

# A practical finite difference scheme for the Navier–Stokes equation on curved surfaces in $\mathbb{R}^3$

Junxiang Yang<sup>a</sup>, Yibao Li<sup>b</sup>, Junseok Kim<sup>a,\*</sup>

<sup>a</sup> Department of Mathematics, Korea University, Seoul, 02841, Republic of Korea

<sup>b</sup> School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China



## ARTICLE INFO

### Article history:

Received 10 October 2019

Received in revised form 8 March 2020

Accepted 9 March 2020

Available online 19 March 2020

### Keywords:

Incompressible Navier–Stokes equation

Curved surfaces

Narrow band domain

Closest-point method

Projection method

## ABSTRACT

We present a practical finite difference scheme for the incompressible Navier–Stokes equation on curved surfaces in three-dimensional space. In the proposed method, the curved surface is embedded in a narrow band domain and the governing equation is extended to the narrow band domain. We use the standard seven-point stencil for the Laplace operator instead of a discrete Laplacian–Beltrami operator by using the closet point method and pseudo-Neumann boundary condition. The well-known projection method is used to solve the incompressible Navier–Stokes equation in the narrow band domain. To make the velocity field be parallel to the surface, a velocity correction step is used. Various numerical experiments, such as the divergence-free test, the convergence rate, and the energy dissipation, are performed on curved surfaces, which demonstrated that our proposed method is robust and practical.

© 2020 Elsevier Inc. All rights reserved.

## 1. Introduction

The incompressible Navier–Stokes (NS) equation has wide applications in various problems with fluid flow. Because the analytic solution for the NS equation is generally hard to find, it is necessary to develop efficient and accurate numerical schemes for solving the NS equation numerically. The NS equation couples the velocity and pressure fields, which causes difficulty in numerical solution. To overcome this difficulty, the projection method [1–3], which decouples the solution of the momentum equation from the solution of the continuity equation, has been widely used. Later, many researchers have developed robust and accurate numerical schemes for solving the NS equation. For example, Zhang et al. [4] performed a direct discontinuous Galerkin (DDG) method for solving incompressible NS equation and their method has the merit of high-order accuracy in arbitrary grids. Lin et al. [5] developed an efficient and practical numerical scheme based on an auxiliary energy variable for the NS equation. Suchde et al. [6] proposed a new meshfree solver for the NS equation which achieves the better mass conservation and accuracy than the previous meshfree methods. Recently, various numerical schemes based on the finite element method have been developed for the NS equation [7–10]. Although many numerical schemes work well for the incompressible NS equation on two-dimensional planes, most of them are limited to the flat surfaces.

Solving the partial differential equations on curved surfaces in  $\mathbb{R}^3$  is an interesting problem. Recently, many efficient and robust numerical methods have been proposed for solving the Cahn–Hilliard (CH) [11–14], Allen–Cahn (AC) [15], phase-field crystal (PFC) [16,17], and diblock copolymer (DC) equations [18] on curved surfaces. However, the numerical simulations of

\* Corresponding author.

E-mail address: cfdkim@korea.ac.kr (J. Kim).

URL: <http://math.korea.ac.kr/~cfdkim> (J. Kim).

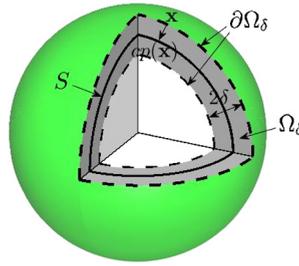


Fig. 1. Schematic representations of  $S$ ,  $\Omega_\delta$ ,  $\partial\Omega_\delta$ , and  $\text{cp}(\mathbf{x})$ .

NS equation on curved surfaces in  $\mathbb{R}^3$  are not very rare. At present, two efficient methods such as discrete exterior calculus method and surface finite element method for solving the NS equation on curved surfaces in  $\mathbb{R}^3$  have been developed. Mohamed et al. [19] developed a mass conservative discretization for the incompressible NS equation on surfaces, which is based on the discrete exterior calculus (DEC) method. Their scheme has second-order accuracy for structured meshes and first-order accuracy for unstructured meshes. Later, Nitschke et al. [20] used the DEC method to solve the vorticity-stream function instead of the incompressible NS equation on various curved surfaces. Reuther and Voigt [21] solved the incompressible NS equation on surfaces based on the surface finite element and their scheme can be implemented in most finite element approaches. Different from the previous methods mentioned above, our proposed method uses the standard seven-point finite difference Laplacian operator instead of Laplacian–Beltrami operator by using the closet point method and the pseudo-Neumann boundary condition.

In the proposed method, the projection method [1] is used to solve the incompressible NS equation in a narrow band domain. To obtain the velocity field parallel to the surface, a velocity correction step is used. Compared with the previous methods mentioned above, our proposed method is very easy to implement because the standard finite difference Laplacian operator is used. The proposed method can also be applied to simulate the flow on more complex surfaces, such as a bunny. As a future work, the proposed method will be extended to solve the two-phase flow system on various curved surfaces in a phase-field approach. Recently, the phase-field approaches have been widely applied to simulate multi-phase fluid flow problems [22–24].

The organization of this paper is as follows. In Section 2, we describe the incompressible NS equation on a narrow band domain in three-dimensional space. In Section 3, we provide the algorithm of numerical solution. We present the numerical results in Section 4. In Section 5, conclusions are drawn.

## 2. Incompressible NS equation on a narrow band domain

We consider the following incompressible NS equation on a curved surface  $S$  in three-dimensional space  $\mathbb{R}^3$ :

$$\frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} + \mathbf{u}(\mathbf{x}, t) \cdot \nabla_s \mathbf{u}(\mathbf{x}, t) = -\nabla_s p(\mathbf{x}, t) + \frac{1}{Re} \Delta_s \mathbf{u}(\mathbf{x}, t), \quad \mathbf{x} \in S, \quad 0 < t \leq T, \quad (1)$$

where  $\mathbf{u}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t), w(\mathbf{x}, t))$  and  $p(\mathbf{x}, t)$  are the velocity field and pressure at the position  $\mathbf{x}$  on surface  $S$  and time  $t$ .  $Re$  is the Reynolds number,  $\nabla_s = \mathbf{P}\nabla$  indicates the surface gradient operator, where  $\mathbf{P} = \mathbf{I} - (\nabla d)^T \nabla d$  is projection operator onto the tangent plane [16,25]. Here,  $\mathbf{I}$  is the identity matrix [25,26] and  $d: \mathbb{R}^3 \rightarrow \mathbb{R}$  is a signed distance function such that  $S = \{\mathbf{x} \in \mathbb{R}^3 \mid d(\mathbf{x}) = 0\}$  with  $d < 0$  inside and  $d > 0$  outside of  $S$ . The  $\delta$ -neighborhood band of  $S$  is defined as  $\Omega_\delta = \{\mathbf{y} \mid \mathbf{y} = \mathbf{x} + \theta \mathbf{n}(\mathbf{x}), \theta \in [-\delta, \delta], \mathbf{x} \in S\}$ , where  $\mathbf{n}(\mathbf{x})$  is a unit normal vector at  $\mathbf{x}$ . Next, we extend Eq. (1) to the narrow band domain  $\Omega_\delta$ :

$$\frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} + \mathbf{u}(\mathbf{x}, t) \cdot \nabla \mathbf{u}(\mathbf{x}, t) = -\nabla p(\mathbf{x}, t) + \frac{1}{Re} \Delta \mathbf{u}(\mathbf{x}, t), \quad \mathbf{x} \in \Omega_\delta, \quad 0 < t \leq T. \quad (2)$$

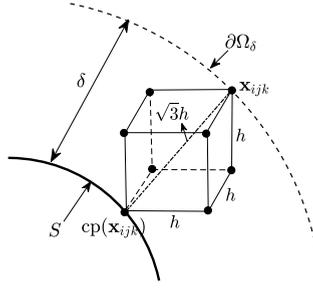
Here, the following pseudo-Neumann boundary conditions are used:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\text{cp}(\mathbf{x}), t), \quad p(\mathbf{x}, t) = p(\text{cp}(\mathbf{x}), t) \quad \text{on } \partial\Omega_\delta, \quad (3)$$

where the closest point  $\text{cp}(\mathbf{x}) \in S$  for  $\mathbf{x} \in \partial\Omega_\delta$  [27], see Fig. 1. If a small enough  $\delta$  is taken, then we have constant values of  $\mathbf{u}$  and  $p$  in the direction normal to the surface  $S$ . Therefore, the standard Laplacian operator can be used to replace the Laplace–Beltrami operator in the narrow band domain  $\Omega_\delta$  [28].

## 3. Numerical solution

We describe the numerical solution for the incompressible NS equation on  $\Omega_\delta$ . We discretize the NS equation in  $\Omega = (a, b) \times (c, d) \times (e, f)$  embedding  $\Omega_\delta$ . Let  $N_x$ ,  $N_y$ , and  $N_z$  be positive integers,  $h = (b-a)/N_x = (d-c)/N_y = (f-e)/N_z$  be the space step, and  $\Omega^h = \{\mathbf{x}_{ijk} = (x_i, y_j, z_k) = (a + hi, c + hj, e + hk) \mid 0 \leq i \leq N_x, 0 \leq j \leq N_y, 0 \leq k \leq N_z\}$  be the discrete domain.



