Engineering with Computers (2022) 38:5507–5518
https://doi.org/10.1007/s00366-022-01671-0

ORIGINAL ARTICLE

Finite volume scheme for the lattice Boltzmann method on curved surfaces in 3D

Junxiang Yang[1] · Zhijun Tan[1,2] · Sangkwon Kim[3] · Chaeyoung Lee[3] · Soobin Kwak[3] · Junseok Kim[3]

Received: 10 January 2022 / Accepted: 29 April 2022 / Published online: 24 May 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

The fluid flows on the surface widely exist in the natural world, such as the atmospherical circulation on a planet. In this study, we present a finite volume lattice Boltzmann method (LBM) for simulating fluid flows on curved surfaces in three-dimensional (3D) space. The curved surfaces are discretized using unstructured triangular meshes. We choose the D3Q19 lattice and the triangular meshes for the velocity and spatial discretizations, respectively. In one time iteration, we only need to compute the distribution functions on each vertex in a fully explicit form. Therefore, our proposed method is highly efficient for solving fluid flows on curved surfaces using LBM. To keep the velocity field tangential to the surfaces, a practical velocity correction technique is adopted. We perform a series of computational experiments on various curved surfaces such as sphere, torus, and bunny to demonstrate the performance of the proposed method.

Keywords Finite volume scheme · Lattice Boltzmann method · 3D curved surfaces · Fluid flows

1 Introduction

The computational fluid dynamics (CFD) plays an important role in various industrial and scientific fields. The core of CFD is to solve the Navier–Stokes (NS) equations for incompressible fluid flows using proper numerical methods. Compared with a large amount of research on the numerical methods for simulating fluid flows in two- and three-dimensional spaces, the CFD problems on curved surfaces have been less explored. The authors in [1] used the following equations to solve the incompressible surface NS equation by surface finite elements:

$$\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t} + \nabla_{\mathbf{u}}\mathbf{u}(\mathbf{x},t) = -\text{grad}_S p(\mathbf{x},t) + \frac{1}{Re}\left(-\Delta^{\text{dR}}\mathbf{u}(\mathbf{x},t) + 2\kappa\mathbf{u}(\mathbf{x},t)\right), \quad (1)$$

$$\text{div}_S \mathbf{u}(\mathbf{x},t) = 0, \ \mathbf{x} \in S, \quad t > 0, \quad (2)$$

where $\mathbf{u}(\mathbf{x},t)$ is the tangential surface velocity, $\nabla_{\mathbf{u}}$ is the covariant direction derivative, grad_S is the surface gradient, S is the surface, $p(\mathbf{x},t)$ is the surface pressure, Re is the surface Reynolds number, Δ^{dR} is the surface Laplace–deRham operator, κ is the Gaussian curvature, and div_S is the surface divergence. For more details about the definition of the terms, see [1] and references therein.

Mohamed et al. [2] solved the incompressible NS equation on surfaces using a discrete exterior calculus (DEC) method. They used a vorticity-stream function formulation based on Helmholtz decomposition; however, it misses the essential Gaussian curvature term in the equations. Nitschke et al. [3] developed an appropriate DEC approach in $\mathbf{u}(\mathbf{x},t)$ and $p(\mathbf{x},t)$ variables with the Gaussian curvature term. Nitschke et al. [4] developed a finite element method (FEM) for two-phase Newtonian surface fluid flows using a surface Cahn–Hilliard–Navier–Stokes equation and a stream function formulation. Reuther and Voigt [5] investigated the interplay of curvature and vortices in fluid flows on curved surfaces using a surface vorticity-stream function approach. Gross and Atzberger [6] formulated hydrodynamic equations and developed spectrally accurate computational schemes for solving surface fluid flows on radial manifolds. The method was based on the closet point method and solved

---

✉ Junseok Kim
cfdkim@korea.ac.kr

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China

[2] Guangdong Province Key Laboratory of Computational Science, Sun Yat-sen University, Guangzhou 510275, China

[3] Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

the equation on extended narrow band domain embedding the curved surface in 3D. Reuther et al. [1] used the surface FEM method for the fluid flows on curved surfaces. Based on the surface Helmholtz decomposition, Li et al. [7] developed a divergence-free radial kernel for the Stokes flow on the surfaces. Recently, a finite difference method (FDM)-based numerical method for the incompressible NS equation on curved surfaces was proposed in [8].

The above-mentioned methods only dealt with stationary surfaces. For evolving surfaces, Torres-Sánches et al. [9] developed a general continuum mechanics and computational method for fluid deformable surfaces, Nitsche et al. [10] presented the derivation and computational method of the flow of passive and active polar liquid crystals on evolving curved surfaces, and Reuther [11] developed a novel method to derive the governing equations of fluid deformable surfaces under the thin film limit.

In mesoscopic hypothesis, the lattice Boltzmann method (LBM) has been widely used for numerically simulating complex fluid flows due to its ease of modeling physical phenomena and simplicity of implementation [12, 13]. Using 3D LBM, non-Newtonian fluids [14], 3D lid-driven cavity flow on non-uniform meshes [15] and immiscible two-phase flow [16] were performed. Sadeghi [17] presented the 3D LBM for simulating multiphase penetration into the porous media numerically and investigated the effects of porosity, high density ratio, contact angle, and flow parameters. In [18], the authors developed the numerical model using LBM for solid-liquid phase change which is one of the common phenomena in physics.

Despite the advantages of LBM, there is a constraint that the discrete velocity direction is involved in the structure of the spatial grid. To overcome this, a finite volume formulation of LBM was developed. Peng et al. [19] proposed a finite volume LBM on 2D unstructured triangular meshes and verified its accuracy through several comparisons with other methods. In [20], Patil and Lakshmisha described a finite volume LBM on 2D unstructured mesh by generating any polygonal tessellation, including triangular mesh, in the domain and by creating the control volume using cell-centered unstructured finite volume discretizations. Here, the total variation diminishing scheme is adopted for the intercell advection of particle distribution function. For the nearly incompressible flows with complex boundaries, the authors in [21] developed a second-order accurate finite volume LBM by which the fluxes were computed in one step, whereas the previous method calculated over two steps. Wang et al. studied steady and unsteady flows on unstructured grids using the cell-center finite volume formulation and LBM, and performed incompressible laminar flow [22] and turbulent flow simulations [23].

The curved surfaces were mostly used as structures such as a capsule and a flapping flag in 3D fluid domain [24]. In [25], the authors studied the performances of boundary conditions in simulating boiling involving curved surfaces. Furthermore, the general atmospheric circulation on a planet can be assumed as the flow on a spherical surface because the thickness of atmosphere is much less than the radius of a planet. Therefore, it is physically meaningful to study the fluid flows on curved surfaces by using appropriate numerical methods.

To simulate the fluid flows on curved surfaces, we develop a finite volume scheme for the LBM on 3D curved surfaces. The curved surfaces are discretized using unstructured triangular meshes. We choose the D3Q19 lattice and the triangular meshes for the velocity and spatial discretizations, respectively. The proposed scheme is based on that of Peng [19] and Di Ilio [26]. However, to the best of the authors' knowledge, this is the first work focusing on an efficient finite volume 3D LBM for simulating fluid flows on 3D curved surfaces without flattening the 1-ring neighborhoods, rotating and aligning the velocity vectors, which were required in [27] where 2D LBM was used. Furthermore, we use a velocity correction procedure to have tangential velocity field on 3D curved surfaces.

The organization of this paper is as follows. In Sect. 2, we provide the finite volume LBM. We present the computational tests in Sect. 3. In Sect. 4, conclusions are drawn.

2 Finite volume lattice Boltzmann method

Let us consider the following equation which is based on the lattice Boltzmann equation using a finite volume method [19]:

$$\frac{\partial f_i}{\partial t}(\mathbf{x},t) + \mathbf{c}_i \cdot \nabla f_i(\mathbf{x},t) = -\frac{1}{\tau}\left(f_i(\mathbf{x},t) - f_i^{eq}(\mathbf{x},t)\right), \tag{3}$$

where $f_i(\mathbf{x},t)$ is the particle distribution function at space $\mathbf{x} = (x,y,z)$ and time t, $\mathbf{c}_i$ is the velocity in the ith direction for $i = 1,2,\ldots,19$ as shown in Fig. 1(c), τ is the relaxation time, and $f_i^{eq}(\mathbf{x},t)$ are the local equilibrium distributions.

