce-back rule for fixed objects [7–9].

Using a more accurate definition of the moving surface and its position would prevent the problems mentioned above. For fixed surfaces, methods were developed where the surface can be defined independently of the grid geometry [10–13]. Chen *et al.* [12] proposed a *volumetric* approach, where the surface is defined by a set of facets and the grid by cubic grid *cells* (rather than nodes). The surface “cuts” some of the grid cells that become noncubic. The overall grid geometry remains cubic, which is very convenient (but not a prerequisite) for lattice-Boltzmann simulations. By setting up a geometry-dependent bounce-back rule for these cells, accurate calculations could be carried out for

fixed surfaces of arbitrary position and shape. Another approach that already has been applied for moving surfaces defines the surface by means of a set of points in space, independent of the grid [14]. With this method, simulations of fluid flow at high Re numbers ($Re=10^5$) have been performed, exhibiting a maximum difference of 5% between the actual and the prescribed velocities at the surface. The position of the surface, however, could not be determined exactly due to interpolation of the hydrodynamical quantities to grid nodes adjacent to the moving surface.

In this paper, we present a method for dealing with moving surfaces of arbitrary shape and position with respect to the grid. It combines the ideas of Chen regarding the surface definition by a set of facets with the method of Ladd to impose a no-slip boundary condition for a moving surface. As the surface is considered to be closed, no internal fluid is used. Additional techniques are proposed, which deal with mass conservation.

The main advantages of our approach are the following. (i) The shape of the surface does not change during its motion; this may avoid unwanted fluctuations. (ii) The method is found to be second-order accurate in the velocity for the simulations performed in this paper, irrespective of the position of the surface. (iii) The resolution of the surface is an additional parameter for increasing the accuracy of the simulations. (iv) The flux of momentum (i.e., the force) can be calculated separately for each facet; this makes it possible to determine the tangential and normal forces acting on the surface locally and accurately.

The paper is organized as follows. In Sec. II, we briefly describe the technique as proposed by Chen *et al.* for fixed surfaces. Then, the method for moving surfaces is presented, preceded by a brief description of the method of Ladd. In Sec. III, the method is validated for numerical consistency. In addition, simulations are compared with experimental and simulation data from literature.

II. THEORY

To employ the lattice-Boltzmann techniques for simulating the flow around moving objects without the use of a staircase shaped object, several issues need to be addressed. First, the surface of the object must be defined in such a way that it is independent of the position of the grid nodes. Compared to the staircase shaped surfaces, this results in a more detailed description of the surface and it makes it possible to position such a surface more accurately in the computational domain. Second, a method for establishing the no-slip boundary condition at a fixed surface must be found, which takes the grid-independent surface definition into account. This method has to be extended to moving surfaces. Finally, the varying position of the surface with respect to the grid requires a special treatment of the mass distribution on the grid nodes adjacent to the surface.

In Sec. II A and Sec. II B, we describe the accurate definition of the surface geometry and a method for establishing the no-slip boundary condition for such surfaces at a fixed position. It essentially represents a modified bounce-back rule for grid nodes in the vicinity of the surface. In Sec. II C,

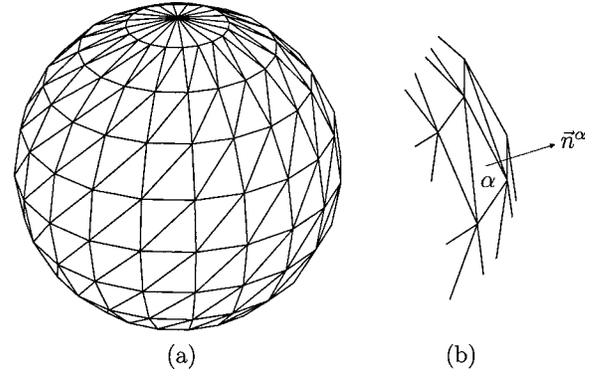


FIG. 1. A surface consisting of triangular facets. (a) A sphere with 360 facets. (b) A detailed view of a facet α and its surface normal.

these ideas are extended to moving surfaces.

A. Surface geometry

The surface of an object in the computational domain can be defined by facets S^α with area A^α and a surface normal \vec{n}^α (see Figs. 1 and 2). These facets can be placed in the domain, independently of the grid geometry. Each grid node \vec{x} is situated in the center of a cubic cell (i.e., *grid cell*) with size $\Delta x = \Delta y = \Delta z$. The mass, usually located on the grid node in the lattice-Boltzmann simulations, is now considered to be uniformly distributed throughout the cell. All grid cells are cubic, except for the cells that are cut by the surface of the object. These cells have a volume $\Delta V(\vec{x})$ and contain mass that is equal to $\sum_{i=1, \dots, b} N_i(\vec{x}, t) = \Delta V(\vec{x}) \sum_{i=1, \dots, b} n_i(\vec{x}, t)$, where $n_i(\vec{x}, t)$ represents the density distribution of the

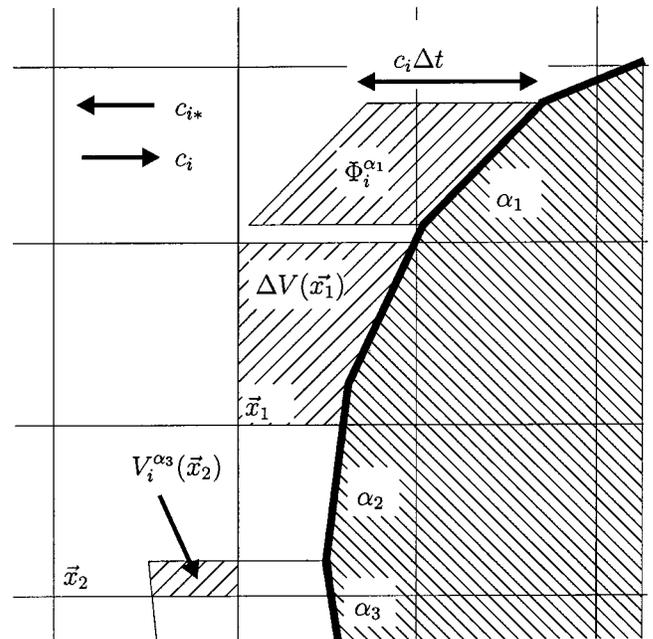


FIG. 2. Two-dimensional representation of the volumes Φ_i^α , $V_i^\alpha(\vec{x})$, and $\Delta V(\vec{x})$. α_1 , α_2 , and α_3 are the facets of the surface. \vec{x}_1 and \vec{x}_2 are boundary cells.

lattice-Boltzmann scheme in velocity direction i at position \vec{x} . The number of velocity directions is equal to b . Grid cells that are situated completely inside the object do not take part in the calculations.

B. No-slip boundary condition for fixed surfaces

An accurate method for calculating the flow around a fixed object according to the geometrical definition in the preceding section has been described by Chen *et al.* [12]. The main idea of this method is that the no-slip boundary condition is enforced by applying a modified bounce-back rule reflecting the definition of the surface. The main points of Chen's method are explained in this section.

During the propagation step, a fraction of the mass in the cells adjacent to the surface (i.e., *boundary cells*) hits the surface of the object. When this mass moves with velocity \vec{c}_i and hits facet S^α , it must move within a parallelepiped, extruded from the facet S^α in direction i^* ($\vec{c}_{i^*} \equiv -\vec{c}_i$) (see Fig. 2). The volume of this parallelepiped is equal to $\Phi_i^\alpha = |\vec{c}_{i^*} \cdot \vec{n}^\alpha| A^\alpha \Delta t = |\vec{c}_i \cdot \vec{n}^\alpha| A^\alpha \Delta t$. Because the parallelepiped may intersect several grid cells, the mass that moves within the parallelepiped may originate from different cells. Therefore, we also need to calculate the volume of the intersections of the parallelepiped and the cubic cells \vec{x} separately. These volumes are defined as $V_i^\alpha(\vec{x}) [= V_{i^*}^\alpha(\vec{x})]$.