**Fig. 2.** Schematic illustrations of boundary point  $\mathbf{x}_{ijk}$  and its closest point  $\text{cp}(\mathbf{x}_{ijk})$ . The minimum value of  $\delta$  should be greater than  $\sqrt{3}h$ .

In this work, the velocity and pressure variables are stored at cell vertices. Let  $\mathbf{u}_{ijk}^n$  and  $p_{ijk}^n$  be discrete approximations of  $\mathbf{u}(\mathbf{x}_{ijk}, n\Delta t)$  and  $p(\mathbf{x}_{ijk}, n\Delta t)$ , respectively, where  $\Delta t$  is the temporal step. Let  $S = \{\mathbf{x} \in \mathbb{R}^3 \mid d(\mathbf{x}) = 0\}$ , where  $d: \mathbb{R}^3 \rightarrow \mathbb{R}$  is the signed distance function to  $S$ . Let  $\Omega_\delta^h = \{\mathbf{x}_{ijk} \mid |d_{ijk}| < \delta\}$  be the discrete narrow band domain with  $\delta > \sqrt{3}h$ . Note that the closet points of all boundary points are calculated by using a trilinear interpolation. Therefore, the discrete narrow band domain  $\Omega_\delta^h$  must include all interpolation stencils, that means we need to take  $\delta > \sqrt{3}h$ ; otherwise, we may include the boundary point as an interpolation stencil, see Fig. 2 as a schematic illustration.

Let  $\partial\Omega_\delta^h = \{\mathbf{x}_{ijk} \mid I_{ijk} |\nabla_h I_{ijk}| \neq 0\}$  be the discrete boundary, where  $\nabla_h I_{ijk} = (I_{i+1,j,k} - I_{i-1,j,k}, I_{i,j+1,k} - I_{i,j-1,k}, I_{i,j,k+1} - I_{i,j,k-1}) / (2h)$ . Here,  $I_{ijk} = 0$  if  $\mathbf{x}_{ijk} \in \Omega_\delta^h$ ; otherwise  $I_{ijk} = 1$ . We first define the discretization of the NS equation:

$$\frac{\mathbf{u}_{ijk}^{n+1} - \mathbf{u}_{ijk}^n}{\Delta t} = -(\mathbf{u} \cdot \nabla_d \mathbf{u})_{ijk}^n - \nabla_d p_{ijk}^{n+1} + \frac{1}{Re} \Delta_d \mathbf{u}_{ijk}^n, \tag{4}$$

$$\nabla_d \cdot \mathbf{u}_{ijk}^{n+1} = 0. \tag{5}$$

We denote that  $\nabla_d$ ,  $\Delta_d$ , and  $\nabla_d \cdot$  are the discrete gradient, Laplacian, and divergence operators, respectively. The standard seven-point discrete Laplacian operator for velocity  $\mathbf{u}$  is

$$\Delta_d \mathbf{u}_{ijk} = (\mathbf{u}_{i+1,j,k} + \mathbf{u}_{i-1,j,k} + \mathbf{u}_{i,j+1,k} + \mathbf{u}_{i,j-1,k} + \mathbf{u}_{i,j,k+1} + \mathbf{u}_{i,j,k-1} - 6\mathbf{u}_{ijk}) / h^2.$$

The pseudo-Neumann boundary condition is defined as

$$\mathbf{u}_{ijk} = \mathbf{u}(\text{cp}(\mathbf{x}_{ijk})) \text{ and } p_{ijk} = p(\text{cp}(\mathbf{x}_{ijk})) \text{ for } \mathbf{x}_{ijk} \in \partial\Omega_\delta^h,$$

where

$$\text{cp}(\mathbf{x}_{ijk}) = \mathbf{x}_{ijk} - \frac{\nabla_d |d_{ijk}|}{|\nabla_d |d_{ijk}||^2} |d_{ijk}|. \tag{6}$$

Since  $\text{cp}(\mathbf{x}_{ijk})$  is generally not a grid point,  $\mathbf{u}(\text{cp}(\mathbf{x}_{ijk}))$  is obtained using the trilinear interpolation method. For each  $\mathbf{x}_{ijk} \in \partial\Omega_\delta^h$ , we compute  $\text{cp}(\mathbf{x}_{ijk})$  using Eq. (6) and find the cube cell,  $[x_p, x_{p+1}] \times [y_q, y_{q+1}] \times [z_r, z_{r+1}]$  containing the point  $\text{cp}(\mathbf{x}_{ijk})$ . Let  $(\alpha_1, \alpha_2, \alpha_3) = \text{cp}(\mathbf{x}_{ijk}) - \mathbf{x}_{pqr}$ , then

$$\begin{aligned} \mathbf{u}(\text{cp}(\mathbf{x}_{ijk})) = & [(h - \alpha_1)(h - \alpha_2)(h - \alpha_3)\mathbf{u}_{pqr} + \alpha_1(h - \alpha_2)(h - \alpha_3)\mathbf{u}_{p+1,qr} \\ & + (h - \alpha_1)\alpha_2(h - \alpha_3)\mathbf{u}_{p,q+1,r} + \alpha_1\alpha_2(h - \alpha_3)\mathbf{u}_{p+1,q+1,r} \\ & + (h - \alpha_1)(h - \alpha_2)\alpha_3\mathbf{u}_{pq,r+1} + \alpha_1(h - \alpha_2)\alpha_3\mathbf{u}_{p+1,q,r+1} \\ & + (h - \alpha_1)\alpha_2\alpha_3\mathbf{u}_{p,q+1,r+1} + \alpha_1\alpha_2\alpha_3\mathbf{u}_{p+1,q+1,r+1}] / h^3. \end{aligned}$$

The outline of the main procedure for solving the discrete equations (4) and (5) is as follows:

*Step 1.* Give the initial velocity field  $\mathbf{u}^0$  which satisfies the divergence-free condition.

*Step 2.* The intermediate velocity field  $\mathbf{u}^*$  is solved explicitly with the absence of pressure gradient term,

$$\frac{\mathbf{u}_{ijk}^* - \mathbf{u}_{ijk}^n}{\Delta t} = -\mathbf{u}_{ijk}^n \cdot \nabla_d \mathbf{u}_{ijk}^n + \frac{1}{Re} \Delta_d \mathbf{u}_{ijk}^n.$$

We use the second-order ENO scheme [29,30] for the spatial discretization of advection term in Eq. (4). The advection term is discretized as:

$$\begin{aligned}
 (\mathbf{u} \cdot \nabla_d \mathbf{u})_{ijk} &= \frac{u_{ijk}}{h} \left( \bar{\mathbf{u}}_{i+\frac{1}{2},j,k} - \bar{\mathbf{u}}_{i-\frac{1}{2},j,k} \right) + \frac{v_{ijk}}{h} \left( \bar{\mathbf{u}}_{i,j+\frac{1}{2},k} - \bar{\mathbf{u}}_{i,j-\frac{1}{2},k} \right) \\
 &\quad + \frac{w_{ijk}}{h} \left( \bar{\mathbf{u}}_{i,j,k+\frac{1}{2}} - \bar{\mathbf{u}}_{i,j,k-\frac{1}{2}} \right).
 \end{aligned} \tag{7}$$

The procedure for computing the quantity  $\bar{\mathbf{u}}_{i+\frac{1}{2},j,k}$  is as follows:

$$\begin{aligned}
 l &= \begin{cases} i & \text{if } u_{i+\frac{1}{2},j,k} > 0, \\ i+1 & \text{otherwise,} \end{cases} \\
 a &= \frac{u_{ljk} - u_{l-1,j,k}}{h}, \quad b = \frac{u_{l+1,j,k} - u_{ljk}}{h}, \quad d = \begin{cases} a & \text{if } |a| \leq |b|, \\ b & \text{otherwise,} \end{cases} \\
 \bar{\mathbf{u}}_{i+\frac{1}{2},j,k} &= u_{ljk} + \frac{h}{2} d (1 - 2(k-i)),
 \end{aligned}$$

where  $u_{i+\frac{1}{2},j,k} = (u_{ijk} + u_{i+1,j,k})/2$ . The other quantities are similarly computed.