Specific values of $\mathbf{c}_i$ are given as

$$\begin{aligned}
&\mathbf{c}_1 = (1,0,0),\ \mathbf{c}_2 = (-1,0,0),\ \mathbf{c}_3 = (0,1,0),\\
&\ \mathbf{c}_4 = (0,-1,0),\ \mathbf{c}_5 = (0,0,1),\\
&\mathbf{c}_6 = (0,0,-1),\ \mathbf{c}_7 = (1,1,0),\ \mathbf{c}_8 = (1,-1,0),\\
&\ \mathbf{c}_9 = (1,0,1),\ \mathbf{c}_{10} = (1,0,-1),\\
&\mathbf{c}_{11} = (-1,1,0),\ \mathbf{c}_{12} = (-1,-1,0),\\
&\ \mathbf{c}_{13} = (-1,0,1),\ \mathbf{c}_{14} = (-1,0,-1),\\
&\mathbf{c}_{15} = (0,1,1),\ \mathbf{c}_{16} = (0,1,-1),\\
&\ \mathbf{c}_{17} = (0,-1,1),\ \mathbf{c}_{18} = (0,-1,-1),\ \mathbf{c}_{19} = (0,0,0).
\end{aligned}$$

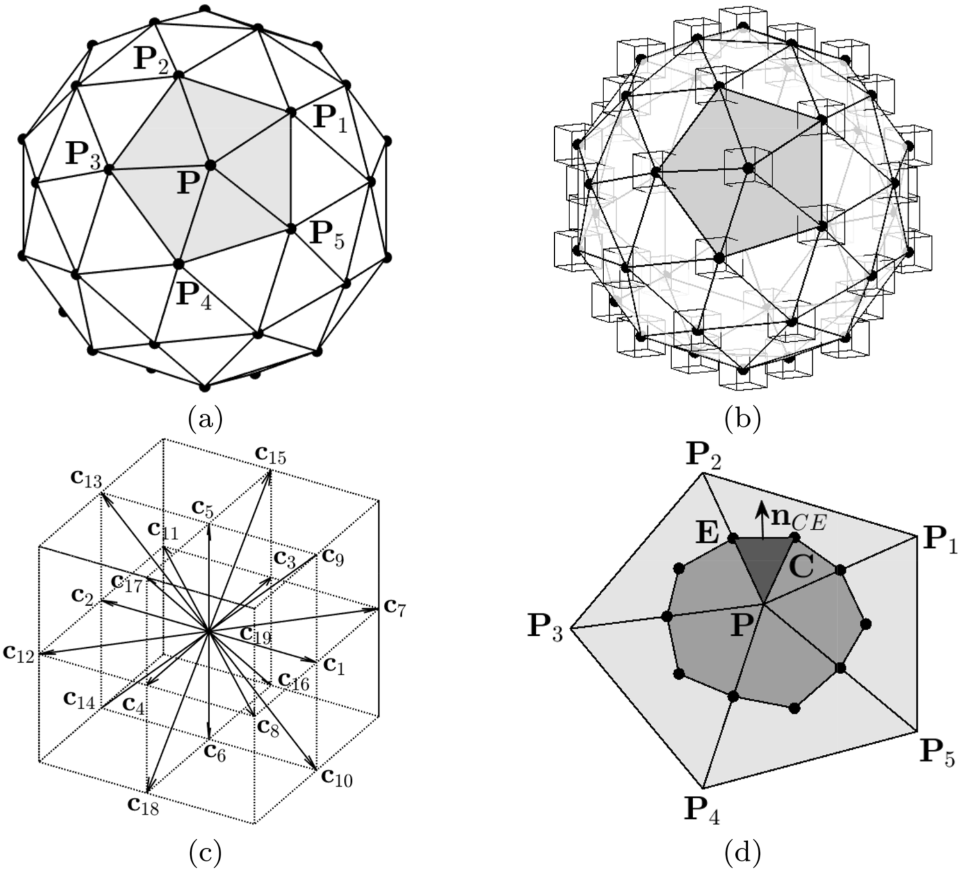


Fig. 1 Schematic representations of **a** a surface mesh with a one-ring neighborhood of point **P**, **b** D3Q19 at each point, **c** D3Q19 lattice, and **d** a control volume at point **P**

For $i = 1, \dots, 19$, the local equilibrium distributions are defined as

$$f_i^{eq}(\mathbf{x}, t) = w_i \rho(\mathbf{x}, t)\left[1 + \frac{\mathbf{c}_i \cdot \mathbf{u}(\mathbf{x}, t)}{c_s^2} + \frac{1}{2}\left(\frac{\mathbf{c}_i \cdot \mathbf{u}(\mathbf{x}, t)}{c_s^2}\right)^2 - \frac{|\mathbf{u}(\mathbf{x}, t)|^2}{2c_s^2}\right], \tag{4}$$

where the lattice speed of sound $c_s = 1/\sqrt{3}$ for the D3Q19 lattice and

$$\rho(\mathbf{x}, t) = \sum_{i=1}^{19} f_i(\mathbf{x}, t), \tag{5}$$

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho(\mathbf{x}, t)} \sum_{i=1}^{19} f_i(\mathbf{x}, t)\mathbf{c}_i \tag{6}$$

are the density and velocity, respectively, and the lattice weights are

$$w_i = \begin{cases} \dfrac{2}{36}, & i = 1, \dots, 6, \\[2mm] \dfrac{1}{36}, & i = 7, \dots, 18, \\[2mm] \dfrac{12}{36}, & i = 19. \end{cases} \tag{7}$$

Given surface S, we make a triangular surface mesh M as shown in Fig. 1a. Figure 1b illustrates a mesh grid point **P** of M and one-ring neighbor vertices of P, i.e., $P_1, P_2, \dots, P_5$. Figure 1(c) shows the points C and E defined as the center of an edge PP_2 and the centroid of a triangle $\triangle PP_1P_2$, respectively, for a control volume centered at a point **P**. The integration of Eq. (3) over the triangle $\triangle PCE$ results in

$$\int_{\triangle PCE} \frac{\partial f_i}{\partial t}(\mathbf{x}, t)d\sigma + \int_{\triangle PCE} \mathbf{c}_i \cdot \nabla f_i(\mathbf{x}, t)d\sigma = -\frac{1}{\tau}\int_{\triangle PCE} \left(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)\right)d\sigma. \tag{8}$$

The first term in Eq. (8) is approximated as

$$\int_{\triangle PCE} \frac{\partial f_i}{\partial t}(\mathbf{x},t)d\sigma = \frac{\partial f_i}{\partial t}(\mathbf{P},t)\int_{\triangle PCE} d\sigma = \frac{\partial f_i}{\partial t}(\mathbf{P},t)A_{\triangle PCE} = \frac{f_i(\mathbf{P},t+\Delta t)-f_i(\mathbf{P},t)}{\Delta t}A_{\triangle PCE}, \tag{9}$$

where $A_{\triangle PCE}$ is the area of triangle $\triangle PCE$ and $f_i(\mathbf{P},t)$ is the f_i value at node $\mathbf{P}$ and time t. The second term in Eq. (8) is approximated as

$$\begin{aligned}\int_{\triangle PCE} \mathbf{c}_i\cdot\nabla f_i(\mathbf{x},t)d\sigma &= \int_{\triangle PCE} \nabla\cdot[f_i(\mathbf{x},t)\mathbf{c}_i]d\sigma \\ &= \int_{\partial\triangle PCE} f_i(\mathbf{x},t)\mathbf{c}_i\cdot\mathbf{n}dl \\ &= \frac{f_i(\mathbf{P},t)+f_i(\mathbf{C},t)}{2}\mathbf{c}_i\cdot\mathbf{n}_{PC}l_{PC} \\ &\quad+\frac{f_i(\mathbf{C},t)+f_i(\mathbf{E},t)}{2}\mathbf{c}_i\cdot\mathbf{n}_{CE}l_{CE} \\ &\quad+\frac{f_i(\mathbf{E},t)+f_i(\mathbf{P},t)}{2}\mathbf{c}_i\cdot\mathbf{n}_{EP}l_{EP},\end{aligned} \tag{10}$$