With the definitions stated above, an adjusted lattice-Boltzmann equation for boundary cells can be derived, which accounts for the geometrical definition of the surface described in the preceding section. This adjusted equation reads

$$N_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) = P_i^{undist}(\vec{x}) N'_i(\vec{x}, t) + Q_i(\vec{x} + \vec{c}_i \Delta t, t), \quad (2.1)$$

where $N'_i(\vec{x}, t) \equiv N_i(\vec{x}, t) + \Omega_i(\vec{x}, t)$ is the right-hand side of the "normal" lattice-Boltzmann equation for a cubic grid. $P_i^{undist}(\vec{x})$ represents the fraction of mass that does not hit the surface during the propagation step. This fraction moves undisturbed from cell \vec{x} to cell $\vec{x} + \vec{c}_i \Delta t$. The term $Q_i(\vec{x} + \vec{c}_i \Delta t, t)$ is the mass that is reflected from the surface and arrives in cell $\vec{x} + \vec{c}_i \Delta t$. When no mass hits the surface [$P_i^{undist}(\vec{x}) = 1$] and, consequently, no mass is reflected from the surface [$Q_i(\vec{x} + \vec{c}_i \Delta t, t) = 0$], the "normal" lattice-Boltzmann equation is recovered.

The undisturbed fraction of mass $P_i^{undist}(\vec{x})$ can be determined by calculating the *disturbed* fraction. The fraction of mass located in cell \vec{x} that moves in the direction i and hits any facet S^α of the object is

$$P_i^{dist}(\vec{x}) = \sum_{\alpha} \frac{V_i^\alpha(\vec{x})}{\Delta V(\vec{x})} \equiv \sum_{\alpha} P_i^\alpha(\vec{x}) \quad (\vec{c}_i \cdot \vec{n}^\alpha < 0). \quad (2.2)$$

From Eq. (2.2) we obtain

$$P_i^{undist}(\vec{x}) = 1 - P_i^{dist}(\vec{x}) \quad (\vec{c}_i \cdot \vec{n}^\alpha < 0). \quad (2.3)$$

To determine the amount of mass that reflects from the surface $Q_i(\vec{x} + \vec{c}_i \Delta t, t)$, the total incoming mass per facet S^α must be calculated first. This mass, moving in the direction i , can be calculated from $P_i^\alpha(\vec{x})$ and reads

$$\Gamma_i^{in,\alpha}(t) = \sum_{\vec{x}} P_i^\alpha(\vec{x}) N_i(\vec{x}, t) \quad (\vec{c}_i \cdot \vec{n}^\alpha < 0). \quad (2.4)$$

Additionally, we must define how this mass is reflected into the fluid. In the case of the no-slip boundary condition at the surface, the bounce-back rule can be applied. In this case, the mass $\Gamma_i^{in,\alpha}(t)$ is reflected back in the opposite direction i^* . Consequently,

$$\Gamma_{i^*}^{out,\alpha}(t) = \Gamma_i^{in,\alpha}(t), \quad (2.5)$$

where $\Gamma_{i^*}^{out,\alpha}(t)$ represents the mass that is reflected from the facet S^α in the direction i^* . If we further assume that the reflected mass is *uniformly* distributed in the parallelepiped with volume Φ_i^α , the term $Q_i(\vec{x} + \vec{c}_i \Delta t, t)$ can be calculated according to

$$Q_i(\vec{x} + \vec{c}_i \Delta t, t) = \sum_{\alpha} \frac{V_i^\alpha(\vec{x} + \vec{c}_i \Delta t)}{\Phi_i^\alpha} \Gamma_{i^*}^{out,\alpha}(t) \quad (\vec{c}_i \cdot \vec{n}^\alpha > 0). \quad (2.6)$$

Now that $P_i^{undist}(\vec{x})$ and $Q_i(\vec{x} + \vec{c}_i \Delta t, t)$ are known, Eq. (2.1) can be used as the lattice-Boltzmann equation for boundary cells.

C. No-slip boundary condition for moving surfaces

We now consider the implementation of the method described in the previous sections for an object moving through a fluid with a velocity \vec{u}_b . A modified set of equations for boundary cells can be set up, which is dependent on the geometrical quantities $\Delta V(\vec{x}, t)$, $\Phi_i^\alpha(t)$, $V_i^\alpha(\vec{x}, t)$, and $\vec{n}^\alpha(t)$ (the quantities are time dependent in the case of a moving surface). One of the equations represents the collision and propagation steps of the lattice-Boltzmann scheme and is an extended version of Eq. (2.1). The other equations relate to the changing position of the surface relative to the grid cells. Due to the motion of the surface, volumes of boundary cells change, new cells may appear and some cells may disappear into the object. Mass in these cells must be properly adjusted in order to avoid unphysical density fluctuations around the object.

1. Modification of the propagation rule

In order to establish the no-slip boundary condition for a moving surface, Eq. (2.1) is adjusted according to the method of Ladd for moving staircase shaped surfaces. We briefly describe this method first.

In the method of Ladd [3], the surface is situated exactly in between two grid nodes. During the propagation step, mass on a grid node adjacent to the surface is not only reflected into the fluid, but extra mass is transferred across the surface of the object to (or from) a grid node *inside* the object. When this amount of mass transferred is proportional to the surface velocity \vec{u}_b , the no-slip boundary condition is established at the surface on the macroscopic level. By applying this procedure also for the grid node inside the object and by introducing fluid grid nodes throughout the *entire* object, mass is conserved. The modified propagation rule for the grid nodes adjacent to the surface inside and outside the object (in terms of densities) reads

$$\begin{aligned} n_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) &= n_i^*(\vec{x} + \vec{c}_i \Delta t, t_+) + 2t_{p,i} \rho(\vec{x}, t) \\ &\quad \times (\vec{u}_b \cdot \vec{c}_i) / c_s^2, \\ n_i^*(\vec{x}, t + \Delta t) &= n_i(\vec{x}, t_+) - 2t_{p,i} \rho(\vec{x}, t) (\vec{u}_b \cdot \vec{c}_i^*) / c_s^2, \end{aligned} \quad (2.7)$$

where the t_+ indicates the moment after the collision but before the propagation step. The speed of sound is denoted by c_s and $t_{p,i}$ represents a direction dependent weight factor originating from the lattice-Boltzmann scheme (see also Sec. III A).

Analogous to the method of Ladd, Eq. (2.1) can be modified such that the velocity \vec{u}_b at the surface is recovered. The modified equation reads

$$\begin{aligned} N_i(\vec{x} + \vec{c}_i \Delta t, t^*) &= P_i^{undist}(\vec{x}) N_i'(\vec{x}, t) + Q_i(\vec{x} + \vec{c}_i \Delta t, t) \\ &\quad + B_i(\vec{x} + \vec{c}_i \Delta t, t). \end{aligned} \quad (2.8)$$

Equation (2.8) represents the collision and propagation steps during a time shift $t \rightarrow t^*$, where t^* denotes the moment before the surface is moved to a new position. The new term $B_i(\vec{x} + \vec{c}_i \Delta t, t)$ is comparable with $2t_{p,i} \rho(\vec{x}, t) (\vec{u}_b \cdot \vec{c}_i) / c_s^2$ in Eq. (2.7), but is also dependent on geometrical quantities.