*Step 3.* In order to solve the pressure at  $(n+1)$ th time step, we solve the following discrete equations

$$\frac{\mathbf{u}_{ijk}^{n+1,1} - \mathbf{u}_{ijk}^*}{\Delta t} = -\nabla_d p_{ijk}^{n+1}, \tag{8}$$

$$\nabla_d \cdot \mathbf{u}_{ijk}^{n+1,1} = 0. \tag{9}$$

We take the divergence operation to Eq. (8) and use the divergence-free condition, then the following pressure Poisson equation is obtained

$$\Delta_d p_{ijk}^{n+1} = \frac{1}{\Delta t} \nabla_d \cdot \mathbf{u}_{ijk}^*, \tag{10}$$

where the source term is written as

$$\nabla_d \cdot \mathbf{u}_{ijk}^* = \frac{u_{i+1,jk}^* - u_{i-1,jk}^*}{2h} + \frac{v_{i,j+1,k}^* - v_{i,j-1,k}^*}{2h} + \frac{w_{ij,k+1}^* - w_{ij,k-1}^*}{2h}$$

and  $\Delta_d p_{ijk}^{n+1}$  is discretized by the standard seven-point discrete Laplacian. We use the Jacobi iteration to solve the implicit Eq. (10). The pseudo-Neumann boundary condition is used for pressure  $p^{n+1}$  in relaxation. After each Jacobi iteration, we take the following pressure correction step to maintain the unique solution

$$p_{ijk}^{n+1,m+1} = p_{ijk}^{n+1,m} - \frac{1}{\#\Omega_\delta^h} \sum_{\mathbf{x}_{ijk} \in \Omega_\delta^h} p_{ijk}^{n+1,m+1},$$

where  $\#\Omega_\delta^h$  is the total number of points in narrow band domain. The Jacobi iteration stops when the  $L^2$ -error of the consecutive iterative solutions is less than a given tolerance, i.e.,

$$\|p^{n+1,m+1} - p^{n+1,m}\|_{L^2(\Omega_\delta^h)} < tol,$$

where  $p^{n+1,m+1}$  and  $p^{n+1,m}$  represent the iterative solutions after  $(m+1)$ th and  $m$ th rounds of Jacobi iterations.

*Step 4.* Update the velocity field at  $(n+1)$ th time step which satisfies the divergence-free condition

$$\mathbf{u}^{**} = \mathbf{u}^* - \Delta t \nabla_d p^{n+1} \tag{11}$$

*Step 5 (Velocity correction step).* When we solve the NS equation on curved surfaces, we want the velocity field is parallel to the surfaces. Although the velocity field  $\mathbf{u}^{**}$  satisfies the divergence-free condition, it may not be parallel to the curved surface and causes the nonphysical velocity distribution. To avoid this, the following velocity correction method is used: First, let  $\mathbf{m}$  be the unit normal vector to the curved surface and  $\lambda$  be a coefficient to make the vector  $\lambda \mathbf{m}$  be the component of  $\mathbf{u}^{**}$ . Then, we can easily have the relation  $(\mathbf{u}^{**} - \lambda \mathbf{m}) \cdot \mathbf{m} = 0$ , which implies  $\lambda = \mathbf{u}^{**} \cdot \mathbf{m}$ . The corrected velocity field can be obtained by

$$\mathbf{u}^{n+1} = \mathbf{u}^{**} - (\mathbf{u}^{**} \cdot \mathbf{m}) \mathbf{m}.$$

A schematic illustration of velocity correction is shown in Fig. 3. The steps 1 – 5 complete the numerical solution of the NS equation in one time cycle.

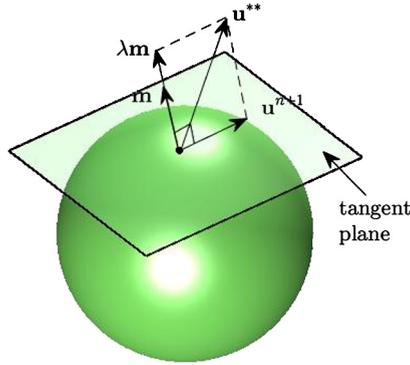


Fig. 3. Schematic illustration of velocity correction step.

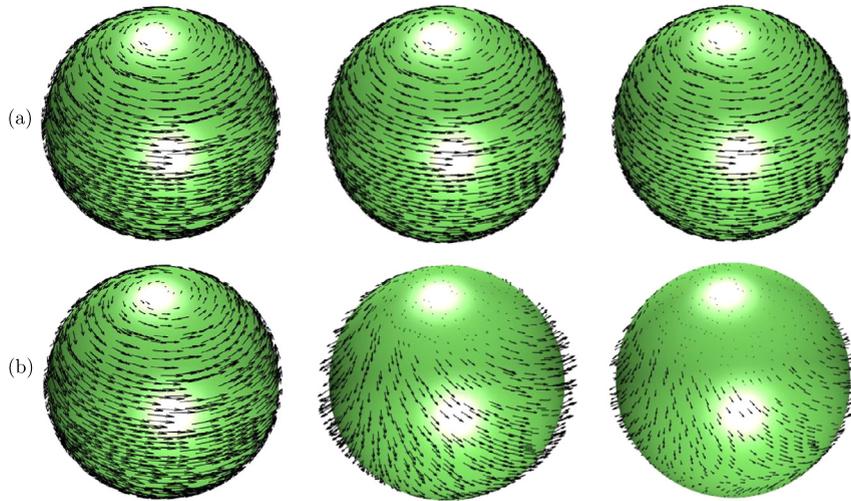


Fig. 4. Temporal evolutions of circumfluence with velocity projection (a) and without velocity projection (b). The computational times from left to right in each row are:  $t = 0, 1.92,$  and  $3.52$ .

#### 4. Numerical results

In the following parts, various numerical tests are performed to investigate the practicability and the convergence rate of the proposed method. Unless otherwise indicated,  $tol = 0.001$  is used.

##### 4.1. Divergence-free test

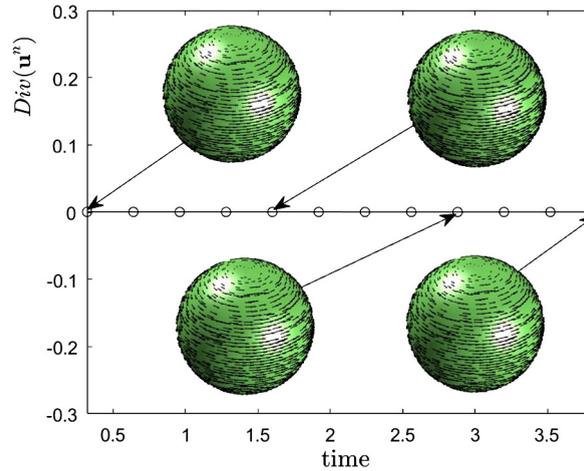
The divergence-free condition (i.e.,  $\nabla \cdot \mathbf{u} = 0$ ) is an important issue for incompressible NS equation. In order to show the accuracy of our proposed method, we consider the divergence-free condition on a curved surface. The evolution of circumfluence is investigated on a sphere with the following numerical parameters:  $h = 1/25$ ,  $\Delta t = 0.2h^2$ , and  $Re = 50$  in the computational domain  $\Omega = (0, 2)^3$ . The initial conditions are:

$$u(x, y, z, 0) = y - 1, \quad v(x, y, z, 0) = 1 - x, \quad w(x, y, z, 0) = 0. \tag{12}$$

The spherical surface is defined by the zero level-set of the distance function  $d(x, y, z) = \sqrt{(x - 1)^2 + (y - 1)^2 + (z - 1)^2} - 0.8$ . Fig. 4(a) and (b) show the temporal evolutions of velocity field on a sphere with and without velocity correction step, respectively.

If the velocity correction is absent, then the velocity field passes through the surface with time even if the initial velocity field is parallel to the surface. With the correction of velocity field, we can find that the velocity field always keeps parallel to the surface. To study the effect of the proposed velocity correction step on the divergence-free condition, we consider the temporal evolution of discrete divergence  $Div(\mathbf{u}^n)$  which is defined as:

$$Div(\mathbf{u}^n) = \frac{1}{\#\Omega_\delta^h} \sum_{\mathbf{x}_{ijk} \in \Omega_\delta^h} |\nabla_d \cdot \mathbf{u}_{ijk}^n|. \tag{13}$$



**Fig. 5.** Temporal evolution of divergence,  $Div(\mathbf{u}^n)$ . The insets represent the snapshots at the corresponding computational moments.

**Table 1**

Discrete divergence  $Div(\mathbf{u}^n)$  before and after correction step.

Case	$t = 0.64$	$t = 0.96$	$t = 1.28$	$t = 1.60$
Before Step 5	0.405	0.409	0.350	0.267
After Step 5	2.435e-4	2.364e-4	2.298e-4	2.232e-4

**Table 2**

Discrete divergence at specific moments with respect to different tolerances.

Case	$t = 0.13$	$t = 0.19$	$t = 0.64$
$tol = 1.0e-3$	7.482e-4	3.079e-4	2.492e-4
$tol = 1.0e-4$	7.480e-4	3.063e-4	2.497e-4
$tol = 1.0e-5$	7.482e-4	3.063e-4	2.498e-4

The result is shown in Fig. 5, where the insets represent the evolutions of velocity field. Note that the divergence-free condition is not rigorously satisfied with the present grid arrangement. We use an approximate projection method, where  $\Delta_d p \approx \nabla_d \cdot \nabla_d p$  is satisfied in general. For more details about the approximate projection method, we refer to [31]. To practically satisfy the divergence-free condition, an augmented correction step, i.e., Step 5 is used to enforce the velocity field in narrow band domain to be the tangential direction of the surface. In Table 1, we plot the values of discrete divergence  $Div(\mathbf{u}^n)$  before and after Step 5 at some specific temporal steps. As we can observe, the divergence-free condition is approximately satisfied by using the correction step.