where we have used the divergence theorem and $\mathbf{n}$ is the unit outward normal vector to the edges of the triangle $\triangle PCE$. Here, l_{CE} is the length of CE, and $\mathbf{n}_{CE}$ is the unit outward normal vector to the edge CE. That is, the outward normal vector is defined as

$$\mathbf{n}'_{CE} = \overrightarrow{PE} - \langle\overrightarrow{PE},\overrightarrow{CE}\rangle\frac{\overrightarrow{CE}}{|\overrightarrow{CE}|^2}$$

and the unit outward normal vector is $\mathbf{n}_{CE} = \mathbf{n}'_{CE}/|\mathbf{n}'_{CE}|$, see Fig. 1(d). Here, $\langle\mathbf{u},\mathbf{v}\rangle$ is the inner product of two vectors $\mathbf{u}$ and $\mathbf{v}$. The other terms are similarly defined. The values of $f_i(\mathbf{C},t)$ and $f_i(\mathbf{E},t)$ are interpolated as follows:

$$f_i(\mathbf{C},t) = \frac{f_i(\mathbf{P},t)+f_i(\mathbf{P}_1,t)+f_i(\mathbf{P}_2,t)}{3} \quad \text{and}$$
$$f_i(\mathbf{E},t) = \frac{f_i(\mathbf{P},t)+f_i(\mathbf{P}_1,t)}{2}.$$

The third term in Eq. (8) is approximated as

$$\begin{aligned}&-\frac{1}{\tau}\int_{\triangle PCE}\left(f_i(\mathbf{x},t)-f_i^{eq}(\mathbf{x},t)\right)d\sigma \\ &= -\frac{A_{\triangle PCE}}{\tau}\left(\frac{f_i(\mathbf{P},t)+f_i(\mathbf{C},t)+f_i(\mathbf{E},t)}{3}\right. \\ &\left.-\frac{f_i^{eq}(\mathbf{P},t)+f_i^{eq}(\mathbf{C},t)+f_i^{eq}(\mathbf{E},t)}{3}\right).\end{aligned} \tag{11}$$

Therefore, summing up Eqs. (9), (10), and (11) on the control volume, we get the following time-stepping scheme for $f_i(\mathbf{P},t)$, $i = 1,\dots,19$ and all $\mathbf{P}\in M$:

$$f_i(\mathbf{P},t+\Delta t) = f_i(\mathbf{P},t) + \frac{\Delta t}{A_{\mathbf{P}}}\left(\text{Collision}_{\mathbf{P}} - \text{Flux}_{\mathbf{P}}\right), \tag{12}$$

where $A_{\mathbf{P}}$, $\text{Collision}_{\mathbf{P}}$, and $\text{Flux}_{\mathbf{P}}$ are the total area, sum of Eq. (11), and sum of Eq. (10) of the control area at $\mathbf{P}$, respectively.

With the initially defined velocity field $\mathbf{u}(\mathbf{x},0)$, we can compute the initial equilibrium distribution function $f_i^{eq}(\mathbf{x},0)$ from Eq. (4), then we set the initial distribution function $f(\mathbf{x},0) = f_i^{eq}(\mathbf{x},0)$. Let f^n be the numerical approximation of function $f(\mathbf{x},t)$ at $t = n\Delta t$, the algorithm in one time iteration can be summarized as follows

Step 1. Compute each of flux terms from Eq. (10).

Step 2. Compute each of collision terms from Eq. (11).

Step 3. Update distribution function f_i^{n+1} from Eq. (12).

Step 4. Update the density function ρ^{n+1} and velocity field $\mathbf{u}^{n+1}$ from Eqs. (5) and (6), respectively.

Step 5. Correct the velocity field by $\mathbf{u}^{n+1} = \mathbf{u}^{n+1} - (\mathbf{u}^{n+1}\cdot\mathbf{n}^v)\mathbf{n}^v$.

Step 6. Update the equilibrium distribution $f_i^{eq,n+1}$ from Eq. (4).

In actual simulations, we want the velocity fields tangential to the curved surface. However, we notice that Steps 1–4 can not guarantee this property. Therefore, we propose a correction step (Step 5) to ensure that the updated velocity field is tangential to the surface all along, the previous studies [8, 28] showed that this technique worked well for fluid flows on curved surfaces. In Step 5, $\mathbf{n}^v$ represents the outward unit normal vector at each vertex, the definition is

$$\mathbf{n}^v = \frac{\sum_{q\in I(\mathbf{P})} w_q\mathbf{n}_q}{|\sum_{q\in I(\mathbf{P})} w_q\mathbf{n}_q|},$$

where $I(\mathbf{P})$ is the set of index of the triangles neighboring a vertex $\mathbf{P}$, $\mathbf{n}_q$ is the outward unit normal vector of triangle T_q, the weighting coefficient $w_q = |\mathbf{C}_q - \mathbf{P}|^{-2}$, and $\mathbf{C}_q$ is the coordinates of centroid of T_q [29], see Fig. 2. We should note that in the proposed finite volume LBM for simulating fluid flows on 3D curved surfaces, we do not flatten the 1-ring neighborhoods, rotate and align the velocity vectors, which were done in [27] where 2D LBM was used.

It is well known that in the small Knudsen and low Mach number limit of the Boltzmann equation, there is a relation between the Boltzmann equation and the incompressible Navier–Stokes system in the Cartesian coordinate, see [30] and references therein for more details. To the authors' knowledge, it is still an open problem to prove a relation between the surface Boltzmann equation and the surface NS system. It would be an interesting future research topic to prove that relation.

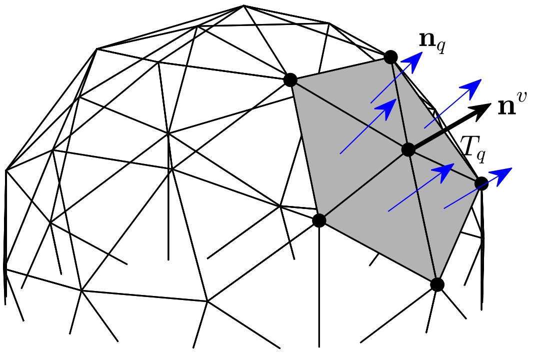


Fig. 2 Schematic illustration of the outward unit normal vector, where the black and dark blue arrows represent the vector on the vertex and centroid, respectively

3 Numerical results

In this section, we will validate our proposed numerical method by investigating fluid flows on various curved surfaces in 3D space.

3.1 Effect of velocity correction

In actual computation, the distribution function f_i of velocity field is updated in 3D space, which can not ensure the velocity field always be tangential to the surface. To fix this problem, we proposed a practical technique of velocity correction in Sect. 2. To validate the practicability of this technique, we consider the circumfluence on a sphere. The initial velocity field [Fig. 3(a)] is defined as

$$\mathbf{u}(\mathbf{x}, 0) = (0.01y,\ -0.01x,\ 0). \tag{13}$$

The average triangular mesh size $h_{ave} = 0.1257$, time step $\Delta t = 0.01$, and relaxation time $\tau = 1$ are considered. The spherical surface is defined by the zero level-set of the distance function $d(\mathbf{x}) = \sqrt{x^2 + y^2 + z^2} - R$, where $R = 10$ is used as the radius of sphere. The computations without and with velocity correction are performed until $t = 4$ and the numerical results are plotted in Figs. 3(b) and (c). We can observe that the velocity field passes through the surface if we do not perform the correction and this velocity field is nonphysical. On the contrary, the velocity field is always tangential to the surface if we use the velocity correction.