To enforce a velocity $\vec{u}_b(t)$ at a facet S^α , an extra amount of mass must be added to (or removed from) boundary cells. This extra amount of mass $\Delta \Gamma_i^{out,\alpha}(\vec{x}, t)$ is proportional to $\vec{u}_b(t)$ and the volume fraction $V_i^\alpha(\vec{x}, t) / \Delta V(\vec{x}, t)$ and is equal to

$$\begin{aligned} \Delta \Gamma_i^{out,\alpha}(\vec{x}, t) &= 2t_{p,i} \frac{V_i^\alpha(\vec{x}, t)}{\Delta V(\vec{x}, t)} M(\vec{x}, t) [\vec{u}_b(t) \cdot \vec{c}_i] \Big/ c_s^2 \\ &\quad [\vec{c}_i \cdot \vec{n}^\alpha(t) > 0], \end{aligned} \quad (2.9)$$

with $M(\vec{x}, t^*) = \sum_i N_i(\vec{x}, t^*)$. Thus the extra mass, originating from all contributing facets to cell $\vec{x} + \vec{c}_i \Delta t$ and moving in direction i reads

$$B_i(\vec{x} + \vec{c}_i \Delta t, t) = \sum_\alpha \Delta \Gamma_i^{out,\alpha}(\vec{x} + \vec{c}_i \Delta t, t) \quad [\vec{c}_i \cdot \vec{n}^\alpha(t) > 0]. \quad (2.10)$$

Following Aidun *et al.* [5], the total mass $M(\vec{x}, t^*)$ in cell \vec{x} on $t = t^*$ is used in equation (2.9). This mass reads

$$M(\vec{x}, t^*) = \frac{\tilde{M}(\vec{x}, t^*)}{1 - 2 \sum_\alpha \sum_{i \in \vec{c}_i \cdot \vec{n}^\alpha(t) > 0} \{V_i^\alpha(\vec{x}, t) / \Delta V(\vec{x}, t) t_{p,i} [\vec{u}_b(t) \cdot \vec{c}_i] / c_s^2\}}, \quad (2.11)$$

where $\tilde{M}(\vec{x}, t^*)$ represents the total mass in cell \vec{x} after the collision and propagation steps but without the extra mass term $B_i(\vec{x} + \vec{c}_i \Delta t, t)$.

From the mass distribution in the boundary cells, the net flux of momentum, and therefore the net forces on each facet S^α can be calculated following

$$\begin{aligned} \vec{F}^\alpha(t) &= \frac{1}{\Delta t A^\alpha} \left[\sum_{i \in \vec{c}_i \cdot \vec{n}^\alpha(t) \leq 0} \vec{c}_i \Gamma_i^{in,\alpha}(t) \right. \\ &\quad \left. - \sum_{i \in \vec{c}_i \cdot \vec{n}^\alpha(t) > 0} \vec{c}_i [\Gamma_i^{out,\alpha}(t) + \Delta \Gamma_i^{out,\alpha}(t)] \right]. \end{aligned} \quad (2.12)$$

2. Adjustment of the mass distribution function in boundary cells

After the collision and propagation steps on $t = t^*$, the object is shifted over a finite distance $\Delta \vec{x} = \vec{u}_b(t) \Delta t$. Concerning this finite displacement of the object, three issues must be considered with respect to the mass balance in the boundary cells. (i) The volume of the boundary cells may change. (ii) New cells may appear and cells (and thus mass) may disappear. (iii) In contrast to the method of Ladd, the addition of $B_i(\vec{x} + \vec{c}_i \Delta t, t)$ is not balanced by a grid cell inside the object. These issues are discussed below.

First, the volume of boundary cells may change when the surface is shifted. Not adjusting the mass in these cells in accordance with the volume change during one time step

may result in large unphysical density fluctuations (and therefore pressure fluctuations). The mass in the boundary cells has to be adjusted such that the density in the cells is not affected by the discrete displacement of the surface. Hence

$$N_i(\vec{x}, t + \Delta t) = \frac{\Delta V(\vec{x}, t + \Delta t)}{\Delta V(\vec{x}, t)} N_i(\vec{x}, t^*). \quad (2.13)$$

Second, new cells may appear and some cells may disappear during one time step. As proposed by Aidun *et al.* [5], new cells are filled with the equilibrium mass distribution, based on the surface velocity $\vec{u}_b(t)$ and the averaged density of the surrounding cells. For new cells, the mass distribution reads

$$N_i(\vec{x}_{fluid}, t + \Delta t) = N_i^{eq}(\vec{x}_{fluid}, t + \Delta t). \quad (2.14)$$

This may result in small physical errors, because the physical state of the fluid close to the surface is far from equilibrium. However, Aidun *et al.* found that applying the equilibrium distribution does not result in significant nonphysical fluctuations, although staircase shaped surfaces were used. We might expect that using the equilibrium distribution with the volumetric method presented in this paper will result in even smaller errors. When a new cell appears, its volume is very small. This implies a very small contribution of this cell to the volumetric bounce-back process. As a result, we might expect that the small nonphysical effects die away quickly after a few time steps.

The mass in the disappearing cells is considered to be lost,

$$N_i(\vec{x}_{object}, t + \Delta t) = 0. \quad (2.15)$$

Third, in contrast to the method of Ladd [Eq. (2.7)] cells inside the object are not taken into account. Mass, represented by Eq. (2.10), is added to (or removed from) boundary cells outside the surface, which is not balanced by an inner grid cell. Consequently, mass in the computational domain is not *a priori* conserved globally.

Referring to the three issues concerning the mass balance in boundary cells, mass is added and removed, dependent on the position of the boundary cells with respect to the orientation of the surface. For closed surfaces (i.e., objects), the global mass change is very small. When density differences are small around the object, the mass added at one side is almost equal to the mass removed at the other side. It has been found that application of Eqs. (2.8), (2.13), (2.14), and (2.15) does not produce a significant extra amount of mass (see also Sec. III B).

3. Steps of the method

For each time step, the flow field around a moving object is calculated according to the following procedure.

(1) Calculate the geometrical quantities $\Delta V(\vec{x}, t)$, $\Phi_i^\alpha(t)$, $V_i^\alpha(\vec{x}, t)$, and $n^\alpha(t)$. In cases where the trajectory of the object is known, these quantities can also be determined and stored before carrying out the simulations.

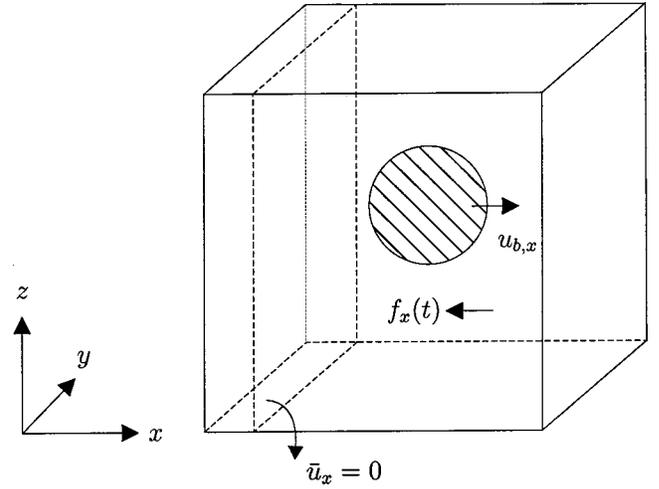


FIG. 3. Geometry moving with velocity $u_{b,x}$ in a fully periodic computational domain. A time-dependent body force $\vec{f}(t) = [f_x(t), 0, 0]$ enforced the average fluid velocity in the x direction \bar{u}_x to be equal to zero at every cross-sectional area of the flow field.

(2) Calculate the flow field for a fixed object, using the no-slip boundary technique described in Sec. II B.