#### 4.2. Test of tolerance

To verify the present tolerance ( $tol = 0.001$ ) is small enough for this work, we consider flow on a sphere surface. The initial velocity field and numerical parameters are the same as those in Section 4.1. A series of decreasing tolerances ( $tol = 1.0e-3$ ,  $1.0e-4$ , and  $1.0e-5$ ) are used. Table 2 lists the values of discrete divergence at some specific temporal steps with respect to different tolerances. Although the value of discrete divergence converges with the decrease of tolerance, the difference between them are not significant. For the purpose of saving computational time, we will use  $tol = 0.001$  in the following tests.

#### 4.3. Effect of mesh size on the discrete divergence

Next, we investigate the effect of mesh size on the values of discrete divergence. The sphere surface with radius 0.6 is considered. The initial velocity field, and numerical parameters are the same as those in Section 4.1 except the mesh size. Here, a series of increasing finer mesh sizes ( $h = 1/10$ ,  $1/20$ , and  $1/40$ ) are considered. The values of discrete divergence at  $t = 0.16$ ,  $0.4$ ,  $0.8$  with respect to different mesh sizes are listed in Table 3. We can find that the value of discrete divergence decreases with the increase of mesh size.

**Table 3**  
Discrete divergence at specific moments with respect to different mesh sizes.

Case	$t = 0.16$	$t = 0.4$	$t = 0.8$
$h = 1/10$	8.2e-3	3.2e-3	1.6e-3
$h = 1/20$	1.5e-3	4.915e-4	4.558e-4
$h = 1/40$	1.689e-4	1.635e-4	1.544e-4

**Table 4**  
 $L^2$ -errors and convergence rates at  $t = 3.5$ .

Case	$u$	$v$	$w$
$20^3 - 40^3$	0.641	0.598	0.464
Rate	2.134	2.179	2.025
$40^3 - 80^3$	0.146	0.132	0.114

4.4. Convergence test

To test the second-order accuracy in space. We consider the following initial conditions in the computational domain  $\Omega = (0, 11)^2$ :

$$\begin{aligned}
 u(x, y, z, 0) &= \frac{-\hat{y} - 2\hat{x}\hat{z}}{8(\hat{x}^2 + \hat{y}^2)}, \quad v(x, y, z, 0) = \frac{\hat{x} - 2\hat{y}\hat{z}}{8(\hat{x}^2 + \hat{y}^2)}, \\
 w(x, y, z, 0) &= \frac{\sqrt{\hat{x}^2 + \hat{y}^2} - 2}{4\sqrt{(\hat{x}^2 + \hat{y}^2)}},
 \end{aligned} \tag{14}$$

where  $\hat{x} = x - 5.5$ ,  $\hat{y} = y - 5.5$ ,  $\hat{z} = z - 5.5$ . The distance function of a torus is  $d(x, y, z) = \sqrt{(\sqrt{\hat{x}^2 + \hat{y}^2} - 2)^2} + \hat{z}^2 - 0.5$ . The following numerical parameters are used:  $h = 11/(20 \times 2^n)$ ,  $\Delta t = 0.01h$ , and  $Re = 1000$ . Here,  $n = 0, 1$ , and  $2$  are considered. We define the  $L^2$ -error between two different mesh sizes to be

$$\|\mathbf{u}^{h,\Delta t} - \mathbf{u}^{h/2,\Delta t/4}\|_{L^2(\Omega_\delta^{h/2})} = \sqrt{\frac{1}{(\#\Omega_\delta^{h/2} - N^{h/2})} \sum_{\mathbf{x}_{ijk} \in \Omega_\delta^h} |\mathbf{u}_{ijk}^{h,\Delta t} - \mathbf{u}_{2i-1,2j-1,2k-1}^{h/2,\Delta t/4}|^2}.$$

Here,  $N^{h/2}$  represents the total numbers of the even grid points in the narrow band domain with finer mesh size  $h/2$ ,  $\mathbf{u}^{h,\Delta t}$  represents the solution obtained with a space step  $h$  and a time step  $\Delta t$ . The convergence rate is defined as

$$\text{Rate} = \log_2 \left( \frac{\|\mathbf{u}^{h,\Delta t} - \mathbf{u}^{h/2,\Delta t/4}\|}{\|\mathbf{u}^{h/2,\Delta t/4} - \mathbf{u}^{h/4,\Delta t/16}\|} \right).$$

We list the convergence rates in Table 4 at  $t = 3.5$ . The results show the second-order accuracy in space.

4.5. Energy dissipation with different mesh sizes

For the incompressible NS equation, the total energy is non-increasing with time since the effect of viscosity term. In this part, we consider the kinetic energy dissipation with respect to different mesh size. The initial harmonic velocity field is defined as:

$$\begin{aligned}
 u(x, y, z, 0) &= \frac{-\hat{y} - 2\hat{x}\hat{z}}{8(\hat{x}^2 + \hat{y}^2)}, \quad v(x, y, z, 0) = \frac{\hat{x} - 2\hat{y}\hat{z}}{8(\hat{x}^2 + \hat{y}^2)}, \\
 w(x, y, z, 0) &= \frac{\sqrt{\hat{x}^2 + \hat{y}^2} - 2}{4\sqrt{(\hat{x}^2 + \hat{y}^2)}},
 \end{aligned}$$

where  $\hat{x} = x - 4$ ,  $\hat{y} = y - 4$ ,  $\hat{z} = z - 4$ . The simulation is performed on a torus in the computational domain  $\Omega = (0, 8)^3$ . The signed distance function of torus is  $d(x, y, z) = \sqrt{(\sqrt{\hat{x}^2 + \hat{y}^2} - 2)^2} + \hat{z}^2 - 0.5$ . Different mesh sizes  $h = 8/20, 8/40$ , and  $8/60$  are considered and we fix the Reynolds number  $Re = 10$  and time step  $\Delta t = 4.4e-3$  in this simulation. Figs. 6(a)–(c) show the temporal evolution of velocity field on a torus with the mesh size  $h = 8/60$ . Note that the magnitude of velocity field is properly scaled at each time step for the better views. The discrete energy is defined as

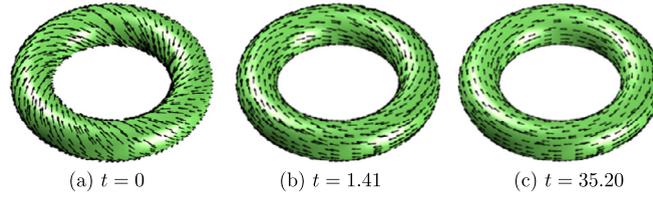


Fig. 6. Temporal evolutions of velocity field on a torus. The computational times are indicated below each figure.

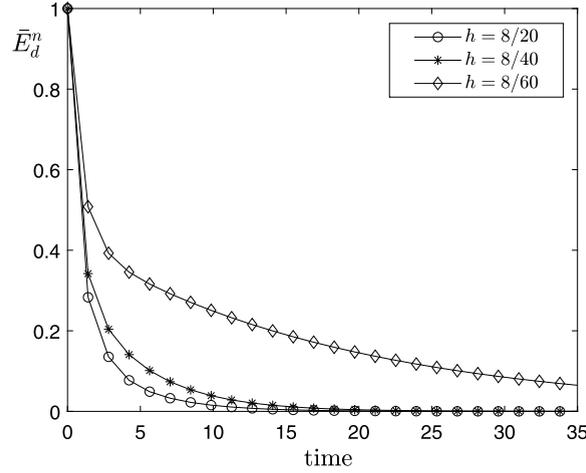


Fig. 7. Temporal evolutions of normalized discrete energy  $\bar{E}_d^n$  with respect to different mesh sizes.

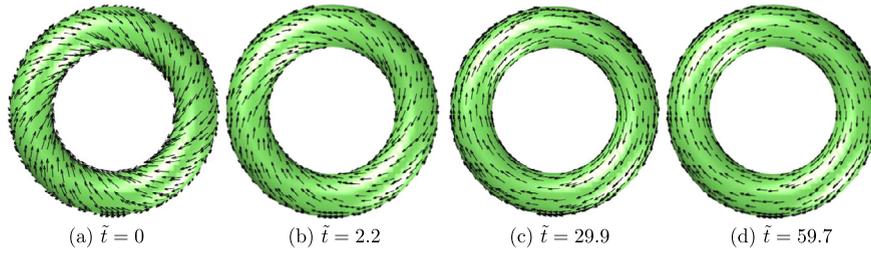


Fig. 8. Temporal evolution of velocity field on a torus.

$$E_d^n = \frac{\eta}{2\#\Omega_\delta^h} \sum_{\mathbf{x}_{ijk} \in \Omega_\delta^h} \left( u_{ijk}^2 + v_{ijk}^2 + w_{ijk}^2 \right), \quad (15)$$

