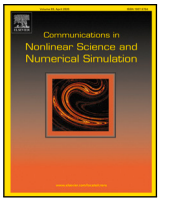
Commun Nonlinear Sci Numer Simulat 144 (2025) 108676

Contents lists available at ScienceDirect

Communications in Nonlinear Science and Numerical Simulation

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



Research paper

An efficient computational method for simulating incompressible fluid flows on a virtual cubic surface

Junxiang Yang [a], Seungyoon Kang [b], Sangkwon Kim [b], Youngjin Hwang [b], Soobin Kwak [b], Seokjun Ham [b], Junseok Kim [b,*]

[a] School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Macao Special Administrative Region of China

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

ARTICLE INFO

Keywords:
Navier-Stokes equation
Finite difference method
Virtual cubic surface

ABSTRACT

We propose an efficient computational algorithm for simulating incompressible fluid flows on a virtual cubic surface. By neglecting the influence of gravitational force, we effectively eliminate its impact on the system. Therefore, the dynamics can be characterized as essentially two-dimensional (2D) and confined to a plane. A projection method and a finite difference method (FDM) are used to numerically solve the Navier–Stokes (NS) equations governing the fluid flow. In the projection method, we need to solve the Poisson equation for the pressure field, which is effectively solved using a multigrid method. The multigrid method is a powerful computational approach that optimizes the solution process by using a hierarchical grid structure. The multigrid method allows efficient and accurate calculations of pressure values and increases the overall simulation accuracy and performance. The effectiveness of the proposed numerical algorithm is demonstrated through computational results of the shear flow and multi-vortex problems on a virtual cubic surface. These computational results validate the reliability and efficiency of the proposed approach, and therefore demonstrate its potential to contribute significantly to the field of computational analysis in fluid dynamics on a virtual cubic surface.

1. Introduction

The Navier–Stokes (NS) equation is a system of partial differential equations that describe the movement of incompressible fluid with viscosity [1]. As the complexity of fluid flow increases, the non-linearity of the NS equation intensifies, and makes analytical solutions harder or even impossible to obtain. Analytical solutions are only applicable in certain special cases or under simple assumptions [2–4]. Consequently, due to the limited applicability of analytical solutions, a wide range of numerical methods have been developed and extensively used to solve the NS equation [5–13]. Furthermore, multiphase fluid flows have been analyzed using the Navier–Stokes equation combined with the phase-field model [14–17]. Marynets [18] proposed a mathematical equation for the equatorial current across the Pacific Ocean based on the rotating NS equation, which incorporates Coriolis effects and mass conservation. However, many studies involving three-dimensional surface NS flow require heavy computations, which can sometimes be inefficient. One approach is to simplify the problem by first considering a cubic surface where each individual face has zero curvature, and then map it to the spherical surface with appropriate weighting. Weyn et al. [19] proposed a data-driven

* Corresponding author.
E-mail address: cfdkim@korea.ac.kr (J. Kim).
URL: https://mathematicians.korea.ac.kr/cfdkim/ (J. Kim).

https://doi.org/10.1016/j.cnsns.2025.108676
Received 6 October 2024; Received in revised form 7 February 2025; Accepted 8 February 2025
Available online 15 February 2025