We also define the discrete kinetic energy on the surface as follows

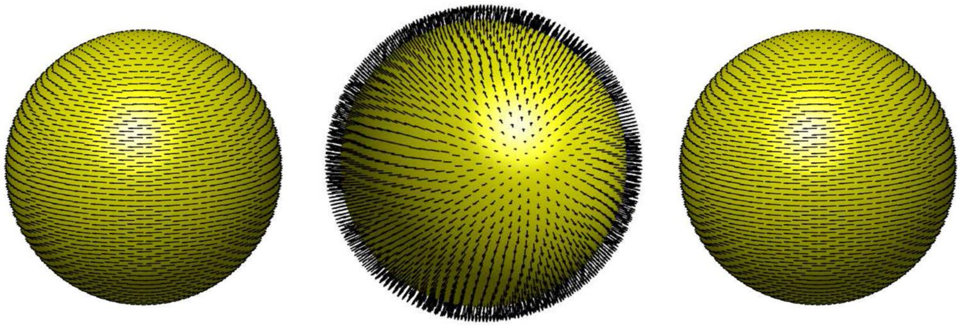
$$\mathcal{E}_d^n = \sum_{p=1}^{M} \frac{1}{6} |\mathbf{u}(\mathbf{x}_p, t^n)|^2 \mathcal{A}(\mathbf{x}_p), \tag{14}$$

where the superscript n represents the solution at nth time level, i.e., $t^n = n\Delta t$, M is the numbers of vertex on the surface. $\mathcal{A}(\mathbf{x}_p) = \sum_{q \in M_1(p)} A_q$, where A_q is the area of T_q, $M_1(p) = p_1, p_2, \ldots, p_n$ is the set of vertex indices of one-ring neighbors of $\mathbf{x}_p$ with $p_1 = p_n$ [see Figs. 4(a) and (b)]. For a system without input of external force, the basic physical property is the energy dissipation with time evolution. In Figs. 4(c) and (d), we plot the time evolutions of normalized kinetic energy $\mathcal{E}_d^n / \mathcal{E}_d^0$ with respect to the cases without and with velocity correction, respectively. It can be observed that the case with velocity correction indeed follows the energy dissipation law. However, the case without velocity correction leads to the increase of kinetic energy, which is a nonphysical phenomenon. The results in this section indicate that our proposed velocity correction technique plays an important role in the simulation of fluid flow on surface.

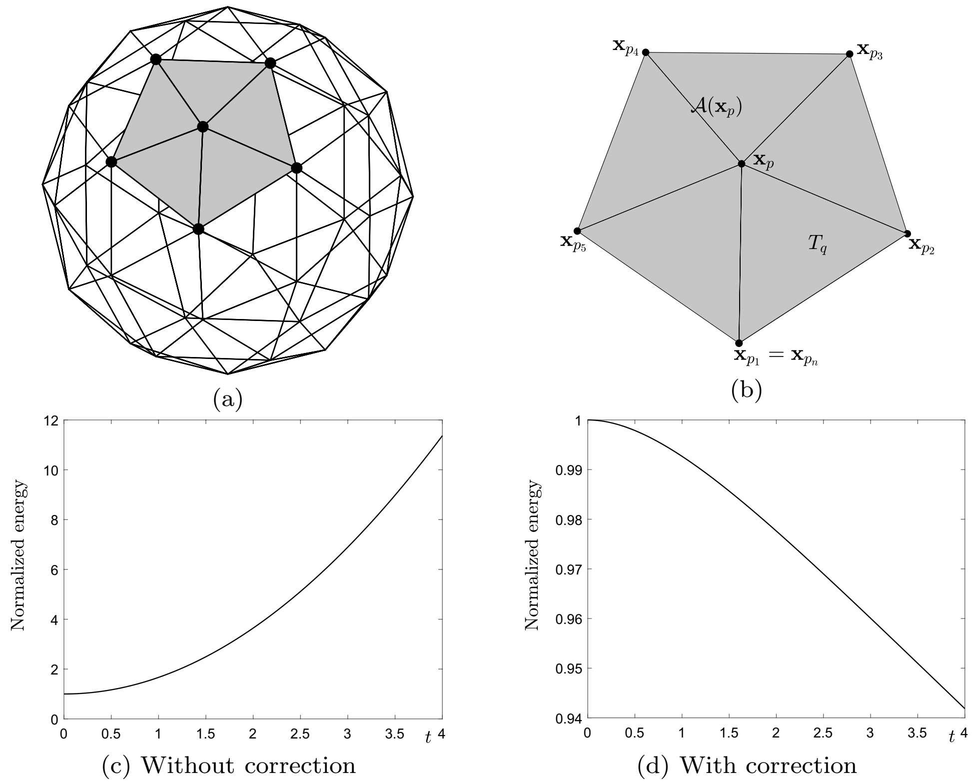
3.2 Effect of mesh size on energy dissipation

Here, we investigate the effect of mesh size by considering the dissipation of kinetic energy. The simulations are performed on a sphere with radius $R = 10$. The distance function and other parameters keep unchanged as those in Sect. 3.1. Three different average mesh sizes $h_{ave}^1 = 0.4927$, $h_{ave}^2 = 0.2504$, $h_{ave}^3 = 0.1257$ are used (see Fig. 5). The evolutions of normalized kinetic energy are shown in Fig. 5(d). It can be observed that the energy curve converges with the refinement of mesh size.

Fig. 3 Initial velocity field (**a**) and the results at $t = 4$ with respect to the cases without (**b**) and with (**c**) projection



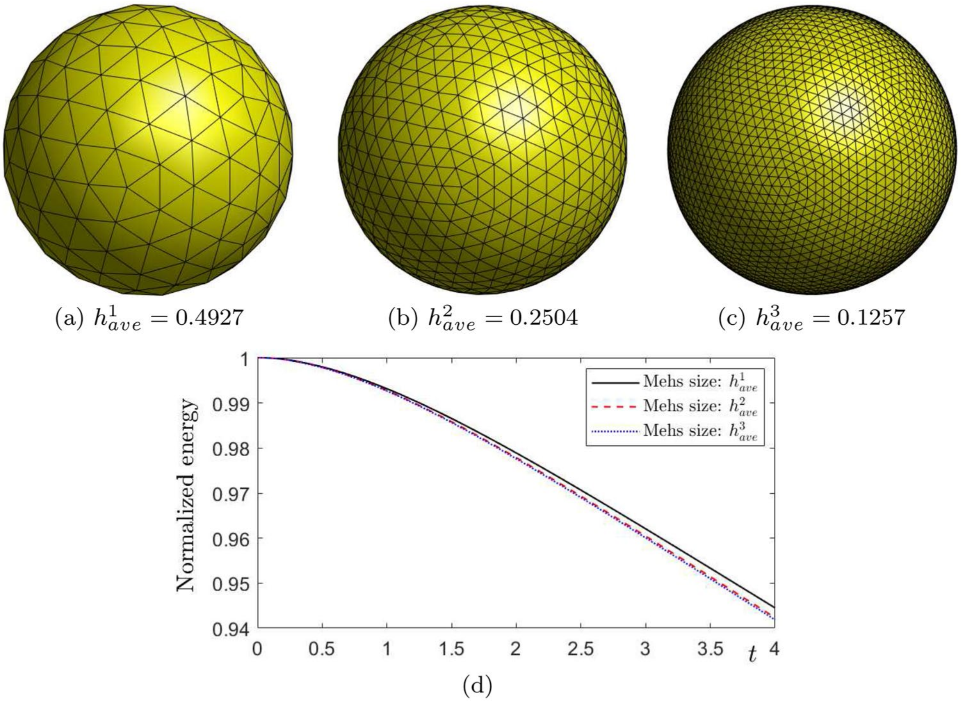
(a) Initial state (b) Without correction (c) With correction



(a) (b)

(c) Without correction (d) With correction

Fig. 4 Schematic illustrations of **a** triangular surface and **b** vertices neighbors $\mathbf{x}_p$, area $\mathcal{A}(\mathbf{x}_p)$, and triangle T_q. The evolutions of normalized kinetic energy with respect to the cases without (**c**) and with (**d**) correction



(a) $h_{ave}^1 = 0.4927$ (b) $h_{ave}^2 = 0.2504$ (c) $h_{ave}^3 = 0.1257$

(d)

Fig. 5 **a**, **b**, and **c** are spherical surfaces with difference average mesh sizes. **d** is the evolutions of normalized kinetic energy with respect to different mesh sizes

3.3 Convergence tests in time and space

To quantitatively validate the proposed method, we investigate the discrete L^2-errors and the corresponding convergence rates with respect to space and time. For the definition of discrete version of L^2 norm on a surface with triangular mesh, please refer to [31–33] and references therein. In this subsection, a spherical surface with radius $R = 1$ is considered. The initial velocity condition is defined to be $\mathbf{u}(\mathbf{x},0)=(u,v,w)=(y,-x,0)$. The other parameter values are the same as in subsection 3.1. To investigate the convergence rates in time, we discretized the surface with average mesh size $h_{ave}=0.1257$. A finer time step $\Delta t^e=0.001$ is used to obtain the numerical reference solution. A set of increasingly coarser time steps $\Delta t=4\Delta t^e$, $8\Delta t^e$, and $16\Delta t^e$ are used to perform the computations until $t=0.4$. Table 1 lists the L^2-errors and convergence rates with respect to three velocity components under different time steps. It can be observed that the proposed method has first-order accuracy in time. To investigate the convergence rates in space, we fix $\Delta t=0.001$ and coarsen the grid by a factor of two. The reference solution is calculated by using $h_{ave}=0.0629$. From the results listed in Table 2, we can confirm that the proposed method achieves second-order accuracy in space.