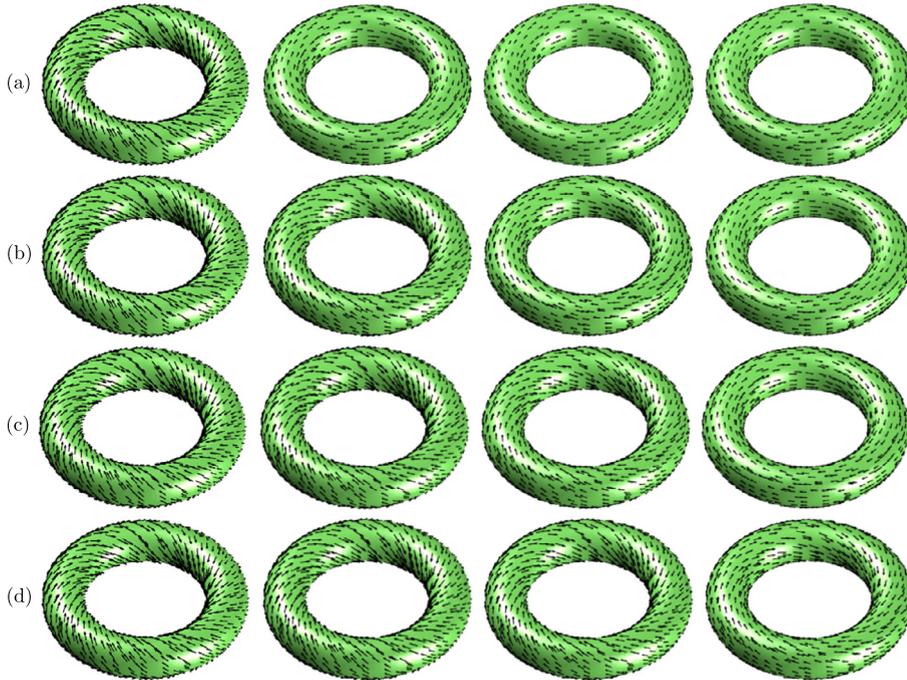
where the superscript  $n$  indicates the time steps and  $\eta$  represents the viscous coefficient which is set to be 1 in this work for convenience. The normalized discrete energy is defined as  $\bar{E}_d^n = E_d^n/E_d^0$ . Fig. 7 illustrates the temporal evolutions of normalized discrete energy  $\bar{E}_d^n$  with respect to different mesh sizes. We can find that the discrete energy is non-increasing for each mesh size.

#### 4.6. Comparison with the previous results

We consider the temporal evolution of velocity field on a torus. Note that the similar simulations can be found in [20,21] which are performed by the discrete exterior calculus and surface finite element methods, respectively. The previous simulations illustrated that the maximum and minimum values of velocity field on a torus surface are 0.12 and 0.02, respectively. In the present simulation, the torus surface, computational, initial condition, and numerical parameters are set to be same like those in Section 4.5. The temporal evolution of velocity field is shown in Fig. 8 at  $\tilde{t} = 0, 2.2, 29.9, 59.7$  in the time scale of [21] which is related to ours by  $\tilde{t} \approx 1.5t$ . Here, the velocity field is properly scaled at each time step. As we can see from Fig. 8 and Table 5, the present results and previous simulations are in good agreement.

**Table 5**  
Comparison with the reference values in [20,21]. The maximum and minimum values of velocity field are  $|\mathbf{u}|_{\max}$  and  $|\mathbf{u}|_{\min}$ , respectively.

	Reference	Simulation
$ \mathbf{u} _{\max}$	0.12	0.118
$ \mathbf{u} _{\min}$	0.02	0.02



**Fig. 9.** Temporal evolutions of velocity field on a torus with different  $Re$ : (a) 10, (b) 50, (c) 100, and (d) 200. The computational times from the left to right in each row are:  $t = 0, 1.41, 5.63,$  and  $35.20$ .

#### 4.7. Energy dissipation with respect to Reynolds number

The Reynolds number  $Re$  plays an important role in the viscosity term of the NS equation. A larger value of  $Re$  indicates the weaker effect of viscosity, while a smaller  $Re$  indicates the viscosity is dominant. In this part, we investigate the effect of  $Re$  on the energy dissipation on a curved surface. The initial conditions, computational domain, and torus surface are similarly defined as in Section 4.5. We consider different  $Re = 10, 50, 100,$  and  $200$  with a fixed mesh size  $h = 8/60$  in the simulation. Fig. 9(a)–(d) show the temporal evolutions of velocity field on a torus with respect to different  $Re$ . Note that the magnitude of velocity field is properly scaled at each time step for the better views. We can find that the velocity field quickly becomes parallel to  $x$ - $y$  plane with the decrease of  $Re$ . This phenomenon can be explained by the results in Fig. 10 which illustrates the temporal evolutions of normalized discrete energy with different  $Re$ . As  $Re$  becomes smaller, the dominant effect of viscosity makes the energy dissipate quickly with time.

#### 4.8. Flow on various surfaces

We consider the fluid field on the following three surfaces:  $S_1, S_2,$  and  $S_3$  (see Fig. 11). The simulations are performed in the domain  $\Omega = (0, 4)^3$  with parameters:  $h = 4/60, \Delta t = 0.1h^2, Re = 50$ . The level-set functions of  $S_1, S_2,$  and  $S_3$  are defined in the following Eqs. (16), (17), and (18), respectively:

$$d_1(x, y, z) = \sqrt{\frac{(x-2)^2}{0.5^2} + \frac{(y-2)^2}{0.5^2} + \frac{(z-2)^2}{1.5^2}} - 0.6, \tag{16}$$

$$d_2(x, y, z) = ((z-1.5) - f)^2 + \frac{1}{0.8^2}(y-2)^2 + (x-2)^2 - 1, \tag{17}$$

$$d_3(x, y, z) = \left((x-2)^2 - 1\right)^2 + \left((y-2)^2 - 1\right)^2 + \left((z-2)^2 - 1\right)^2 - 2$$

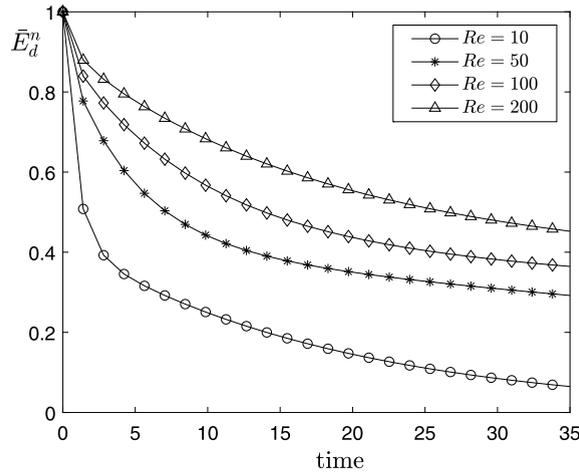


Fig. 10. Temporal evolutions of normalized discrete energy  $\bar{E}_d^n$  with respect to different  $Re$ .

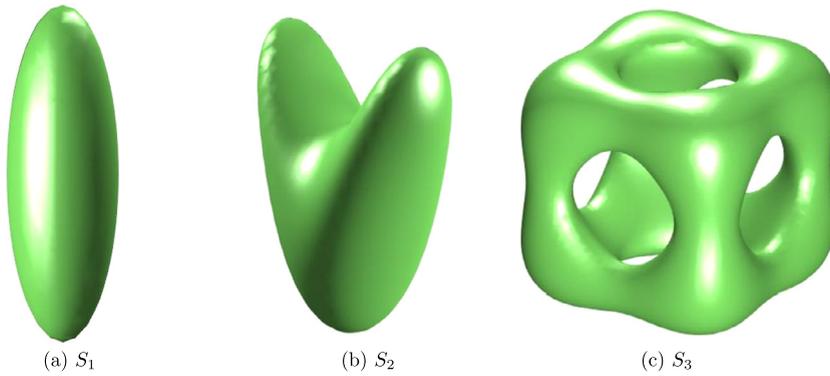


Fig. 11. Various curved surfaces: (a)  $S_1$ , (b)  $S_2$ , and (c)  $S_3$ .

$$\begin{aligned}
 &+ \left( (x-2)^2 + (y-2)^2 - 1 \right)^2 + \left( (x-2)^2 + (z-2)^2 - 1 \right)^2 \\
 &+ \left( (y-2)^2 + (z-2)^2 - 1 \right)^2, \tag{18}
 \end{aligned}$$