(3) Calculate $N_i(\vec{x}, t^*)$ in the boundary cells with Eq. (2.8) by adding the new term $B_i(\vec{x} + \vec{c}_i \Delta t, t)$ to the mass distribution.

(4) Move the object over a distance $\Delta \vec{x} = \vec{u}_b(t) \Delta t$.

(5) Calculate the mass distribution $N_i(\vec{x}, t + \Delta t)$ in the boundary cells according to Eqs. (2.13), (2.14), and (2.15).

III. SIMULATIONS

A number of simulations at $Re=0.5$ and $Re=50$ were performed to study the numerical consistency of the method and to compare the results with physical reality. All simulations relate to a periodic array of cubes or spheres, moving with a constant velocity $\vec{u}_b = [u_{b,x}, 0, 0]$. The computational domain contained $N_x \times N_y \times N_z$ grid cells and was periodic to all sides (see Fig. 3). A time-dependent body force $\vec{f}(t) = [f_x(t), 0, 0]$ enforced the average fluid velocity in the x direction \bar{u}_x to be equal to zero at every cross-sectional area of the flow field (i.e., all fluid nodes in the y and z directions for a certain value of x). This body force was updated at every time step according to $\vec{f}(t + \Delta t) = [\alpha \bar{u}_x + \beta f_x(t), 0, 0]$. An adequate rate of convergence was reached for $\beta=0.95$ and $\alpha = -0.5 (Re=0.5)$ or $\alpha = -0.005 (Re=50)$. The time necessary to assume the system to be converged was roughly equal to $3N_x/u_{b,x}$ time steps.

For the validation of the numerical consistency, a moving array of cubes was chosen. The order of accuracy, the tendency towards grid independence, global mass conservation, and the dependence on the size of the facets were studied. In addition, comparisons were made to a *fixed* array of cubes immersed in a flow with a constant cross-sectional velocity $\bar{u}_x = -u_{b,x}$. A cube was chosen here because the shape of a cube is not affected by the size of the facets.

TABLE I. Overview of the tests performed on numerical consistency.

Case	Objective	Re	$u_{b,x}$	ν	D	$N_x \times N_y \times N_z$	Δ_f
1	Mass conservation	0.5,50	$\frac{1}{1728}, \dots, \frac{1}{25}$	$\frac{1}{96}, \dots, \frac{2}{3}$	4, ..., 32	$16^3, \dots, 128^3$	0.3, ..., 5.0
2.1	Moving vs fixed	0.5	$\frac{1}{96}$	$\frac{1}{6}$	8	32^3	1.0
2.2	Moving vs fixed	50	$\frac{1}{25}$	$\frac{4}{625}$	8	$96 \times 32 \times 32$	1.0
3	Drag force test	0.5,50	$\frac{1}{1728}, \dots, \frac{1}{25}$	$\frac{1}{96}, \dots, \frac{2}{3}$	4, ..., 32	$16^3, \dots, 128^3$	0.3, ..., 5.0
4	Grid independence and accuracy	0.5	$\frac{1}{48}, \dots, \frac{1}{384}$	$\frac{1}{6}$	4, ..., 32	$16^3, \dots, 128^3$	1.0

For physical validation, a moving array of spheres was chosen. Extensive and detailed experimental data, analytical solutions, and reliable computational data on the flow field properties around a periodic array of spheres and a single sphere are available in literature (see Refs. [15–18] and references therein). The data comprise drag coefficients, streamline patterns, and pressure fields.

A. Numerical setup

The Navier-Stokes equation was solved by means of the lattice-Boltzmann method. The scheme used in this paper was a 15-speed Bhatnagar-Gross-Krook (BGK) model [19] with a collision operator

$$\Omega_i(\vec{x}, t) = -\omega[N_i(\vec{x}, t) - N_i^{eq}(\vec{x}, t)]. \quad (3.1)$$

For all simulations, $\Delta x = \Delta y = \Delta z = 1$ and $\Delta t = 1$. The discretized particle velocities \vec{c}_i are defined as follows:

$$\vec{c}_i = \begin{cases} (0,0,0) & \text{for } i=0, \\ (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1) & \text{for } i=1, \dots, 6, \\ (\pm 1, \pm 1, \pm 1) & \text{for } i=7, \dots, 14. \end{cases} \quad (3.2)$$

In the BGK scheme used here, the equilibrium distribution N_i^{eq} is equal to

$$N_i^{eq}(\vec{x}, t) = t_{p,i} M(\vec{x}, t) \left(1 + 3(\vec{c}_i \cdot \vec{u}) + \frac{9}{2}(\vec{c}_i \cdot \vec{u})^2 - \frac{3}{2}u^2 \right), \quad (3.3)$$

where $t_{p,i} = \frac{2}{9}$ for $i=0$, $t_{p,i} = \frac{1}{9}$ for $i=1, \dots, 6$, and $t_{p,i} = \frac{1}{12}$ for $i=7, \dots, 14$.

The speed of sound for this scheme is equal to $c_s = \sqrt{1/3}$. The pressure is calculated following the equation of state $p = \rho c_s^2$. The viscosity amounts to $\nu = \frac{1}{6}(2/\omega - 1)$. The absolute value of the velocities $|\vec{u}|$ in this paper did not exceed 0.15; this constraint is necessary to approximately solve the incompressible Navier-Stokes equation with the lattice-Boltzmann scheme.¹

¹All quantities in this paper were expressed in terms of the lattice units (l.s. lattice spacing) and (l.t. lattice time). Both l.s. and l.t. equal unity.

The grid used had a maximum size of $128 \times 128 \times 128$ cells. The surface of the geometry (cube or sphere) was defined by the triangular facets with edges of roughly equal size Δ_f (see Fig. 1). The effect of the size of the facets on the flow field was studied by varying Δ_f between 0.3 l.s. and 5.0 l.s.

The code was written in FORTRAN 77. The simulations were carried out on a Pentium III 700 MHz processor with 1 GB of memory capacity. A simulation of a moving object consisting of 360 facets, in a grid of 1.28×10^6 grid cells, took about 25 h (wall clock time) for 20 000 time steps.

B. Numerical aspects

For studying the numerical aspects of the method, a moving array of cubes was chosen. The size of the cube D , its velocity, the size of the computational domain with volume $V_{domain} = N_x \times N_y \times N_z$ and the viscosity were varied. Two Re numbers were used: $\text{Re} = u_{b,x} D / \nu = 0.5$ and $\text{Re} = 50$. An overview of the tests performed can be found in Table I.

Case 1 refers to tests on global mass conservation, and is necessary because the method does not conserve mass explicitly (see Sec. II C 2). Comparison between simulations with a fixed and a moving periodic array of cubes at $\text{Re} = 0.5$ showed that the increase of mass was of the same order of magnitude in both cases, though the increase was somewhat larger for a moving array. For all simulations performed in this paper, the extra mass created per time step was in the order of 10^{-3} . No significant dependence on the velocity of the cube, on its size, the viscosity, and the size of the domain was found. It may be concluded that the method described in this paper does not produce a significantly extra amount of mass and that the increase in mass may be attributed to numerical noise.

Although the mass increase is very small, pressures can be corrected for the change of total mass in the computational domain. Especially when pressure fields from different simulations are compared, a correction may be necessary. For a change in the total mass M_{tot} with ΔM_{tot} , the average density in the domain equals $\rho_{corr} = (M_{tot} + \Delta M_{tot}) / V_{domain}$, thus the pressure in each grid cell is $p = (\rho - \rho_{corr}) c_s^2$.