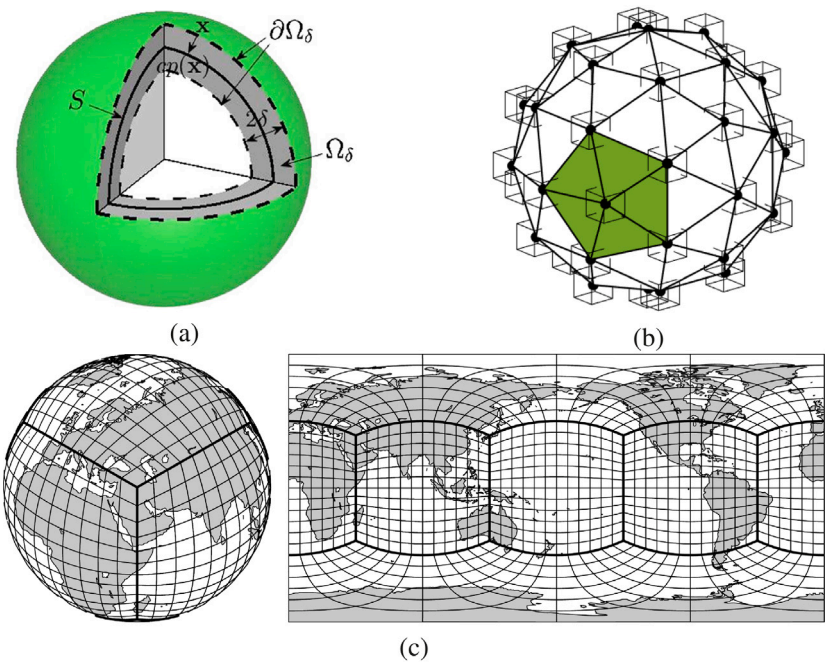


Fig. 1. (a) Schematic diagram of closet point method. Reprinted from [24] with permission from Elsevier. (b) Schematic diagram of finite volume lattice Boltzmann method [25]. (c) Gnomonic equiangular cubed-sphere grid. Reprinted from [26] with permission from Elsevier.

global weather forecasting framework that uses cubed-sphere remapping. Alternatively, a cubed sphere can also be considered as an alternative option [20–23].

Numerically solving the NS equation on a three-dimensional (3D) surface requires a significant computational effort. Yang et al. [24] solved the viscous incompressible flow on a 3D surface using narrow band domain and the closest point method, as illustrated in Fig. 1(a). Yang et al. [25] proposed a finite volume lattice Boltzmann method (LBM) for computing flows on curved surfaces in 3D space. Using unstructured triangular meshes for spatial discretization and the D3Q19 lattice for velocity discretization, Fig. 1(b), the method efficiently computed distribution functions explicitly at each vertex per time iteration. Shashkin et al. [26] applied the summation-by-parts finite difference (SBP-FD) approach to solve the shallow water equations on the gnomonic equiangular cubed-sphere grid as shown in Fig. 1(c). However, this approach requires a thorough understanding and application of the fundamental principles of grids in curvilinear coordinate systems. Specifically, the need to define covariant basis vectors and the metric tensor adds complexity to the formulation. However, the above-mentioned approaches are computationally costly due to the complexity of their solution algorithms.

In this study, we propose a mathematical equation and its numerical solution algorithm for approximating fluid flows on spherical surfaces with small curvature which is embedded in a 3D space. The spherical surface is approximated by a virtual cubic domain represented as a two-dimensional domain, on which the computation can be performed more efficiently and rapidly. Because the finite difference method is combined with a multigrid method in our numerical approach, the computational cost or complexity of the proposed algorithm scales linearly with the number of node points, denoted as N. This results in a computational complexity of $O(N)$ [27].

The rest of the paper is as follows. We first describe the governing equations and the numerical solution algorithm in Section 2. Section 3 presents numerical experiments such as the shear flow and multi-vortex problems on a cubic surface. In Section 4, conclusions are given.

2. Governing equations and numerical method

We present an efficient computational scheme for computing viscous incompressible fluid flows on a virtual cubic surface. Firstly, we describe the fluid flow on the virtual cubic surface, and then we perform dimensionality reduction by considering the template of the cube. The time-dependent non-dimensional NS equations for a two-dimensional incompressible fluid are expressed as follows:

$$u_t(x,y,t) + u(x,y,t)u_x(x,y,t) + v(x,y,t)u_y(x,y,t) = -p_x(x,y,t) \tag{1}$$

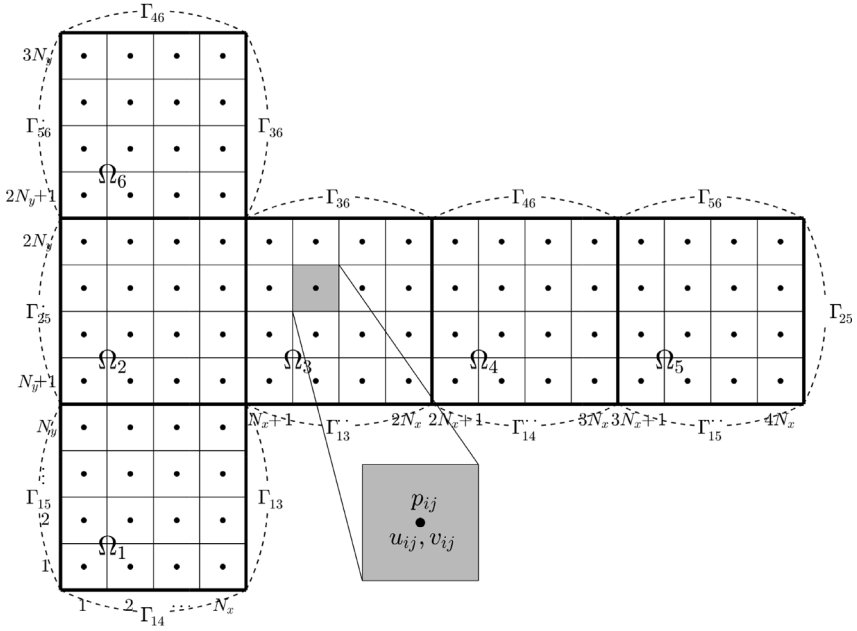


Fig. 2. Computational domain: collocated grid for the unfolded cubic surface.

$$
\begin{aligned}
& & & + \frac{1}{Re}\Delta u(x,y,t), \\
v_t(x,y,t) + u(x,y,t)v_x(x,y,t) + v(x,y,t)v_y(x,y,t) &= -p_y(x,y,t) & & \\
& & & + \frac{1}{Re}\Delta v(x,y,t), \\
u_x(x,y,t) + v_y(x,y,t) &= 0, & & (3)
\end{aligned}
\tag{2}
$$

where $u(x,y,t)$ and $v(x,y,t)$ represent the velocity field and $p(x,y,t)$ is the pressure field of the incompressible viscous fluid. The Reynolds number Re is a dimensionless constant that describes the fluid flow configuration, defined as the ratio between inertial and viscous forces. The incompressibility condition is given by Eq. (3) [28]. We will extend the governing equation to investigate multiphase flows on cubic surfaces by using the Allen–Cahn equations [29,30].

We consider a 2D unfolded cube as the representation of the cubic surface. Instead of solving the NS equation on a 3D cubic surface, reducing the computational domain to a 2D unfolded cube offers improvements in efficiency and simplicity for solving the NS equation. On the T-shaped unfolded cube, which represents the unit cubic surface, we introduce a uniformly discretized domain with a mesh size of h for the discrete numerical scheme. Let Ω_k, $k = 1, 2, \ldots, 6$, denote the six faces of the unit cube. Each face is discretized into an $N_x \times N_y$ grid. The coordinates of each grid's center are given as $(x_i = (i-0.5)h,\ y_j = (j-0.5)h)$ for $1 \le i \le 4N_x$ and $1 \le j \le 3N_y$. Discrete velocity field (u_{ij}^n, v_{ij}^n) and pressure p_{ij}^n are defined on the center of each cell. The schematic illustration of discretization is shown in Fig. 2, and the same Γ_{ij} should be grouped together. This type of boundary condition is difficult to implement using the Fourier spectral method [31], although its high computational efficiency.

We note that when doubly periodic boundary conditions are applied to a rectangular domain, a virtual torus is obtained, as shown in Fig. 3(a) and (b), respectively. In the transformation process of a rectangular grid into a virtual torus, the rectangular grid squares are mapped onto the toroidal surface, which leads to noticeable distortions. However, in the proposed method, the unfolded cubic surface domain shown in Fig. 3(c) is transformed into a virtual cubic surface displayed in Fig. 3(d) without introducing distortions.

Applying the above-described discretization, Eqs. (1), (2) and (3) can be written in discrete equations:

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = -(uu_x + vu_y)_{ij}^n - (p_x)_{ij}^{n+1} + \frac{1}{Re}\Delta_d u_{ij}^n, \tag{4}$$

$$\frac{v_{ij}^{n+1} - v_{ij}^n}{\Delta t} = -(uv_x + vv_y)_{ij}^n - (p_y)_{ij}^{n+1} + \frac{1}{Re}\Delta_d v_{ij}^n, \tag{5}$$

$$(u_x)_{ij}^{n+1} + (v_y)_{ij}^{n+1} = 0. \tag{6}$$

For each time step, the goal is to find $(u_{ij}^{n+1}, v_{ij}^{n+1})$ and p^{n+1} from the previous time step's velocity data (u_{ij}^n and v_{ij}^n). The numerical algorithm of solving Eqs. (4), (5), and (6) for one time step is as follows.