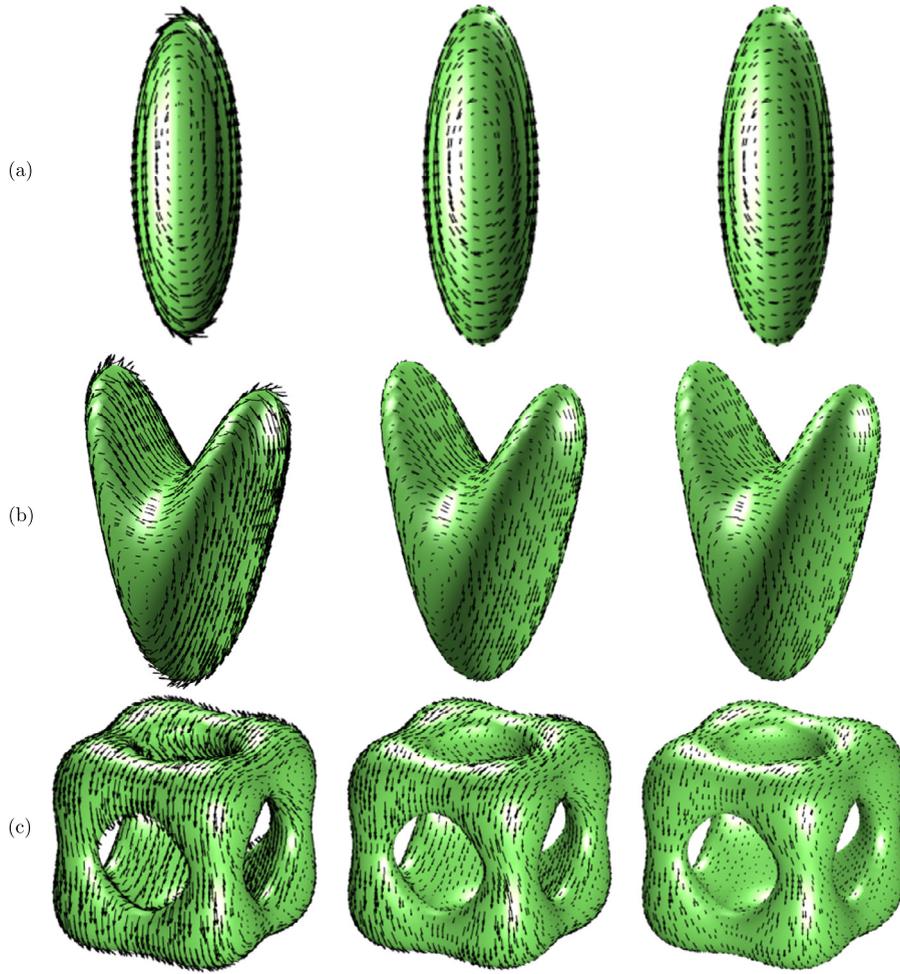
where  $f = 0.375(x-2)^2 [((x-2)+1)^2(4-3(x-2)) + 0.9((x-2)-1)^2(4+3(x-2))]$ . The initial velocity field is defined as:  $\mathbf{u}^0 = \mathbf{n} \times \nabla \psi$ , where  $\mathbf{n}$  is the unit normal vector to the surface and  $\psi = y + 0.1z$  is a scalar function. The evolutions of fluid field on various surfaces are shown in Fig. 12. The normalized discrete energy is illustrated in Fig. 13. We can observe that the proposed method can treat the fluid field on various surfaces and the energy dissipation is satisfied.

#### 4.9. Flow on a bunny surface

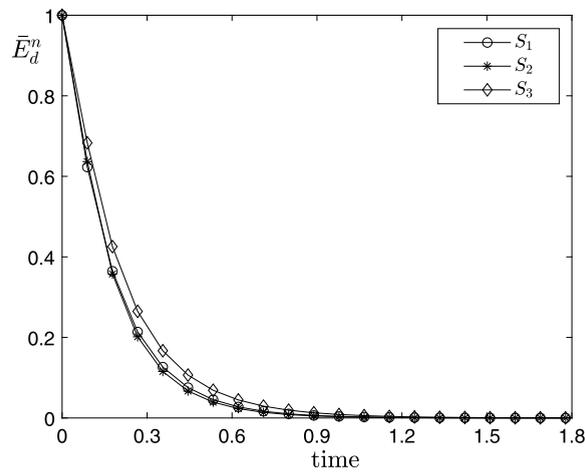
We further consider the fluid flow on a bunny surface which has a more complex shape. The computational domain is:  $\Omega = (0, 68.5)^3$  and the numerical parameters are:  $h = 0.5$ ,  $\Delta t = 0.1h^2$ ,  $Re = 50$ . The initial velocity field is define as:  $\mathbf{u}^0 = \mathbf{n} \times \nabla \psi$ , where  $\psi = x + 0.1y$  is a scalar function. The evolutions from different perspectives are shown in Fig. 14. For better views, the velocity field at each time step are properly scaled. We can find that the proposed method can also be applied to simulate the fluid flow on a bunny surface.

#### 4.10. Co-rotating vortex pair

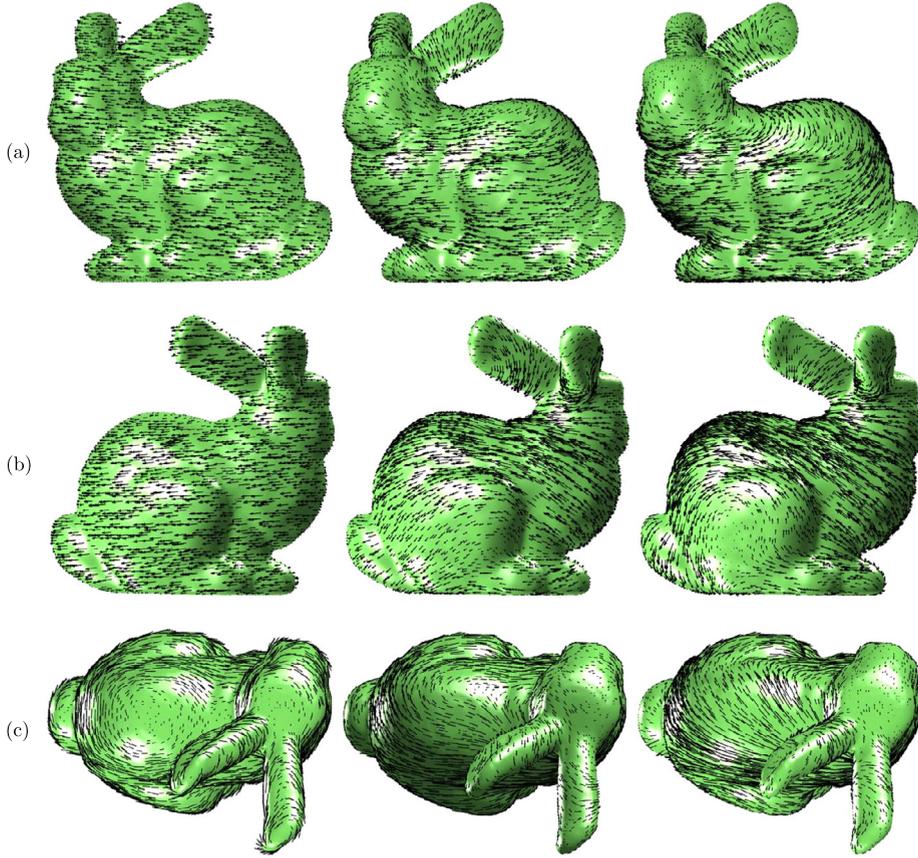
In this part, we consider the co-rotating vortex pair on a sphere which is defined by the signed distance function  $d(x, y, z) = \sqrt{(x-1)^2 + (y-1)^2 + (z-1)^2} - 0.8$  in the computational domain  $\Omega = (0, 2)^3$ . The initial conditions are given as



**Fig. 12.** Temporal evolutions of velocity field on various surfaces: (a)  $S_1$ , (b):  $S_2$ , and (c):  $S_3$ . The computational times from the left to right in each row are: 0, 0.89, 1.78. For better views, the velocity field at each time step is properly scaled.



**Fig. 13.** Temporal evolutions of normalized discrete energy  $\bar{E}_d^n$  on various surfaces.



**Fig. 14.** Temporal evolutions of velocity field on a bunny surface: (a), (b), and (c) are the evolutions from different perspectives. The computational times from the left to right in each row are: 0, 50, 100. For better views, the velocity field at each time step is properly scaled.

$$u(x, y, z, 0) = \begin{cases} 4(y-1)e^{0.3(1-l_1^2)} & \text{if } x < 1, z > 1, \\ 4(y-1)e^{0.3(1-l_2^2)} & \text{if } x > 1, z > 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$v(x, y, z, 0) = \begin{cases} -4(x-c_1)e^{0.3(1-l_1^2)} & \text{if } x < 1, z > 1, \\ -4(x-c_2)e^{0.3(1-l_2^2)} & \text{if } x > 1, z > 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$w(x, y, z, 0) = 0,$$

where  $l_1 = \sqrt{(x-c_1)^2 + (y-1)^2}/0.08$ ,  $l_2 = \sqrt{(x-c_2)^2 + (y-1)^2}/0.08$ ,  $c_1 = 0.8$ , and  $c_2 = 1.2$ . We use  $h = 1/30$ ,  $\Delta t = 2h^2$ , and  $Re = 3 \times 10^4$  in this simulation. As illustrated in Fig. 15, the initial two vortices deform and rotate away from each other. Because of the energy dissipation, the magnitude of vorticity field decreases in time.

Next, we change the initial distance between two vortices to investigate the dynamics of co-rotating vortex pair. We use  $c_1 = 0.9$  and  $c_2 = 1.1$  in this simulation. Other parameters take the same values as above. Fig. 16 shows the temporal evolution of vortex pair. If a closer distance between two vortices is used, then the two vortices quickly merge with each other and the vorticity deforms and dissipates in time.