In case 2, the steady-state flow field of a moving array of cubes was compared with that of a fixed array at $\text{Re} = 0.5$ and $\text{Re} = 50$. The pressure field, drag force, and velocity field

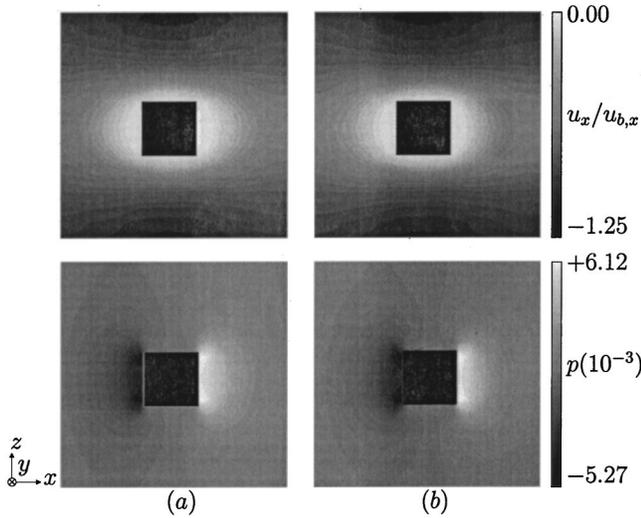


FIG. 4. Pressure and velocity contour plots around a fixed (a) and a moving (b) cube at $Re=0.5$. $D=8$, $N_x=N_y=N_z=32$, $\nu=\frac{1}{6}$, and $u_{b,x}=\frac{1}{96}$. In the case of a moving cube, the velocity field was transformed by $u_x \rightarrow u_x - u_{b,x}$. For the fixed cube, the average cross-sectional velocity was equal to $\bar{u}_x = -u_{b,x}$.

were studied. The flow field around the fixed array was calculated with the method for fixed objects (described in Sec. II B). Therefore, deviations between both cases may show additional effects caused by the extra term in Eq. (2.8) and the mass equations (2.13), (2.14), and (2.15). For the fixed array, the cross sectional average velocity was $\bar{u}_x = -u_{b,x}$. By applying the transformation $u_x \rightarrow u_x - u_{b,x}$ in the case of a moving array of cubes, both flow fields could be compared at a steady-state situation.

Case 2.1, where $Re=0.5$, showed practically identical pressure contours for the moving and fixed arrays. The velocity profiles were almost identical as well, with very small deviations close to the surface of the cube (Fig. 4). The drag forces in both cases differed by 0.3% only.

In case 2.2 ($Re=50$), small pressure fluctuations with a frequency $\tau^{-1} = u_{b,x}/\Delta x = \frac{1}{25}$ were observed for the moving array of cubes, which were not present in the case of a fixed array. These fluctuations are a numerical artifact and relate to the varying volume of the boundary cells and the surface velocity \vec{u}_b . For boundary cells with $\Delta V(\vec{x}, t)$ close to 0 or 1, the pressure in these cells was somewhat lower than for $\Delta V(\vec{x}, t)$ close to 0.5. In addition, the amplitude of these pressure fluctuations appeared to depend on the surface velocity $u_{b,x}$. When the pressure field was compared to cases with the same Re number and geometrical setup, but with a lower velocity, it was found that increasing the surface velocity increases the amplitude. This might explain the fact that these fluctuations were not observed in case 2.1, where the velocity was significantly lower ($u_{b,x} = \frac{1}{96}$). When the pressure field was averaged over τ time steps, a pressure field very similar to that around a fixed cube was obtained.

The velocity field of the moving cube was almost identical to that of the fixed cube, in spite of the small pressure fluctuations. To study the influence of the pressure fluctuations on the velocity field, the pressure gradient term

$-(1/\rho)(dp/dx)$ in the Navier-Stokes equation was examined. These local pressure gradients, which represent the fluctuating forces, caused a fluid element to be accelerated during $\frac{1}{2}\tau$ time steps and then decelerated during the same number of time steps. This acceleration (deceleration) resulted in an increase (decrease) $\Delta u_x \sim -(1/\rho)(dp/dx)\frac{1}{2}\tau$ of the local velocity u_x . For the simulation carried out here, the relative change in velocity compared to the local velocity amounted to $\Delta u_x/u_x < 1\%$. Moreover, fluid elements were accelerated and then decelerated, causing the net velocity change to be even smaller.

The drag force on the cube showed oscillations with the same frequency τ^{-1} as the pressure field and the same dependence on the surface velocity and the position of the surface relative to the grid cells. However, when the drag force under steady-state conditions was averaged over a relatively large period ($\Delta t \gg \Delta x/u_{b,x}$), the drag force on the moving cube deviated only 0.4% from the drag force on a fixed cube.

In case 3, the equation for the total force acting on each facet [Eq. (2.12)] was tested by comparing it to the forces derived from the total force balance in the domain. According to Eq. (2.12), the total force acting on the cube amounts to $\vec{F}_{cube}(t) = \sum_{\alpha} \vec{F}^{\alpha}(t)$. Because $\vec{F}_{cube}(t)$ is the sum of the drag force and the buoyancy force [caused by the body force $f_x(t)$ acting on the fluid], the drag force reads $\vec{F}_{d,1}(t) = \vec{F}_{cube}(t)/(1 - D^3/V_{domain})$.

The drag force can also be derived from the total force balance. The force balance in the domain reads $\vec{F}_{d,2}(t) + \vec{F}_{buoyancy} + \vec{F}_{body} = 0$, hence the drag force is $\vec{F}_{d,2}(t) = V_{domain}\vec{f}(t)$.

For most simulations performed in this paper, the drag forces $\vec{F}_{d,1}(t)$ and $\vec{F}_{d,2}(t)$ were compared. For all simulations, the deviation was smaller than 0.1%. We may therefore conclude that Eq. (2.12) accurately describes the force acting on a moving object.

In case 4, grid independence and accuracy in the drag force and velocity were examined. To this end, the resolution of the grid, the size of the domain $N_x = N_y = N_z = L$, the size of the cube, and its velocity were varied, while the Re number, the kinematic viscosity, the size of the facets Δ_f , and the geometrical ratio D/L were kept constant ($D/L = \frac{1}{4}$). The drag force F_D on the cube and the velocity on several fixed points in the domain were recorded until a steady-state situation was reached. The smallest grid comprised $N_x \times N_y \times N_z = 16 \times 16 \times 16$ cells, the largest $128 \times 128 \times 128$ cells. The accuracy was calculated for the drag force F_d acting on the cube and for the velocity $u(\vec{x}) = |\vec{u}(\vec{x})|$. The error in the drag force $E_d = (F_d - F_d^*)/F_d^*$ and the error in the velocity, $E_u = [u(\vec{x}) - u^*(\vec{x})]/u^*(\vec{x})$, were plotted on logarithmic scales against the relative grid spacing $\Delta x/N_x = 1/N_x$. F_d^* and $u^*(\vec{x})$ represent the drag force and the velocity, respectively, both calculated on the grid with the highest resolution ($L = L^* = 128$). The parameter E_d showed a deviation of 0.08% from the second highest resolution; for E_u , the deviation amounted to 0.1%. The order of accuracy was determined by calculating the slope, which is about 2.8 for E_d

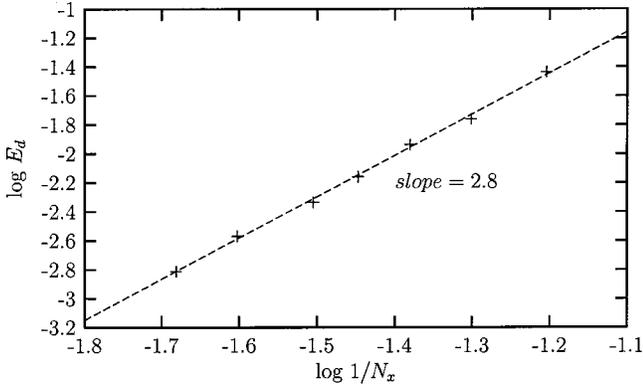


FIG. 5. Relative error in the drag force on a periodic array of cubes for different grid resolutions ($\text{Re}=0.5$, $\nu=\frac{1}{6}$, and $\Delta_f=1.0$). The geometrical ratio D/L was kept constant ($D/L=\frac{1}{4}$), hence $u_{b,x}=\Delta x^2/3L\Delta t$. Grid sizes in the range from $L=16$ to $L=128$ were used.