Step 1. We solve Eqs. (4) and (5) without the pressure gradient term. By decoupling the pressure gradient term, we can efficiently determine the intermediate velocity field and proceed with the subsequent stages of our numerical solution algorithm. We compute

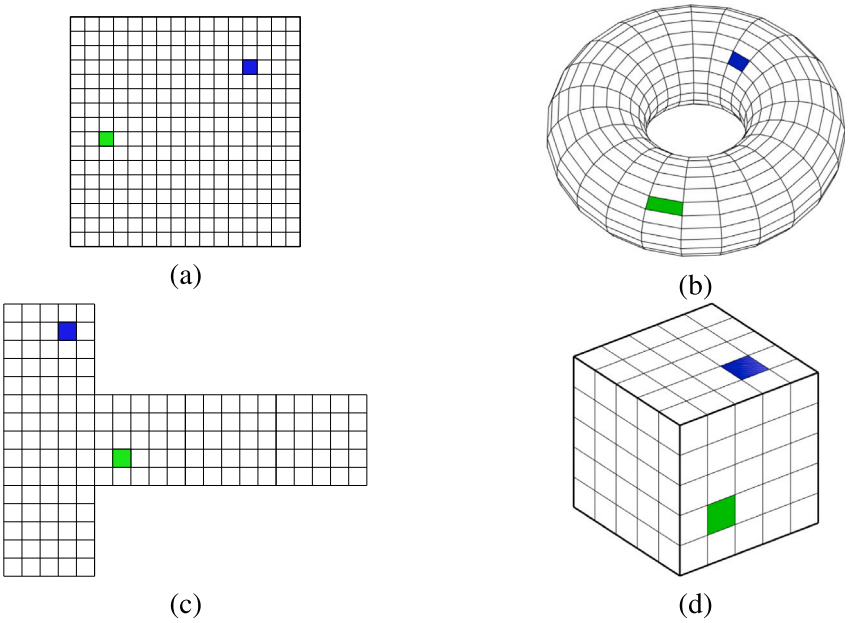


Fig. 3. (a) Rectangular domain with periodic boundary conditions and (b) the corresponding virtual torus. (c) Unfolded cubic surface domain and (d) the corresponding virtual cubic surface.

an intermediate velocity field $(\tilde{u}_{ij}, \tilde{v}_{ij})$ by solving the following equations:

$$\begin{aligned}\frac{\tilde{u}_{ij}-u^n_{ij}}{\Delta t} &= -(uu_x+vu_y)^n_{ij}+\frac{1}{Re}\Delta_d u^n_{ij},\\ \frac{\tilde{v}_{ij}-v^n_{ij}}{\Delta t} &= -(uv_x+vv_y)^n_{ij}+\frac{1}{Re}\Delta_d v^n_{ij},\end{aligned}$$

where the advection terms are defined as follows:

$$(uu_x+vu_y)^n_{ij}=u^n_{ij}\bar{u}^n_{x_{ij}}+v^n_{ij}\bar{u}^n_{y_{ij}},\qquad (uv_x+vv_y)^n_{ij}=u^n_{ij}\bar{v}^n_{x_{ij}}+v^n_{ij}\bar{v}^n_{y_{ij}}.$$

By applying the upwind procedure, we compute the quantities $\bar{u}^n_{x_{ij}}$ and $\bar{u}^n_{y_{ij}}$ as follows:

$$\bar{u}^n_{x_{ij}}=\begin{cases}\frac{u^n_{ij}-u^n_{i-1,j}}{h} & \text{if } u^n_{ij}>0,\\ \frac{u^n_{i+1,j}-u^n_{ij}}{h} & \text{otherwise}\end{cases}\quad\text{and}\quad \bar{u}^n_{y_{ij}}=\begin{cases}\frac{u^n_{ij}-u^n_{i,j-1}}{h} & \text{if } v^n_{ij}>0,\\ \frac{u^n_{i,j+1}-u^n_{ij}}{h} & \text{otherwise.}\end{cases}$$

Similarly, we compute quantities $\bar{v}^n_{x_{ij}}$ and $\bar{v}^n_{y_{ij}}$ as follows:

$$\bar{v}^n_{x_{ij}}=\begin{cases}\frac{v^n_{ij}-v^n_{i-1,j}}{h} & \text{if } u^n_{ij}>0,\\ \frac{v^n_{i+1,j}-v^n_{ij}}{h} & \text{otherwise}\end{cases}\quad\text{and}\quad \bar{v}^n_{y_{ij}}=\begin{cases}\frac{v^n_{ij}-v^n_{i,j-1}}{h} & \text{if } v^n_{ij}>0,\\ \frac{v^n_{i,j+1}-v^n_{ij}}{h} & \text{otherwise.}\end{cases}$$