3.4 Comparison with previous simulations

In [8], the authors proposed a practical FDM for the incompressible fluid flows on various curved surfaces. Based on the closest point type method, the standard Laplacian operator is used to replace a vector Laplacian operator, e.g., a Bochner or Laplace–de Rham Laplacian operator [34] and the computation of NS equation is extended into a 3D narrow band domain containing the surface. Later, this method was extended for solving two-phase fluid flows on arbitrarily curved surfaces [28, 35]. In this subsection, we compare the present and previous method [8] by considering the circumfluence on a sphere. The spherical radius $R=10$, initial velocity field $\mathbf{u}(\mathbf{x},0)=(0.01y,-0.01x,0)$, time step $\Delta t=0.01$, relaxation time $\tau=1$, and Reynolds number $Re=1$ are used. Figures 6(a) and (b) display the numerical results obtained using the previous [8] and present methods, respectively, in the same view point. Note that the velocity field in (a) is obtained by interpolating the velocities in a three-dimensional narrow band embedding the surface. The velocity field in (b) is directly defined on surface points. We can observe that the present and previous simulations are qualitatively similar.

Table 1 Discrete L^2-errors and convergence rates for three velocity components under different time steps

Time step:	$16\Delta t^e$		$8\Delta t^e$		$4\Delta t^e$
Error for u:	9.81e-2		4.59e-2		1.98e-2
Rate for u:		1.09		1.21	
Error for v:	9.96e-2		4.64e-2		1.99e-2
Rate for v:		1.10		1.22	
Error for w:	8.16e-2		3.81e-2		1.64e-2
Rate for w:		1.10		1.22	

The reference time step is $\Delta t^e=0.001$

Table 2 Discrete L^2-errors and convergence rates for three velocity components under different mesh sizes

Mesh size:	$8h_{ave}$		$4h_{ave}$		$2h_{ave}$
Error for u:	3.70e-1		1.00e-1		2.52e-2
Rate for u:		1.89		1.99	
Error for v:	3.22e-1		9.20e-2		2.51e-2
Rate for v:		1.81		1.87	
Error for w:	2.32e-1		6.57e-2		1.95e-2
Rate for w:		1.82		1.75	

The reference mesh size is $h_{ave}=0.0629$

For the Stokes flows on 3D curved surfaces, Brandner et al. [34] proposed finite element methods based on the parametric TraceFEM and parametric SurfaceFEM approaches. The results indicated that their proposed methods not only achieved desired accuracy but also worked well on spherical and biconcave surfaces. We herein simulate the velocity field on the same biconcave surface adopted in [34] by using the proposed LBM method. The surface is represented by the zero level-set of $d(\mathbf{x})=(m^2+x^2+y^2+z^2)^3-8m^2(y^2+z^2)-n^4$, where $m=0.96$ and $n=0.95$. The initial velocity is $\mathbf{u}(\mathbf{x},0)=\mathbf{n}^v\times\nabla\psi(\mathbf{x})$ and $\psi(\mathbf{x})=x^2y-5z^3$. To approximate the Stokes flow, a smaller relaxation time $\tau=0.001$ is used. We use a smaller time step $\Delta t=0.0001$ to perform the relatively short-time simulation. Figure 7(a) displays the spatial discretization of a biconcave surface, (b) and (c) show the previous result adapted from [34] and the present result, respectively. In our simulation, the black arrows represent the velocity directions and the background color represents the magnitude of velocity field. It can be observed that the present and previous results are qualitatively similar.

In [1], the authors studied the evolution of a specific current on a torus by using the surface FEM method. By adopting the closest point type FDM, Yang et al. [8] performed the following benchmark simulation. A torus is defined by the zero level-set of $d(\mathbf{x})=\sqrt{(\sqrt{x^2+y^2}-2)^2+z^2}-0.5$. The

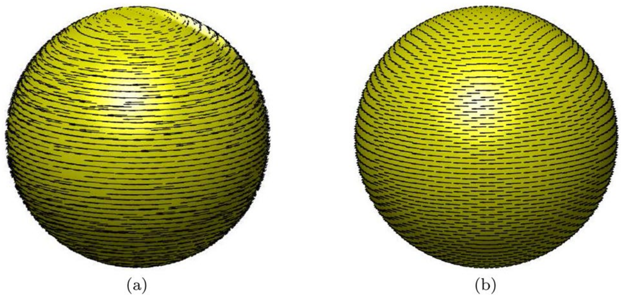
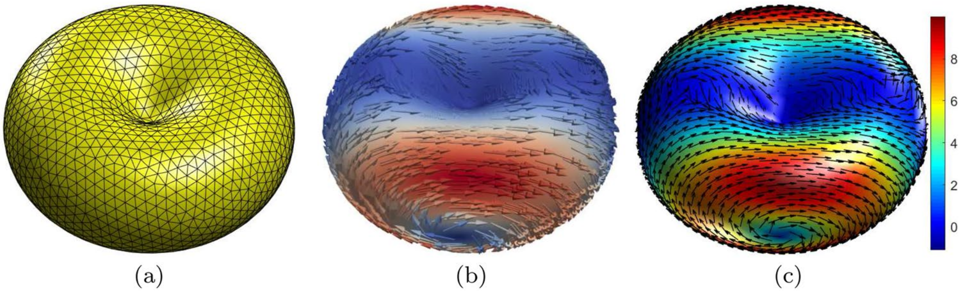


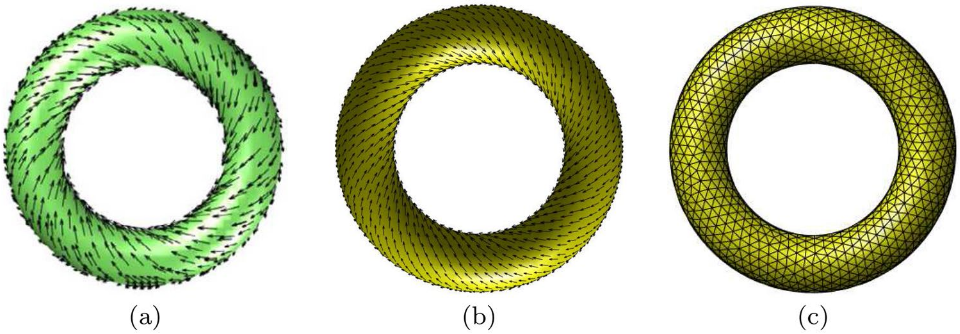
Fig. 6 Circumfluence on a sphere: **a** and **b** display the computational results obtained using the previous [8] and present methods, respectively

Fig. 7 Velocity field on a biconcave surface. Here, **a** shows the spatial discretization, **b** displays the previous result adapted from [34], **c** is the present result



(a) (b) (c)

Fig. 8 Velocity field on a torus. Here, **a** is adapted from [8] with the permission of Elsevier Science. The present result and mesh generation are shown in (**b**) and (**c**), respectively



(a) (b) (c)

Table 3 Comparison with the previous results [1, 8]

	Results in [1]	Results in [8]	Present results
$\|\mathbf{u}\|_{\min}$	0.02	0.0203	0.0196
$\|\mathbf{u}\|_{\max}$	0.12	0.1232	0.1179

Here, $|\mathbf{u}|_{\min}$ and $|\mathbf{u}|_{\max}$ are the minimum and maximum magnitudes of the velocity field, respectively

parameters are $\Delta t = 0.01$, $\tau = 1000$, and $h_{ave} = 0.2$. The velocity field is defined to be

$$\mathbf{u}(\mathbf{x}, 0) = \left(\frac{-y - 2xz}{8(x^2 + y^2)}, \ \frac{-x - 2yz}{8(x^2 + y^2)}, \ \frac{\sqrt{x^2 + y^2} - 2}{4\sqrt{x^2 + y^2}} \right). \tag{15}$$

Figure 8(a) displays the velocity distribution on a torus in [8]. The present numerical result and discrete torus are shown in Fig. 8(b) and (c), respectively. Table 3 lists the maximum and minimum values of velocity field on the surface computed by three different methods. The results are quantitatively consistent.