(Fig. 5) and 2.2 for E_u (Fig. 6). In contrast to the error in the drag force, the values of E_u did not fall on a straight line. Because the velocity converged more slowly to the real (physical) solution than the drag force, calculations at even higher resolutions should be performed. Nevertheless, as the slope did not change significantly for $L^*\geq 64$, the slope may remain about 2 for $L^*>128$ (Fig. 7). It may be concluded that the simulations exhibit second-order accuracy in the velocity. Apparently, the accuracy of the lattice-Boltzmann scheme in the fluid is not degraded to a lower order by the boundary conditions. The simulations show third-order accuracy in the drag force and show a clear tendency towards grid independence.

C. Physical consistency

In order to compare the simulations with physical reality, two cases at different Re numbers were selected ($\text{Re}=0.5$ and $\text{Re}=50$). In both cases, calculations were performed for a periodic array of spheres under steady-state conditions. The

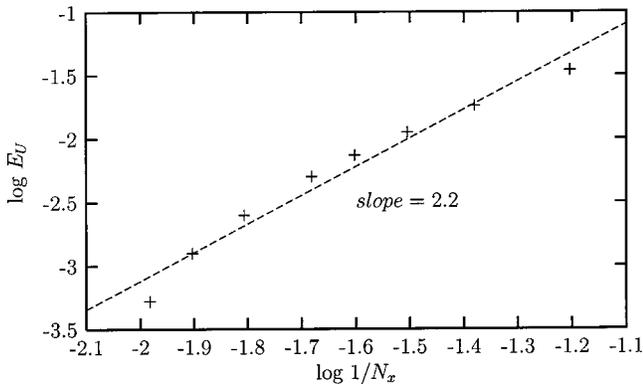


FIG. 6. Relative error in the velocity for a periodic array of cubes at different grid resolutions. The order of accuracy amounted to 2.2, though the points were not situated exactly on a straight line. However, the use of higher resolutions will not result in a significant change of the slope (see also Fig. 10), hence the order of accuracy was approximately 2.

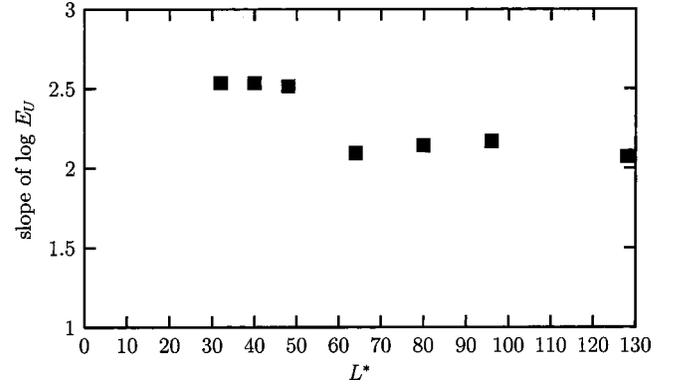


FIG. 7. Slope of the relative error in the velocity for a periodic array of cubes for different maximum grid sizes L^* (e.g., for $L^*=48$, only the velocities for the grid sizes $L=16, 24, 32$, and 40 are taken into account). For $L^*\geq 64$, the slope remained approximately 2.

diameter D , the velocity $u_{b,x}$ of the sphere, the size of the facets Δ_f , and the size of the computational domain $N_x \times N_y \times N_z$ were varied (see Table II).

In case 5, the drag force on a periodic cubic array of moving spheres at $\text{Re}=0.5$ was compared with Hasimoto's analytical solution of the drag force for a dilute cubic array of spheres at Stokes flow [15]. The analytical drag force on a cubic array of moving spheres with diameter D and velocity $u_{b,x}$, in a domain of size $N_x=N_y=N_z=L$ and with a cross-sectional average velocity $\bar{u}_x=0$, reads

$$F_{d,Has} = 3\pi\rho\nu Du_{b,x} (1 - 1.7601\sqrt[3]{\phi} + \phi - 1.5593\phi^2 + \dots)^{-1}, \quad (3.4)$$

where $\phi = \pi D^3/6L^3$ represents the volume fraction of spheres in the array.

For the simulations, spheres consisting of triangular facets S^α were used. The corners of the facets were situated at a distance D from the center of the triangulated sphere. The equivalent diameter of such a triangulated sphere can be derived from its surface and reads

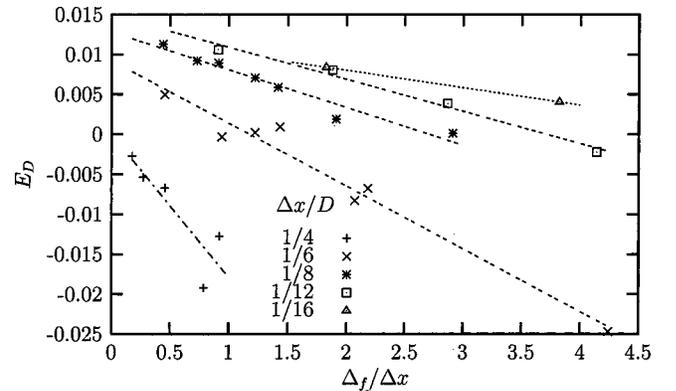


FIG. 8. The relative deviation of the hydrodynamic diameter from the equivalent diameter of a sphere in a cubic array for different grid spacings $\Delta x/D$ and dimensionless facet sizes $\Delta_f/\Delta x$ ($\text{Re}=0.5$, $\nu=\frac{1}{6}$, $\phi=8.18\times 10^{-3}$).

TABLE II. Overview of the tests on physical consistency.

Case	Objective	Compared with	Re
5.1	Numerical effect of ν	Hasimoto [15]	0.5
5.2	Dependence on Δx , Δ_f		
5.3	Drag for a range of ϕ		
6.1	Streamline pattern	Taneda [16]	50
6.2	Drag force	Roos and Willmarth [17]	
6.3	Pressure field	Johnson and Patel [18]	

$$D_A = \sqrt{\frac{\sum_{\alpha} A^{\alpha}}{\pi}}. \tag{3.5}$$

To simulate Stokes flow properly, a suitable Re number had to be found first. For practical reasons, a velocity $u_{b,x}$ as high as possible was preferred, because this reduces the number of positions of the sphere that needs to be calculated. The viscosity was equal to $\frac{1}{6}$, hence a Re number as high as possible had to be found. For $D=8$, $L=32$, and $\Delta_f=2.0$, simulations were performed for Re numbers ranging from 0.1 to 1.0. No significant change in the drag force was observed for $Re \leq 0.5$. Hence, $Re=0.5$ was small enough to consider the flow to be in the Stokes regime.

When the simulated drag force $F_{d,sim}$ was compared with the analytical drag force $F_{d,Has}$, dependence was observed on the grid spacing $\Delta x/D$, the size of the facets Δ_f and the viscosity ν . The influence of these numerical effects had to be investigated first. Then, $F_{d,sim}$ and $F_{d,Has}$ were compared for a range of volume fractions.