Applying advection terms $(uu_x+vu_y)^n_{ij}$ and $(uv_x+vv_y)^n_{ij}$, the finite difference equations without the pressure gradient term can be expressed explicitly.

$$\begin{aligned}\tilde{u}_{ij} &= u^n_{ij}-\Delta t(uu_x+vu_y)^n_{ij}+\frac{\Delta t}{h^2 Re}\left(u^n_{i+1,j}+u^n_{i-1,j}-4u^n_{ij}+u^n_{i,j+1}+u^n_{i,j-1}\right),\\ \tilde{v}_{ij} &= v^n_{ij}-\Delta t(uv_x+vv_y)^n_{ij}+\frac{\Delta t}{h^2 Re}\left(v^n_{i+1,j}+v^n_{i-1,j}-4v^n_{ij}+v^n_{i,j+1}+v^n_{i,j-1}\right),\end{aligned}$$

where we have used the finite difference method [32], which is particularly efficient for problems defined on rectangular domains.

Step 2. The pressure field at the $(n+1)$th time step is determined by solving the equations given below.

$$\frac{\mathbf{u}^{n+1}_{ij}-\tilde{\mathbf{u}}_{ij}}{\Delta t} = -\nabla_d p^{n+1}_{ij}, \tag{7}$$

$$\nabla_d\cdot\mathbf{u}^{n+1}_{ij} = 0, \tag{8}$$

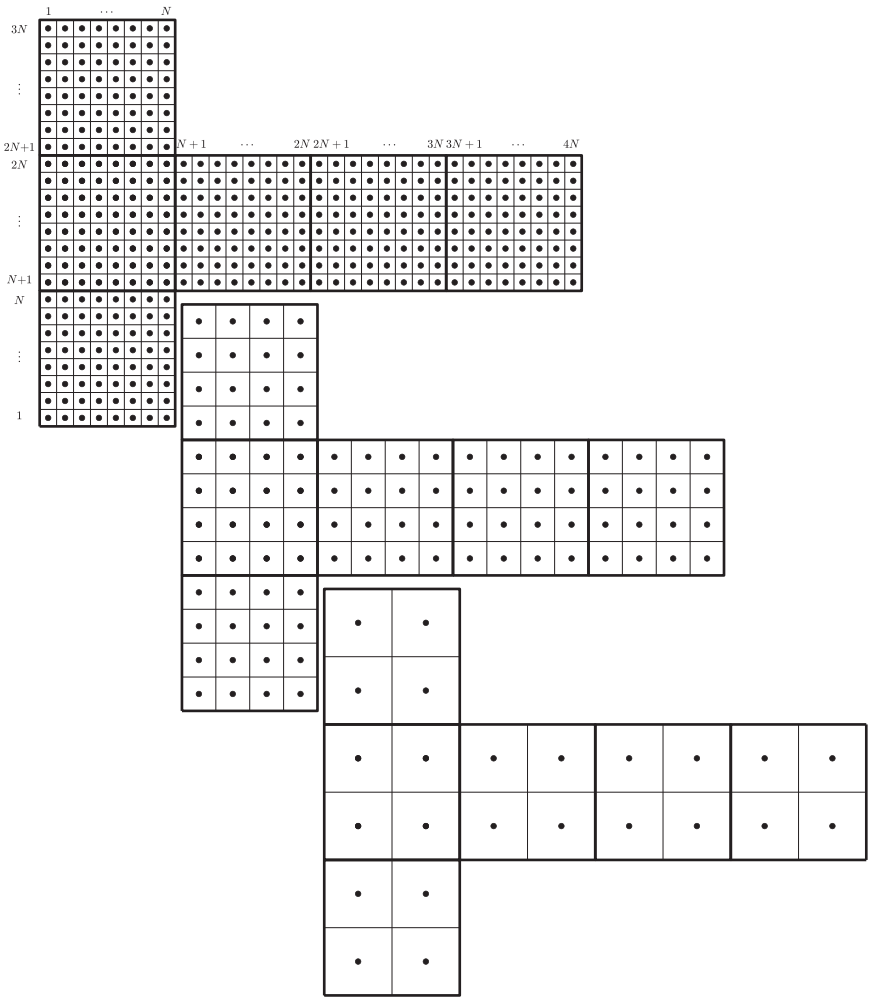


Fig. 4. From top to bottom, a sequence of coarse unfolded computational discrete domains: Ω_3, Ω_2, and Ω_1.

where $\mathbf{u}_{ij}^{n+1} = (u_{ij}^{n+1}, v_{ij}^{n+1})$. By taking the divergence operator to Eq. (7) and applying Eq. (8), Eq. (7) can be reformulated as a Poisson equation for the pressure field:

$$\Delta_d p_{ij}^{n+1} = \frac{1}{\Delta t} \nabla_d \cdot \tilde{\mathbf{u}}_{ij}. \tag{9}$$

Eq. (9) is solved using a multigrid algorithm with a Gauss–Seidel-type relaxation. The details of this algorithm will be described later.