3.5 Kelvin–Helmholtz instability on a sphere

The Kelvin–Helmholtz instability (KHI) is a common fluid instability which appears due to the velocity difference [36]. The initial velocity difference will grow to form vortex structure. During this process, the evolution of fluid vortex will dissipate the total kinetic energy. In this subsection, we consider the KHI which consists of one layer of vortex and two layers of vortex on a sphere with radius $R = 10$. Let us define the following two marker functions

$$\phi_1(\mathbf{x}) = \tanh\left(\frac{-R\theta + 5\pi + \cos(10\psi)}{0.35} \right), \tag{16}$$

$$\phi_2(\mathbf{x}) = \begin{cases} \tanh\left(\frac{R\theta - 12.5\pi/3 - \cos(10\psi)}{0.35} \right) & \text{if} \quad z > 0, \\ \tanh\left(\frac{-R\theta + 17.5\pi/3 + \cos(10\psi)}{0.35} \right) & \text{otherwise}, \end{cases} \tag{17}$$

where $\theta = \cos^{-1}(z/r)$, $\psi = \tanh^{-1}(y, x)$, $r = \sqrt{x^2 + y^2 + z^2}$. The initial velocity field with respect to one layer and two layers of vortex are defined to be

$$\mathbf{u}_1(\mathbf{x}, 0) = (0.01y\phi_1, \ -0.01x\phi_1, \ 0), \tag{18}$$

$$\mathbf{u}_2(\mathbf{x}, 0) = (0.01y\phi_2, \ -0.01x\phi_2, \ 0). \tag{19}$$

We use $h_{ave} = 0.1257$, $\Delta t = 0.01$, and $\tau = 1$ to conduct simulations until $t = 4$. The results with respect to one layer and two layers of vortex are shown in Fig. 9(a) and (b), respectively. It can be observed that the fluid vortex first grows with time evolution and then dissipates due to the viscous resistance of fluid. The normalized energy curves plotted in Fig. 9(c) indicate both processes dissipate the kinetic energy. Moreover, two layers of vortex quickly dissipate the kinetic energy.

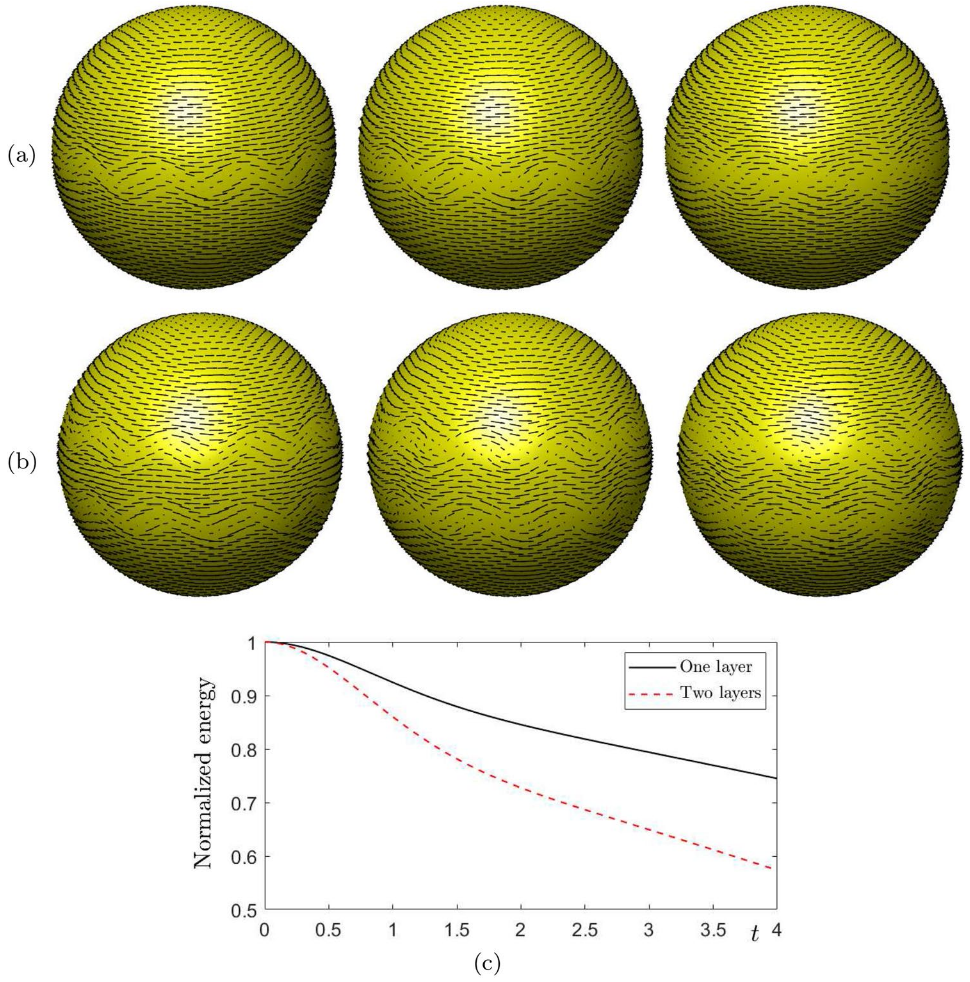


Fig. 9 Snapshots of KHI on a sphere at $t = 1.5,\ 2.5,\ 4$ with respect to one layer (**a**) and two layers (**b**) of vortex (from the left to right in each row). **c** is the evolutions of normalized kinetic energy of KHI

3.6 Flow on torus surfaces

Next, we consider the fluid flow on a torus surface. The surface is defined by the zero level-set of $d(\mathbf{x}) = \sqrt{(\sqrt{x^2 + y^2} - R_1)^2 + z^2} - R_2$, where $R_1 = 30$ is fixed and we consider $R_2 = 12$ and 18 to reflect different geometry sizes. The initial velocity field is defined to be

$$\mathbf{u}(\mathbf{x}, 0) = (-0.1n_z,\ n_z,\ 0.1n_x - n_y), \tag{20}$$

where n_x, n_y, and n_z are the components of $\mathbf{n}^v$. We use $\Delta t = 0.01$ and $\tau = 1$ in this simulation. Figure 10(a) and (b) display the snapshots at different moments with $R_2 = 12$ and 18, respectively. We can find that different geometry sizes obviously affect the evolution of fluid field. With the increase of surface area, the velocity field gently evolves. The results in Fig. 10(c) also show that the energy evolution with $R_2 = 18$ is gentler than the case with $R_2 = 12$.