In case 5.1, the dependence on the viscosity at $Re=0.5$ was studied. It is known that the bounce-back principle results in an unphysical dependence on the viscosity, which depends on the geometry of the surface (see, e.g., Refs. [7,9]). ν was varied within a range of $\frac{1}{96}$ to $\frac{2}{3}$. The drag force at small viscosities resulted in a too high value of $F_{d,sim}$, while at high viscosities the simulated drag force was too low. The results are shown in Table III. For $\nu = \frac{1}{6}$, the deviation was very small. This tendency was also found by Ladd [3], who compared the volume averaged diameter of the sphere with the hydrodynamic diameter D_{hydr} , resulting

TABLE III. Influence of the viscosity ν on the simulated drag force $F_{d,sim}$ and the hydrodynamic diameter D_{hydr} of the sphere ($D=9$, $D_A=8.95$, $\Delta_f=1.0$, $Re=0.5$, $L=32$).

ν	$F_{d,sim}$	$F_{d,Has}$	D_{hydr}
$\frac{1}{96}$	7.770×10^{-3}	6.679×10^{-3}	9.88
$\frac{1}{21}$	0.1530	0.1396	9.52
$\frac{1}{12}$	0.4523	0.4275	9.32
$\frac{1}{6}$	1.720	1.710	9.04
$\frac{1}{3}$	6.229	6.839	8.48
$\frac{2}{3}$	20.37	27.36	7.42

from the simulations. The hydrodynamic diameter can be obtained with Eq. (3.4), by inserting the simulated drag force.

In case 5.2, the dependence on the grid spacing and the facet size was studied. This was done by plotting the relative deviation from the equivalent diameter $E_D = (D_{hydr} - D_A)/D_A$ for different values of $\Delta x/D$ and Δ_f [$D/L = \frac{1}{4}$, $u_{b,x} = (\Delta x)^2/3L\Delta t$, $\nu = \frac{1}{6}$]. For $\Delta_f = 2.0$, the grid spacing was varied within the range of $\frac{1}{28}$ to $\frac{1}{4}$. It was found that E_D was constant for $\Delta x/D < \frac{1}{16}$. Therefore, E_D was determined for $\frac{1}{16} < \Delta x/D < \frac{1}{4}$ for other values of Δ_f . The results for a range of $\Delta x/D$ and Δ_f are shown in Fig. 8. E_D depends linearly on the facet size, the slope of this linearity varying with the grid spacing. A larger value of $\Delta x/D$ resulted in a steeper slope. Thus, for small geometries (relative to the grid spacing), the effect of reducing the facet size on the calculations was stronger than in the case of large ones. From Fig. 8 we also might expect that for $\Delta x/D < \frac{1}{16}$ and $\Delta_f \rightarrow 0$, E_D is more or less independent of the grid spacing and the facet size. In this case, the remaining deviation amounted to approximately 1.5%. This deviation is likely to be caused by the unphysical dependence on the viscosity, originating from the bounce-back principle applied in the method presented in this paper. The question remains, however, *to what extent* the viscosity is responsible for this deviation. For practical situations, the equivalent diameter D_A can be corrected for this effect by introducing a hydrodynamic diameter D_{hydr} . This hydrody-

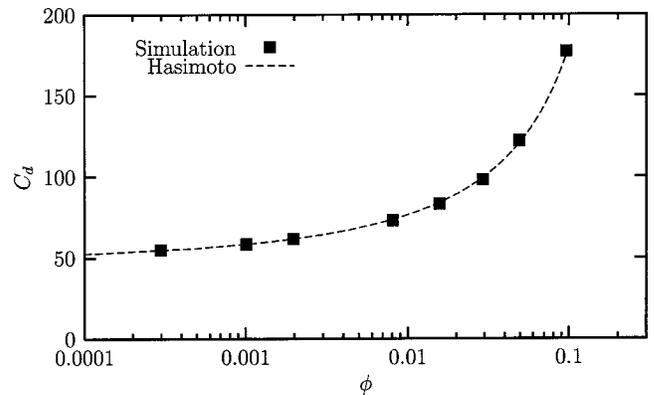


FIG. 9. Comparison of the analytically obtained and simulated drag force [in terms of drag coefficients $C_d = F_d / (\frac{1}{8} \rho u_{b,x}^2 \pi D^2)$] for a range of volume fractions in a cubic array of spheres. Simulations were carried out for different D , L , and $u_{b,x}$. The viscosity ν amounted to $\frac{1}{6}$ and Re was equal to 0.5.

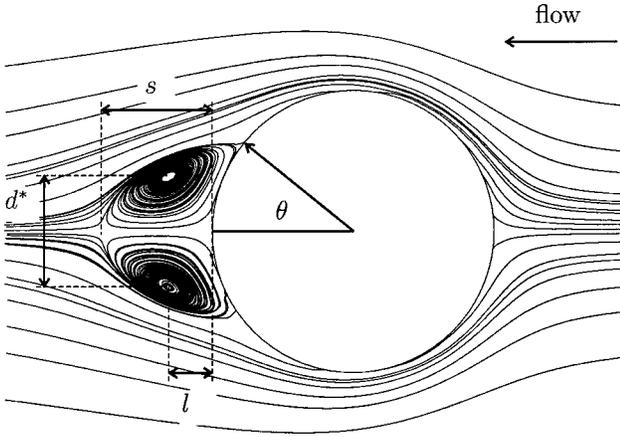


FIG. 10. The characteristic streamline pattern around a fixed sphere in an infinite flow field, resulting from a lattice-Boltzmann simulation at $Re=50$. The angle where the flow separates from the surface of the sphere θ , the separation length s/D , the distance between the two vortices d^*/D , and the distance l/D are shown.

dynamic diameter must be determined first from simulations similar to case 5.1.

In case 5.3, the analytically obtained drag force was compared with the force resulting from the simulations for a range of volume fractions ϕ (Fig. 9). The simulated drag force was not corrected by the introduction of a hydrodynamic diameter. It was found that the difference between the simulated and the analytical drag forces was smaller than 1.5% for volume fractions $\phi < 0.05$. For $\phi = 0.1$, the difference amounted to 3.8%, which may be due to Hasimoto's solution being only applicable to dilute systems.

In case 6, the flow field of an array of moving spheres at $Re=50$ was compared with the experimental and the simulation data on a fixed single sphere in a moving infinite medium [16–18]. The length of the grid (N_x) was increased to examine whether the flow field would approach the case of a single sphere because of the increasing distance between the spheres. At $Re=50$, the flow field is steady and axisymmetric. The flow field properties considered are the drag force F_d , the pressure field, and some characteristics of the streamline pattern (see Fig. 10). Simulations were performed for $D=20$, $u_{b,x} = \frac{1}{25}$, $\Delta_f = 2.0$, and $\nu = \frac{2}{125}$. The equivalent diameter D_A of the sphere was 19.92, where $Re=49.8$. The length of the grid ranged from $N_x=100$ to $N_x=300$. The other dimensions were fixed: $N_y=N_z=80$.

TABLE IV. Drag force and streamline characteristics for a sphere in flow field of different lengths N_x at $Re=49.8$.

N_x	F_d	C_d	s/D	d^*/D	l/D	θ
100	2.12	1.06	0.33	0.37	0.11	36
150	2.48	1.23	0.37	0.40	0.13	37
200	2.90	1.44	0.39	0.41	0.14	38
300	3.22	1.60	0.42	0.41	0.15	39
Infinite field	3.24	1.61 ^a	0.42 ^b	0.42 ^b	0.14 ^b	40 ^b

^aRoos and Willmarth [17].

^bTaneda [16].