Step 3. Lastly, the divergence-free normal velocities u^{n+1} and v^{n+1} are defined by

$$u_{ij}^{n+1} = \tilde{u}_{ij} - \frac{\Delta t}{2h}(p_{i+1,j}^{n+1} - p_{i-1,j}^{n+1}), \quad v_{ij}^{n+1} = \tilde{v}_{ij} - \frac{\Delta t}{2h}(p_{i,j+1}^{n+1} - p_{i,j-1}^{n+1}).$$

We repeat these three steps for each time step to compute the incompressible flow on a virtual cubic surface.

Now, we will describe the multigrid algorithm for solving the Poisson Eq. (9) on a 2D unfolded cube. To provide a clear explanation of the steps used in a single V-cycle, let us consider an 8×8 mesh on each side of the two-dimensional unfolded cube. Let Ω_3, Ω_2, Ω_1, and Ω_0 be four distinct discrete domains, where

$$\Omega_k = \{(x_{k,i} = (i - 0.5)h_k, y_{k,j} = (j - 0.5)h_k) | 1 \le i, j \le 2^{k+1}, \ h_k = 2^{3-k}h\}$$
$$\text{for } k = 0, 1, 2, 3.$$

Each discrete domain Ω_k has a finer grid than Ω_{k-1}, with a half grid step. Fig. 4 illustrates the mesh coarsening of these discrete domains. The multigrid algorithm incorporates discrete domains with different grid sizes. In the multigrid algorithm, we use the pointwise Gauss–Seidel relaxation method as the smoother. The linear system of Eq. (9) is solved by defining the discrete Laplace

operator [33] as

$$\Delta_d p_{ij}^{n+1} = \frac{p_{i+1,j}^{n+1} + p_{i-1,j}^{n+1} - 4p_{ij}^{n+1} + p_{i,j+1}^{n+1} + p_{i,j-1}^{n+1}}{h^2}$$

and the right-hand side of the equation

$$\nabla_d \cdot \tilde{\mathbf{u}}_{ij} = \frac{\tilde{u}_{i+1,j} - \tilde{u}_{i-1,j}}{2h} + \frac{\tilde{v}_{i,j+1} - \tilde{v}_{i,j-1}}{2h}.$$

Eq. (9) is solved using the following multigrid algorithm. Eq. (9) can be rewritten by

$$L_3(p_{3,ij}) = f_{3,ij} \text{ on } \Omega_3, \tag{10}$$

where

$$L_3(p_{3,ij}^{n+1}) = \Delta_d p_{3,ij}^{n+1} \text{ and } f_{3,ij} = \frac{1}{\Delta t}\nabla_d \cdot \tilde{\mathbf{u}}_{3,ij}^{n}.$$

An iteration step starts with an initial condition $p_3^{n+1,0} = p_3^n$. For each multigrid cycle, the initial guess is provided from the previous time step.

Multigrid cycle

$$p_k^{n+1,m+1} = \text{MultigridCycle}(k, u_k^{n+1,m}, L_k, f_k, \nu),$$

where $p_k^{n+1,m}$ and $p_k^{n+1,m+1}$ are the approximations of p_k^{n+1} before and after a MultigridCycle. Here, ν is the number of smoothing relaxation sweeps. Now, we describe one MultigridCycle in detail.