3.7 Flow on a bunny surface

Finally, we consider the fluid flow on a complex bunny surface. Here, we use $\Delta t = 0.01$, $\tau = 2$, and the following three different initial velocity fields

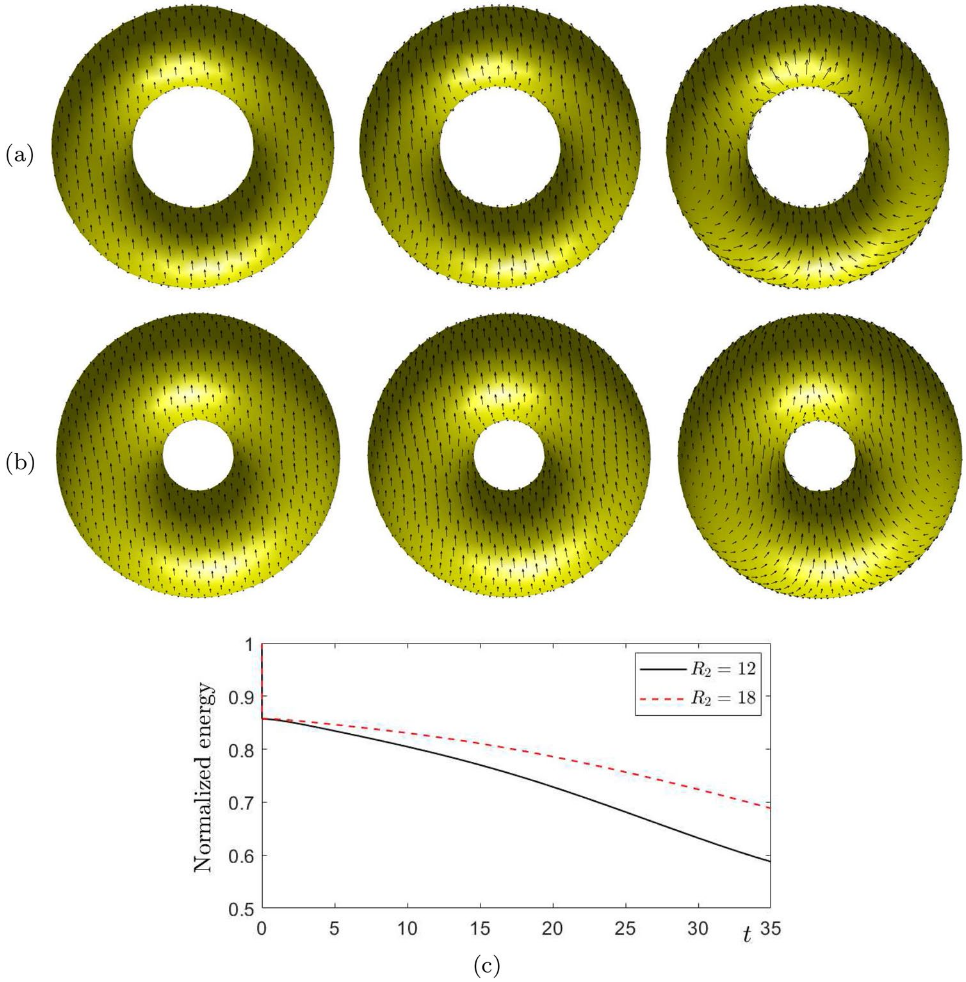
$$\mathbf{u}_1(\mathbf{x}, 0) =(-0.01n_z,\ 0.1n_z,\ 0.01n_x - 0.1n_y), \tag{21}$$

$$\mathbf{u}_2(\mathbf{x}, 0) =(0.1n_y,\ -0.1n_x + 0.01n_z,\ -0.01n_y), \tag{22}$$

$$\mathbf{u}_3(\mathbf{x}, 0) =(0.01n_y - 0.1n_z,\ -0.01n_x,\ 0.1n_x). \tag{23}$$

Figures 11(a), (b), and (c) display the computational results at $t = 8$ with the initial velocity fields Eqs. (21), (22), and (23), respectively. The energy curves with three different initial velocity fields are plotted in Fig. 11(d). It can be observed that our proposed method works well for a complex surface and the kinetic energy follows the energy dissipation property.

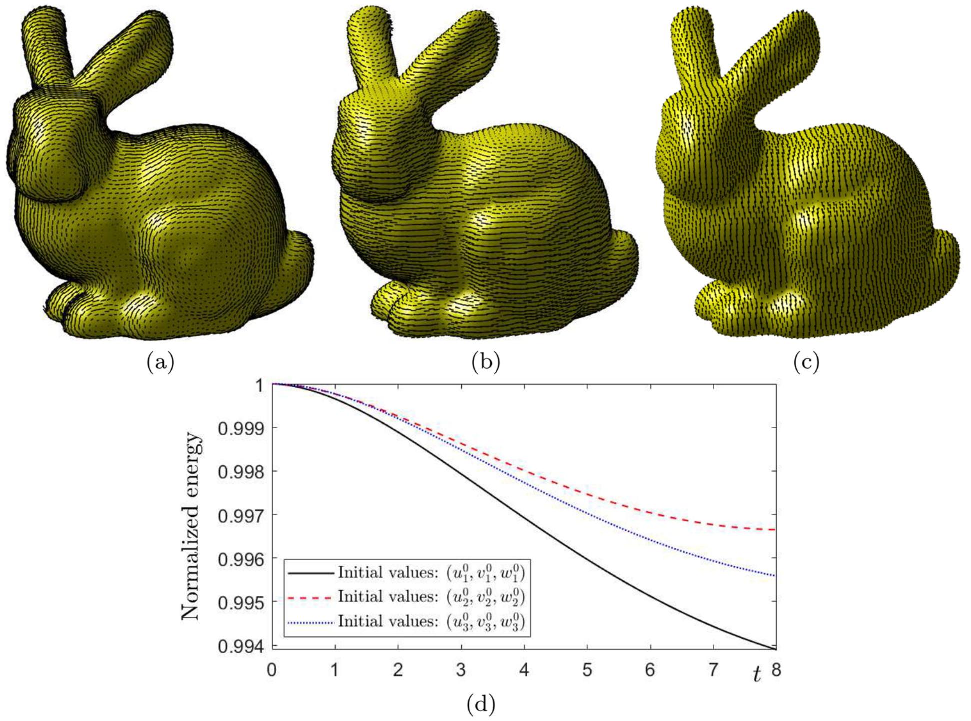
Fig. 10 Snapshots of the fluid flows on tori at $t = 0,\ 6,\ 35$ (from the left to right in each row). The results in top (**a**) and bottom (**b**) rows correspond to $R_2 = 12$ and 18, respectively. **c** is the evolutions of normalized kinetic energy on torus surfaces



4 Conclusions

In this work, we proposed an efficient and practical computational scheme for simulating fluid flows on curved surfaces in 3D space. Instead of computing the NS equations, the finite volume LBM was adopted. In one time iteration, we only needed to directly update the distribution functions on each vertex in a fully explicit form. Therefore, the whole algorithm was highly efficient. A velocity correction technique was used to keep the updated velocity field tangential to the surfaces. Furthermore, the proposed method does not need to flatten the 1-ring neighborhoods, which was done in [27]. To validate our proposed method, we performed the simulations on sphere, torus, and bunny. The numerical results showed that our method indeed worked well for fluid flows on various surfaces with complex shape. In an upcoming work, we will extend our method for predator-prey model [37], Ginzburg–Landau equation [38], phase-field models [39–41], space-fractional parabolic model [42], and multi-component fluid flows [43–47] on arbitrarily curved surfaces.

Fig. 11 **a**, **b**, and **c** are the three different velocity fields on a bunny surface at $t = 8$. **d** is the evolutions of normalized kinetic energy with three different initial velocity fields on a bunny surface



(a) (b) (c)

(d)

Acknowledgements The work of Z. Tan is supported by the National Nature Science Foundation of China (11971502), and Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University (2020B1212060032). S.K. Kim was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1C1C2005275). The corresponding author (J.S. Kim) was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A2C1003844). The authors would like to thank the reviewers for their valuable suggestions and comments to improve the paper.