#### 4.11. Taylor–Green vortices on a flat surface

For a flat surface, we define  $d(x, y, z) = z - \pi$  in the domain  $\Omega = (0, 2\pi)^3$ . The initial conditions of Taylor–Green vortices are defined as:

$$u(x, y, z, 0) = -\cos(x) \sin(y),$$

$$v(x, y, z, 0) = \sin(x) \cos(y),$$

$$w(x, y, z, 0) = 0.$$

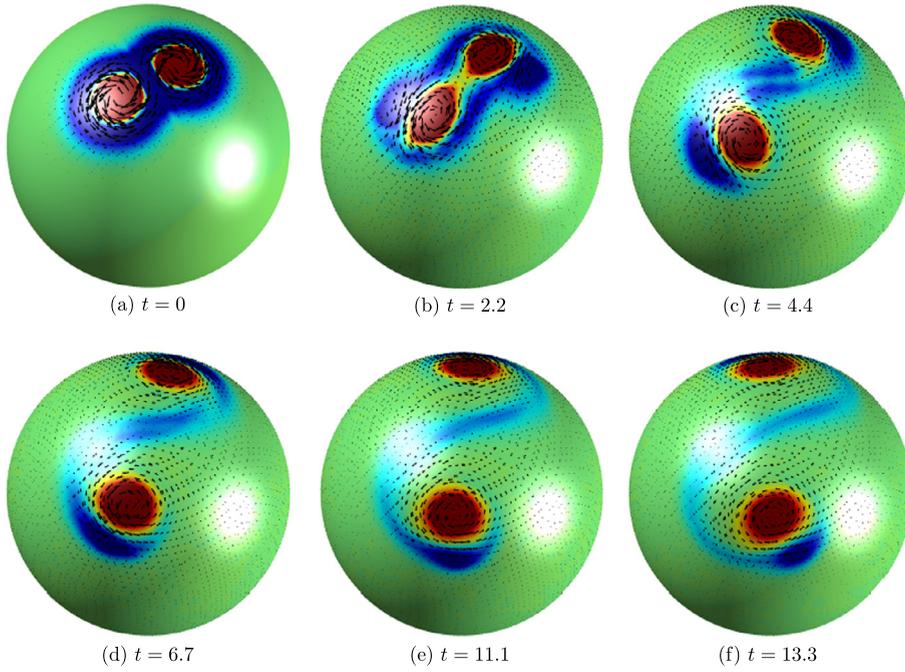


Fig. 15. Temporal evolutions of co-rotating vortex pair with  $c_1 = 0.8$  and  $c_2 = 1.2$ . The computational times are illustrated below each figure.

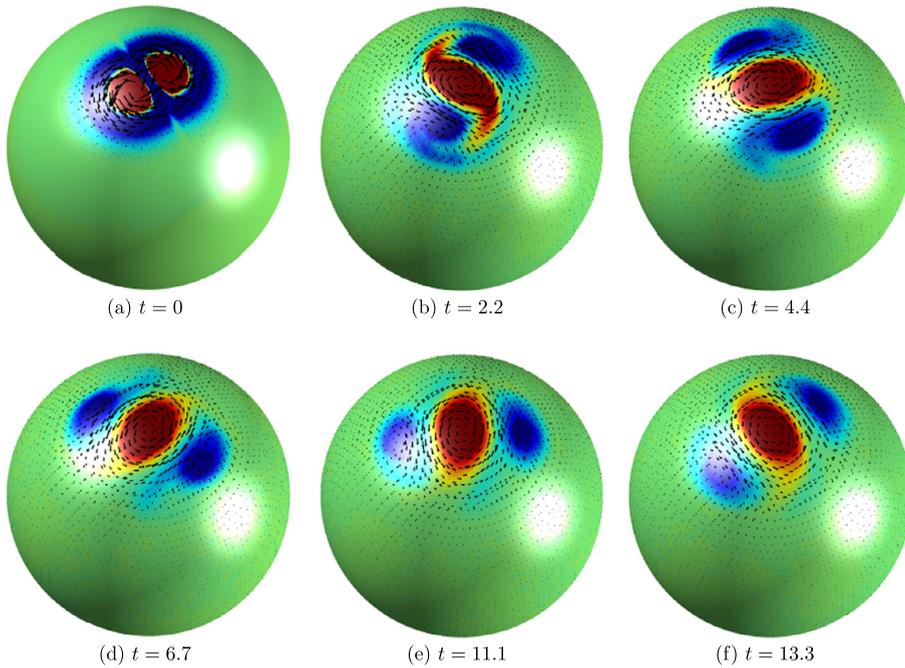


Fig. 16. Temporal evolutions of co-rotating vortex pair with  $c_1 = 0.9$  and  $c_2 = 1.1$ . The computational times are shown below each figure.

Here, we use  $h = 2\pi/60$ ,  $\Delta t = 3.2e-5$ , and  $Re = 100$ . The analytical decay of two-dimensional Taylor–Green vortices with time is expressed as [32,33]

$$u(x, y, t) = -\cos(x) \sin(y) e^{-2\nu t}, \tag{20}$$

$$v(x, y, t) = \sin(x) \cos(y) e^{-2\nu t}, \tag{21}$$

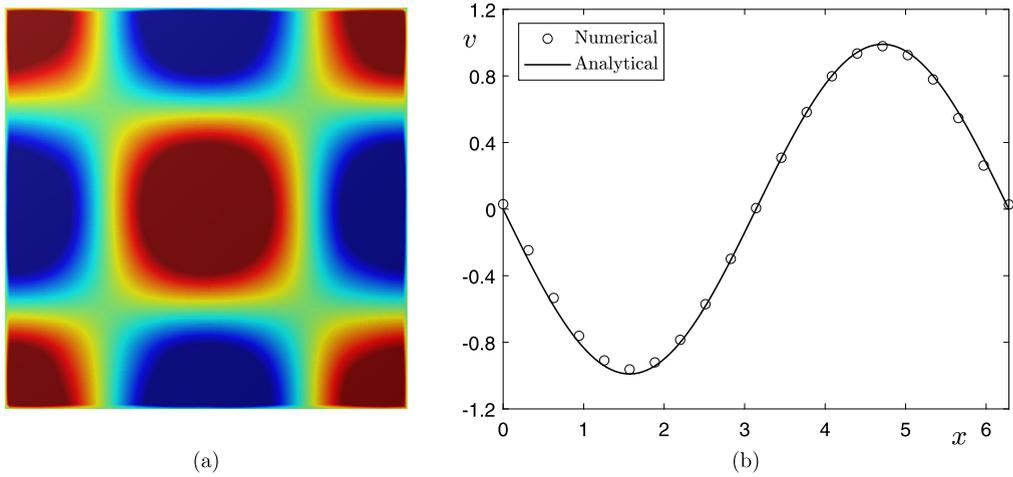


Fig. 17. (a) Evolution of Taylor-Green vortices on a flat surface at  $t = 0.5$ . (b) Numerical and analytical velocities along  $y = \pi$ .

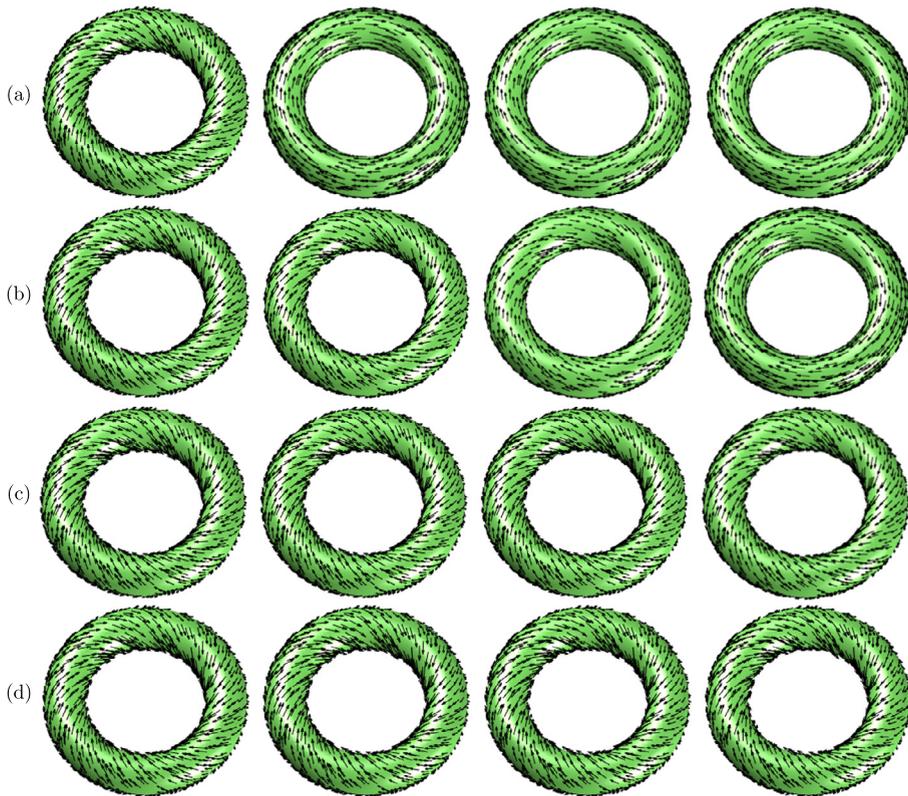


Fig. 18. Temporal evolutions of velocity field on a torus with different  $Re$ : (a) 1, (b) 10, (c) 100, and (d) 1000. The computational times from the left to right in each row are:  $t = 0, 0.11, 0.44,$  and  $0.88$ .

where the viscosity  $\nu = 0.01$  is used. Fig. 17(a) illustrates the Taylor-Green vortices on a flat surface at  $t = 0.5$ . The numerical and analytical velocity values at  $y = \pi$  are shown in Fig. 17(b). We can see that the simulation and analytical results are in good agreement with each other.