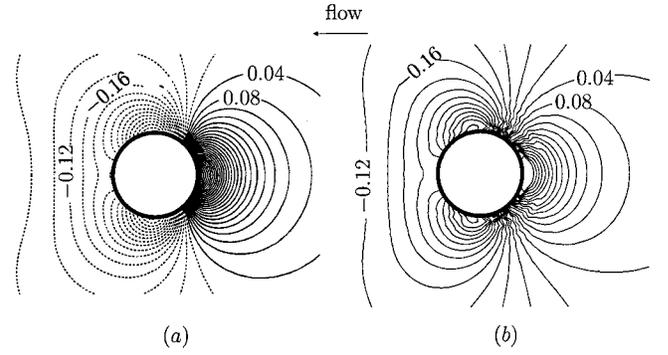


FIG. 11. Contour plots of the pressure coefficient around a sphere. (a) In an infinite field at $Re=50$ (Johnson and Patel). (b) In a field of size $N_x \times N_y \times N_z = 300 \times 80 \times 80$ at $Re=49.8$ (this paper). $D=20$, $D_A=19.92$, $u_{b,x} = \frac{1}{25}$, and $\nu = \frac{2}{125}$.

For the cases 6.1 and 6.2, it was found that, for $N_x = 300$, the drag force and the streamline properties showed good agreement with the experimental data from literature (see Table IV). In contrast to the moving array of spheres at $Re=0.5$ (case 5), no significant numerical effect of the viscosity on the hydrodynamic diameter was found. A possible explanation is that at $Re=49.8$, the effect of the viscous forces on the flow is small compared to that of the inertial forces.

For the pressure field (case 6.3), simulation data were used from Johnson and Patel [18] who performed very accurate calculations on the flow field properties around a single fixed sphere. They calculated the total pressure coefficient, $C_{p^*} = (p^* - p_{\infty}^*) / \frac{1}{2} \rho U_{\infty}^2$. Because of the presence of a body force, the total pressure coefficient in this paper was

$$C_{p^*} = \frac{p - f_x x - p_{\infty}^*}{\frac{1}{2} \rho u_{b,x}^2}, \quad (3.6)$$

with p_{∞}^* as a reference pressure. The total pressure coefficient resulting from the simulations showed a similar pattern [Fig. 11(b)] and a reasonable agreement with the results from Johnson and Patel [Fig. 11(a)]. The pressure field resulting from our simulations was somewhat more stretched, because of the relatively small distance between the spheres in the y and z directions. In this case, the velocity of the fluid is higher between the spheres and results in a local reduction of the pressure.

IV. CONCLUSIONS AND OUTLOOK

A method for moving objects in the lattice-Boltzmann schemes has been presented. In this method, objects of arbitrary shape and size can move through the grid independent of the position of the nodes. Simulations demonstrate a good agreement with real life concerning the drag force acting on the object, the velocity field, and the pressure. Results on the velocity for a moving array of cubes show second-order ac-

accuracy, the drag force acting on the cubes third-order accuracy in the grid spacing. The accuracy appeared to be independent of the position of the surface and therefore does not decrease the order of accuracy of the lattice-Boltzmann scheme itself (which is of second order). Next to the grid spacing, the resolution of the surface can be used as an additional parameter to increase the accuracy, because the surface resolution is decoupled from the grid resolution. Especially for small curved objects (compared to the grid spacing), the effect of the surface resolution on the accuracy is most profound.

The method develops small fluctuations in the pressure field that depends on the surface velocity. It is shown that the velocity field is not significantly affected by these fluctuations. The correct pressure field is obtained by averaging the pressure over the period of one fluctuation. While the hydrodynamic diameter depends on the viscosity at $Re=0.5$, this numerical artifact is not found in the case of $Re=50$; this might be due to the small impact of the viscosity on the flow for $Re \gg 1$. For lower Re numbers, a correction on the hydrodynamic diameter may compensate for this numerical artifact.

Our method can easily be extended to systems with freely moving objects, where the motion of the object is determined by the flow. With Eq. (2.12), the force acting on the object can be accurately calculated. As a result, the translational and rotational motion of a particle moving freely through the fluid can be simulated at great accuracy. In systems with hundreds of particles (e.g., suspensions of particles), our

method may be computationally too demanding because the geometrical quantities $\Delta V(\vec{x}, t)$, $\Phi_i^\alpha(t)$, $V_i^\alpha(\vec{x}, t)$, and $\vec{n}^\alpha(t)$ have to be recalculated for each time step. In the case of equally sized particles, however, these geometrical quantities can be calculated for a fixed number of positions and stored before the simulation is carried out, since only the positions *relative to the grid cells* have to be calculated [e.g., a particle at position $\vec{x}=(x_1, y_1, z_1)$ results in the same geometrical quantities as a particle at position $\vec{x}=(x_1 \pm 1, y_1 \pm 1, z_1 \pm 1)$].

For turbulent flow around moving surfaces (e.g., an impeller in a stirred vessel), an additional algorithm can be implemented, which accounts for the turbulent boundary layer at a solid wall [12]. This algorithm enables us to impose a tangential force at the surface as an arbitrary function of the flow field properties in the vicinity of the surface. In this way, wall functions can be applied.

Scalar transport around moving boundaries can be easily introduced in the method. Due to the exact definition of the surface, the scalar quantity can be accurately determined at the surface. No unphysical transport occurs from the fluid to the nodes inside the object, because the surface is closed. This enables us to treat the scalar transport inside the object separately if necessary.

ACKNOWLEDGMENTS

The authors would like to thank Dr. L.M. Portela and Dr. D. Kandhai for their valuable input and discussions.

-
- [1] J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics* (Springer-Verlag, Berlin, 1996).
- [2] H.H. Hu, *Int. J. Multiphase Flow* **22**, 335 (1996).
- [3] A.J.C. Ladd, *J. Fluid Mech.* **271**, 285 (1994).
- [4] A.J.C. Ladd, in *Dynamics: Models and Kinetic Methods for Nonequilibrium Many Body Systems*, edited by H. Van Beijeren and J. Karkheck (Kluwer Academic Publishers, Dordrecht, 2000).
- [5] C.K. Aidun, Y. Lu, and E.J. Ding, *J. Fluid Mech.* **373**, 287 (1998).
- [6] C.P. Lowe, D. Frenkel, and A.J. Masters, *J. Chem. Phys.* **103**, 1582 (1995).
- [7] X. He, Q. Zou, L-S. Luo, and M. Dembo, *J. Stat. Phys.* **87**, 115 (1997).
- [8] I. Ginzburg and P.M. Adler, *J. Phys. II* **4**, 191 (1994).
- [9] D. Kandhai, A. Koponen, A. Hoekstra, M. Kataja, J. Timonen, and P.M.A. Sloot, *J. Comput. Phys.* **150**, 482 (1999).
- [10] O. Filippova and D. Hänel, *J. Comput. Phys.* **147**, 219 (1998).
- [11] R. Verberg and A.J.C. Ladd, *Phys. Rev. Lett.* **84**, 2148 (2000).
- [12] H. Chen, C. Teixeira, and K. Molvig, *Int. J. Mod. Phys. C* **9**, 1281 (1998).
- [13] R. Mei, L-S. Luo, and W. Shyy, *J. Comput. Phys.* **155**, 307 (1999).
- [14] J.J. Derksen and H.E.A. Van den Akker, *AIChE J.* **45**, 209 (1999).
- [15] H. Hasimoto, *J. Fluid Mech.* **5**, 317 (1958).
- [16] S. Taneda, *J. Phys. Soc. Jpn.* **11**, 1104 (1956).
- [17] F.W. Roos and W.W. Willmarth, *AIAA J.* **9**, 285 (1971).
- [18] T.A. Johnson and V.C. Patel, *J. Fluid Mech.* **378**, 19 (1999).
- [19] Y.H. Qian, D. d'Humieres, and P. Lallemand, *Europhys. Lett.* **17**, 479 (1992).