References

1. Reuther S, Voigt A (2018) Solving the incompressible surface Navier-Stokes equation by surface finite elements. Phys Fluids 30:012107
2. Mohamed MS, Hirani AN, Samtaney R (2016) Discrete exterior calculus discretization of incompressible Navier-Stokes equations over surface simplicial meshes. J Comput Phys 312(1):175–191
3. Nitschke I, Reuther S, Voigt A (2017) Discrete exterior calculus (DEC) for the surface Navier-Stokes equation. Transport processes at fluidic interfaces. Birkhäuser, Cham, pp 177–197
4. Nitschke I, Voigt A, Wensch J (2012) A finite element approach to incompressible two-phase flow on manifolds. J Fluid Mech 708:418–438
5. Reuther S, Voigt A (2015) The interplay of curvature and vortices in flow on curved surfaces. Multiscale Model Simul 13(2):632–643
6. Gross BJ, Atzberger PJ (2018) Hydrodynamic flows on curved surfaces: Spectral numerical methods for radial manifold shapes. J Comput Phys 371:663–689
7. Li J, Gao Z, Dai Z, Feng X (2020) Divergence-free radial kernel for surface Stokes equations based on the surface Helmholtz decomposition. Comput Phys Commun 256:107408
8. Yang J, Li Y, Kim J (2020) A practical finite difference scheme for the Navier-Stokes equation on curved surfaces in $\mathbb{R}^3$. J Comput Phys 411:109403
9. Torres-Sánchez A, Millán D, Arroyo M (2019) Modelling fluid deformable surfaces with an emphasis on biological interfaces. J Fluid Mech 872:218–271
10. Nitschke I, Reuther S, Voigt A (2019) Hydrodynamic interactions in polar liquid crystals on evolving surfaces. Phys Rev Fluids 4(4):044002
11. Reuther S, Nitschke I, Voigt A (2020) A numerical approach for fluid deformable surfaces. J Fluid Mech 900:R8
12. Hejranfar K, Ezzatneshan E (2014) Implementation of a high-order compact finite-difference lattice Boltzmann method in generalized curvilinear coordinates. J Comput Phys 267:28–49
13. Matyka M, Dzikowski M (2021) Memory-efficient Lattice Boltzmann Method for low Reynolds number flows. Comput Phys Commun 267:108044
14. Adam S, Hajabdollahi F, Premnath KN (2021) Cascaded lattice Boltzmann modeling and simulations of three-dimensional non-Newtonian fluid flows. Comput Phys Commun 262:107858
15. Tang X, Su Y, Wang F, Li L (2013) Numerical research on lid-driven cavity flows using a three-dimensional lattice Boltzmann model on non-uniform meshes. Sci China Technol Sci 56(9):2178–2187
16. Liu H, Valocchi AJ, Kang Q (2012) Three-dimensional lattice Boltzmann model for immiscible two-phase flow simulations. Phys Rev E 85(4):046309
17. Sadeghi R, Shadloo MS, Hopp-Hirschler M, Hadjadj A, Nieken U (2018) Three-dimensional lattice Boltzmann simulations of high density ratio two-phase flows in porous media. Comput Math Appl 75(7):2445–2465

18. Li D, Tong ZX, Ren Q, He YL, Tao WQ (2017) Three-dimensional lattice Boltzmann models for solid-liquid phase change. Int J Heat Mass Transf 115:1334–1347
19. Peng G, Xi H, Duncan C, Chou SH (1999) Finite volume scheme for the lattice Boltzmann method on unstructured meshes. Phys Rev E 59(4):4675
20. Patil DV, Lakshmisha KN (2009) Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh. J Comput Phys 228(14):5262–5279
21. Li W, Luo LS (2016) Finite volume lattice Boltzmann method for nearly incompressible flows on arbitrary unstructured meshes. Commun Comput Phys 20(2):301–324
22. Wang Y, Zhong C, Cao J, Zhuo C, Liu S (2020) A simplified finite volume lattice Boltzmann method for simulations of fluid flows from laminar to turbulent regime, part I: numerical framework and its application to laminar flow simulation. Comput Math Appl 79(5):1590–1618
23. Wang Y, Zhong C, Cao J, Zhuo C, Liu S (2020) A simplified finite volume lattice Boltzmann method for simulations of fluid flows from laminar to turbulent regime, part II: extension towards turbulent flow simulation. Comput Math Appl 79(8):2133–2152
24. Ma J, Wang Z, Young J, Lai JCS, Sui Y, Tian FB (2020) An immersed boundary-lattice Boltzmann method for fluid-structure interaction problems involving viscoelastic fluids and complex geometries. J Comput Phys 415:109487
25. Yu Y, Li Q, Wen ZX, Huang RZ (2020) Investigation on boundary schemes in lattice Boltzmann simulations of boiling heat transfer involving curved surfaces. Phys Fluids 32(6):063305
26. Di Ilio G, Ubertini S, Succi S, Falcucci G (2020) Nanofluid heat transfer in wavy-wall channels with different geometries: a finite-volume lattice Boltzmann study. J Sci Comput 83(3):56
27. Fan Z, Zhao Y, Kaufman A, He Y (2005) Adapted unstructured LBM for flow simulation on curved surfaces. In Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp 245–254
28. Yang J, Kim J (2020) A phase-field model and its efficient numerical method for two-phase flows on arbitrarily curved surfaces in 3D space. Comput Methods Appl Mech Engrg 372:113382
29. Chen SG, Wu JY (2004) Estimating normal vectors and curvatures by centroid weights. Comput Aided Geom Des 21:447–458
30. Junk M, Klar A (2000) Discretizations for the incompressible Navier-Stokes equations based on the lattice Boltzmann method. SIAM J Sci Comput 22(1):1–19
31. Li Y, Kim J, Wang N (2017) An unconditionally energy-stable second-order time-accurate scheme for the Cahn-Hilliard equation on surfaces. Commun Nonlinear Sci Numer Simulat 53:213–227
32. Li Y, Qi X, Kim J (2018) Direct discretization method for the Cahn-Hilliard equation on an evolving surface. J Sci Comput 77:1147–1163
33. Li Y, Luo C, Xia B, Kim J (2019) An efficient linear second order unconditionally stable direct discretization method for the phase-field crystal equation on surfaces. Appl Math Model 67:477–490
34. Brandner P, Jankuhn T, Praetorius S, Reusken A, Voigt A (2021) Finite element discretization methods for velocity-pressure and stream function formulations of surface Stokes equations. arXiv preprint arXiv:2103.03843
35. Yang J, Kim J (2021) Numerical study of incompressible binary fluids on 3D curved surfaces based on the conservative Allen-Cahn-Navier-Stokes model. Comput Fluids 228:105094
36. Lee HG, Kim J (2015) Two-dimensional Kelvin-Helmholtz instabilities of multi-component fluids. Eur J Mech B Fluids 49:77–88
37. Zhang M, Xiao X, Feng X (2021) Numerical simulations for the predator-prey model on surfaces with lumped mass method. Eng Comput 37:2047–2058
38. Abbaszadeh M, Dehghan M (2020) The fourth-order time-discrete scheme and split-step direct meshless finite volume method for solving cubic-quintic complex Ginzburg-Landau equations on complicated geometries. Eng Comput. https://doi.org/10.1007/s00366-020-01089-6
39. Liang H, Zhang C, Du R, Wei Y (2020) Lattice Boltzmann method for fractional Cahn-Hilliard equation. Commun Nonlinear Sci Numer Simulat 91:105443
40. Liang H, Xu J, Chen J, Chai Z, Shi B (2019) Lattice Boltzmann modeling of wall-bounded ternary fluid flows. Appl Math Model 73:487–513
41. Yang J, Li Y, Lee C, Lee HG, Kwak S, Hwang Y, Xin X, Kim J (2022) An explicit conservative Saul'yev scheme for the Cahn-Hilliard equation. Int J Mech Sci 217:106985
42. Qiao Y, Qian L, Feng X (2021) Fast numerical approximation for the space-fractional semilinear parabolic equations on surfaces. Eng Comput. https://doi.org/10.1007/s00366-021-01357-z
43. Nee A (2020) Hybrid lattice Boltzmann-Finite difference formulation for combined heat transfer problems by 3D natural convection and surface thermal radiation. Int J Mech Sci 173:105447
44. Zong Y, Zhang C, Liang H, Wang L, Xu J (2020) Modeling surfactant-laden droplet dynamics by lattice Boltzmann method. Phys Fluids 32:122105
45. Sun M, Feng X, Wang K (2020) Numerical simulation of binary fluid-surfactant phase field model coupled with geometric curvature on the curved surface. Comput Methods Appl Mech Engrg 367:113123
46. Xia Q, Yu Q, Li Y (2021) A second-order accurate, unconditionally energy stable numerical scheme for binary fluid flows on arbitrarily curved surfaces. Comput Methods Appl Mech Engrg 384:113987
47. Esfe MH, Esfandeh S, Bahiraei M (2020) A two-phase simulation for investigating natural convection characteristics of nanofluid inside a perturbed enclosure filled with porous medium. Eng Comput. https://doi.org/10.1007/s00366-020-01204-7

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.