#### 4.12. Stability test with respect to different Reynolds numbers

We then investigate the effect of  $Re$  on the flow on a torus surface combining with the second-order spatial accurate scheme. The torus surface is defined in Section 4.7 and we use  $h = 1/10, \Delta t = 0.1h^2$  in the domain  $\Omega = (0, 8)^3$ . The following different values of  $Re$  are considered:  $Re = 1, 10, 100,$  and  $1000$ . Fig. 18(a), (b), (c), and (d) illustrate the temporal

evolution of flow field on a torus with respect to different values of  $Re$ , we can observe that the proposed method can work well with a range of  $Re$  numbers.

## 5. Conclusions

A practical finite difference method for solving the incompressible NS equation on three-dimensional curved surfaces was proposed in this paper. The NS equation on surface was extended into a three-dimensional narrow band domain neighborhood of a curved surface which is defined by the zero level-set of a signed distance function. By using the closet-point method and applying the pseudo-Neumann boundary condition, the standard Laplacian operator can be used to replace the discrete Laplace–Beltrami operator. To obtain the velocity field parallel to the surface, a velocity correction step was used. Various numerical tests were performed on curved surfaces, which demonstrated that our proposed method satisfies the divergence-free condition and the energy dissipation, and almost first-order accuracy in space and time. In the upcoming works, we will investigate the numerical schemes for simulating the two-phase fluid flows on curved surfaces with high order accuracy. Furthermore, we will use the staggered marker and cell (MAC) mesh to enforce the divergence-free velocity field.

## Declaration of competing interest

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

## Acknowledgement

J. Yang is supported by China Scholarship Council (201908260060). Y.B. Li is supported by National Natural Science Foundation of China (No. 11601416, No. 11631012). The corresponding author (J.S. Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1A2C1003053). The authors greatly appreciate the reviewers for their constructive comments and suggestions, which have improved the quality of this paper.

## References

- [1] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* 2 (1967) 12–26.
- [2] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 85 (1989) 257–283.
- [3] M.-J. Ni, M.A. Abdou, A bridge between projection methods and SIMPLE type methods for incompressible Navier–Stokes equations, *Int. J. Numer. Methods Eng.* 72 (2007) 1490–1512.
- [4] F. Zhang, J. Cheng, T. Liu, A direct discontinuous Galerkin method for the incompressible Navier–Stokes equations on arbitrary grids, *J. Comput. Phys.* 380 (2019) 269–294.
- [5] L. Lin, Z. Yang, S. Dong, Numerical approximation of incompressible Navier–Stokes equations based on an auxiliary energy variable, *J. Comput. Phys.* 388 (2019) 1–22.
- [6] P. Suchde, J. Kuhnert, S. Tiwari, On meshfree GFDM solvers for the incompressible Navier–Stokes equations, *Comput. Fluids* 165 (30) (2018) 1–12.
- [7] E. Castillo, R. Codina, Dynamic term-by-term stabilized finite element formulation using orthogonal subgrid-scales for the incompressible Navier–Stokes problem, *Comput. Methods Appl. Mech. Eng.* 349 (1) (2019) 701–721.
- [8] J. Martí, P.B. Ryzhakov, An explicit-implicit finite element model for the numerical solution of incompressible Navier–Stokes equations on moving grids, *Comput. Methods Appl. Mech. Eng.* 350 (15) (2019) 750–765.
- [9] A. Lozovskiy, M.A. Olshanskii, Y.V. Vassilevski, A quasi-Lagrangian finite element method for the Navier–Stokes equations in a time-dependent domain, *Comput. Methods Appl. Mech. Eng.* 333 (1) (2018) 55–73.
- [10] C. Nisters, A. Schwarz, Efficient stress-velocity least-squares finite element formulations for the incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Eng.* 341 (1) (2018) 333–359.
- [11] Q. Du, L. Ju, L. Tian, Finite element approximation of the Cahn–Hilliard equation on surfaces, *Comput. Methods Appl. Mech. Eng.* 200 (29–32) (2011) 2458–2470.
- [12] Y. Li, J. Kim, N. Wang, An unconditionally energy-stable second-order time-accurate scheme for the Cahn–Hilliard equation on surfaces, *Commun. Nonlinear Sci. Numer. Simul.* 53 (2017) 213–227.
- [13] G. Dziuk, C.M. Elliott, Surface finite elements for parabolic equations, *J. Comput. Math.* 25 (4) (2007) 385–407.
- [14] J.B. Green, A.L. Bertozzi, G. Sapiro, Fourth order partial differential equations on general geometries, *J. Comput. Phys.* 216 (1) (2006) 216–246.
- [15] Y. Choi, D. Jeong, S. Lee, M. Yoo, J. Kim, Motion by mean curvature of curves on surfaces using the Allen–Cahn equation, *Int. J. Eng. Sci.* 97 (2015) 126–132.
- [16] H.G. Lee, J. Kim, A simple and efficient finite difference method for the phase-field crystal equation on curved surfaces, *Comput. Methods Appl. Mech. Eng.* 307 (2016) 32–43.
- [17] Y. Li, C. Luo, B. Xia, J. Kim, An efficient linear second order unconditionally stable direct discretization method for the phase-field crystal equation on surfaces, *Appl. Math. Model.* 67 (2019) 477–490.
- [18] D. Jeong, J. Kim, Microphase separation patterns in diblock copolymers on curved surfaces using a nonlocal Cahn–Hilliard equation, *Eur. Phys. J. E* 38 (2015) 117.
- [19] M.S. Mohamed, A.N. Hirani, R. Samtaney, Discrete exterior calculus discretization of incompressible Navier–Stokes equations over surface simplicial meshes, *J. Comput. Phys.* 312 (1) (2016) 175–191.
- [20] I. Nitschke, S. Reuther, A. Voigt, Discrete exterior calculus (DEC) for the surface Navier–Stokes equation, in: *Transport Processes at Fluidic Interfaces*, in: *Advances in Mathematical Fluid Mechanics*, Birkhäuser, Cham, 2017, pp. 177–197.
- [21] S. Reuther, A. Voigt, Solving the incompressible surface Navier–Stokes equation by surface finite elements, *Phys. Fluids* 30 (2018) 012107.
- [22] P.-H. Chiu, Y.-T. Lin, A conservative phase field method for solving incompressible two-phase flows, *J. Comput. Phys.* 230 (2011) 185–204.

- [23] P.-H. Chiu, A coupled phase field framework for solving incompressible two-phase flows, *J. Comput. Phys.* 392 (2019) 115–140.
- [24] C. Liu, F. Frank, C. Thiele, F.O. Alpak, S. Berg, W. Chapman, B. Riviere, An efficient numerical algorithm for solving viscosity contrast Cahn–Hilliard–Navier–Stokes system in porous media, *J. Comput. Phys.* 400 (2020) 108948.
- [25] J.B. Greer, An improvement of a recent Eulerian method for solving PDEs on general geometries, *J. Sci. Comput.* 29 (2006) 321–352.
- [26] D. Jeong, Y. Li, Y. Choi, M. Yoo, D. Kang, J. Park, J. Choi, J. Kim, Numerical simulation of the zebra pattern formation on a three-dimensional model, *Physica A* 475 (2017) 106–116.
- [27] C.B. Macdonald, J. Brandman, S.J. Ruuth, Solving eigenvalue problems on curved surfaces using the closest point method, *J. Comput. Phys.* 230 (2011) 7944–7956.
- [28] S.J. Ruuth, B. Merriman, A simple embedding method for solving partial differential equations on surfaces, *J. Comput. Phys.* 227 (2008) 1943–1961.
- [29] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, *J. Comput. Phys.* 83 (1989) 32–78.
- [30] J. Kim, An augmented projection method for the incompressible Navier–Stokes equations in arbitrary domains, *Int. J. Comput. Methods* 2 (2) (2005) 201–212.
- [31] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* 130 (2) (1997) 269–282.
- [32] A. Green, G. Taylor, Mechanism of the production of small eddies from larger ones, *Proc. R. Soc. A* 158 (1937) 499–521.
- [33] J. Connors, Convergence analysis and computational testing of the finite element discretization of the Navier–Stokes alpha model, *Numer. Methods Partial Differ. Equ.* 26 (6) (2010) 1328–1350.