Step (1) Presmoothing

– We compute $\bar{u}_k^{n+1,m}$ by applying ν relaxation steps to $u_k^{n+1,m}$:

$$\bar{p}_k^{n+1,m} = SMOOTH^{\nu}(p_k^{n+1,m}, L_k, f_k).$$

To calculate the approximate value of $\bar{p}_k^{n+1,m}$, smoothing steps are performed ν with source term f_k and relaxation operator $SMOOTH$. As the $SMOOTH$ relaxation operator, we use the Gauss–Seidel iterative method. First, we rewrite Eq. (10) as

$$L_k(p_{k,ij}^{n+1}) = f_{k,ij} \text{ on } \Omega_k, \tag{11}$$

$$p_{k,ij}^{n+1} = \left[-f_{k,ij} + \frac{p_{i+1,j}^{n+1} + p_{i-1,j}^{n+1} + p_{i,j-1}^{n+1} + p_{i,j-1}^{n+1}}{h^2}\right] \Big/ \left(\frac{4}{h^2}\right). \tag{12}$$

$p_{k,\alpha\beta}^{n+1}$ in Eq. (12) is replaced with $p_{k,\alpha\beta}^{n+1,m}$ if $(\alpha > i)$ or $(\alpha = i \text{ and } \beta > j)$, otherwise $\bar{p}_{k,\alpha\beta}^{n+1,m}$, i.e.,

$$\bar{p}_{k,ij}^{n+1,m} = \left[-f_{k,ij} + \frac{p_{i+1,j}^{n+1,m} + \bar{p}_{i-1,j}^{n+1,m} + p_{i,j+1}^{n+1,m} + \bar{p}_{i,j-1}^{n+1,m}}{h^2}\right] \Big/ \left(\frac{4}{h^2}\right). \tag{13}$$

Solving Eq. (13) for $2^{k-3}N_x \times 2^{k-3}N_y$ grid completes one smooth relaxation operator step in a MultigridCycle.

Step (2) Correction of coarse grid

- Defect computation: $\bar{d}_k^m = f_k - L_k(\bar{p}_k^{n+1,m})$.
- Defect and $\bar{p}_k^m$ restriction: $\bar{d}_{k-1}^m = I_k^{k-1}\bar{d}_k^m$

k-level functions are mapped to $(k-1)$-level functions using the restriction operator I_k^{k-1}.

$$\begin{aligned} d_{k-1}(x_i, y_j) &= I_k^{k-1} d_k(x_i, y_j) \\ &= \frac{1}{4}[d_k(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}) + d_k(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) + d_k(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) + d_k(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})]. \end{aligned}$$

- On the coarse grid Ω_{k-1}, compute an approximate solution$\hat{p}_{k-1}^{n+1,m}$, i.e.,

$$L_{k-1}(p_{k-1}^{n+1,m}) = \bar{d}_{k-1}^m. \tag{14}$$

Eq. (14) is solved with an iterative solver. If $k > 1$, we use the zero grid function as an initial guess and perform k-grid cycles for an approximation, i.e.,

$$\hat{v}_{k-1}^{n+1,m} = MultigridCycle(k-1, 0, L_{k-1}, \bar{d}_{k-1}^m, \nu).$$

- Interpolate the correction: $\hat{q}_k^{n+1,m} = I_{k-1}^k \hat{q}_{k-1}^{n+1,m}$. That is $q_k(x_i, y_j) = I_{k-1}^k q_{k-1}(x_i, y_j) = q_{k-1}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ when i and j are odd numbers.
- Computation of corrected approximation on Ω_k

$$p_k^{m, \text{ after } CGC} = \bar{p}_k^{n+1,m} + \hat{q}_k^{n+1,m}.$$

Step (3) Postsmoothing: $p_k^{n+1,m+1} = SMOOTH^{\nu}(p_k^{m,\text{ after }CGC}, L_k, f_k)$.

Following these three steps completes one MultigridCycle.

The Poisson equation (9) is unique up to a constant with the periodic boundary condition. Applying a Dirichlet condition at a single point or fixing the total sum of pressure to be a constant can make a solution unique [34]. In our proposed method, the latter is chosen. The pressure field is corrected so that the total sum is zero.

$$p_{ij}^{n+1} = p_{ij}^{n+1} - \frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} p_{ij}^{n+1}. \tag{15}$$

The pressure correction is applied after each multigrid method.

Note that most numerical methods for the NS equations on two-dimensional domains can also be applied to virtual cubic surfaces. However, when solving the NS equations on real cubic surfaces, the application of numerical methods is limited due to the presence of singularities at the edges and corners of the cubic surfaces.

3. Numerical results

We perform various simulations of incompressible fluid flows on a cubic surface with different initial conditions to verify the performance and capability of the proposed model and its numerical method. Recall that the fluid flow on the cubic surface is solved in a 2D space. Therefore, a template for the unit cube as shown in Fig. 2 is the computational domain for the numerical tests.

3.1. Parabolic-type flow and stability analysis

We start with a parabolic type fluid flow to demonstrate the capability of the proposed method. A $1 \times 1 \times 1$ cube is considered. First example is fluid flow along the sides of the cube without an z-directional flow. Initial velocities $u(x, y, 0)$ and $v(x, y, 0)$ are given as follows:

$$u(x, y, 0) = \begin{cases} -(y-1)(y-2) & \text{for } 1 \le y \le 2, \\ 0 & \text{otherwise,} \end{cases}$$
$$v(x, y, 0) = 0.$$

The mesh size is $h = 1/32$ and we can construct the computational domain $\Omega_h = \bigcup_{i=1}^{6} \Omega_i$. Reynolds number $Re = 50$ is applied for this example. Based on the previously determined parameter values, we can define the time step size to achieve stability of the numerical scheme and avoid blow ups and oscillations. Time step restriction is determined by two factors [35]: advective Courant–Friedrichs–Lewy (CFL) condition and stability condition involving viscosity. The CFL condition is based on the ideal that every fluid particle's travel distance is bounded by the mesh size h [36]. Therefore, we have

$$\Delta t \le \frac{h}{\max_{(x_i,y_j)\in\Omega_h} |u(x_i, y_j, 0)|} \quad \text{and} \quad \Delta t \le \frac{h}{\max_{(x_i,y_j)\in\Omega_h} |v(x_i, y_j, 0)|}. \tag{16}$$

The other stability condition that involves viscosity is given as

$$\Delta t \le \frac{h^2 Re}{4}. \tag{17}$$

Incorporating Eqs. (16) and (17), we obtain

$$\Delta t \le \Delta t_{\max} = \min\Big(\frac{h}{\max_{(x_i,y_j)\in\Omega_h} |u(x_i, y_j, 0)|}, \frac{h}{\max_{(x_i,y_j)\in\Omega_h} |v(x_i, y_j, 0)|}, \frac{h^2 Re}{4}\Big).$$

Now that $\Delta t_{\max}$ is the maximum time step restriction, we multiply a safety factor $\tau \in (0, 1]$ for a stable time step $\Delta t = \tau \Delta t_{\max}$.

Applying a stable time step size $\Delta t = 0.01$, we perform the numerical test until time $t = 30$. Fig. 5 displays both the initial condition and snapshot images of the fluid flow on cubic surfaces at $t = 30$ for both the 2D template and 3D cubic surface. As the simulation progresses, the flow on the sides of the cube transitions to the top and bottom faces, as expected. The fluid flow on the top and bottom is shown as a circular swirling shape.

We consider the convergence of the numerical scheme on the virtual cubic surface. To perform a convergence test with respect to time, we used the parameters $\Delta t = 0.0005$, $N = 128$, $Re = 20$, and the final time $t = 0.002$. The initial velocities are identical to those used in the previous simulation. The reference solution $\tilde{u}_{ij}$ for the velocity u was generated using $\Delta t_{ref} = 6.25\text{e}{-5}$ and $N = 128$. The numerical error with the temporal step size Δt for the velocity u on Ω_h is defined as $e_{ij}^{\Delta t} = u_{ij}^{N_t} - \tilde{u}_{ij}$, where $N_t = t/\Delta t$. The l_1- and l_2-norms of the error for the temporal step size are defined as

$$\left\|e^{\Delta t}\right\|_1 = \frac{1}{6N^2}\sum_{(x_i,y_j)\in\Omega_h}\left|e_{ij}^{\Delta t}\right| \quad \text{and} \quad \left\|e^{\Delta t}\right\|_2 = \sqrt{\frac{1}{6N^2}\sum_{(x_i,y_j)\in\Omega_h}\left(e_{ij}^{\Delta t}\right)^2},$$

respectively. Numerical errors and temporal convergence rates are given in Table 1. We can see that the temporal convergence rate is approximately 1 for both l_1 and l_2 errors.

The spatial convergence test is also performed in a similar manner using a reference solution $\tilde{u}_{ij}$ that has a fine spatial step size $h_{ref} = 1/256$ with $\Delta t = 10^{-4}$ and $N_t = 17$. To calculate the convergence rate, a set of decreasing spatial step sizes $h, h/2$ and $h/4$ are applied where $h = 1.56\text{e}{-2}$. For a computational result under a particular time step h, the error between the reference solution is

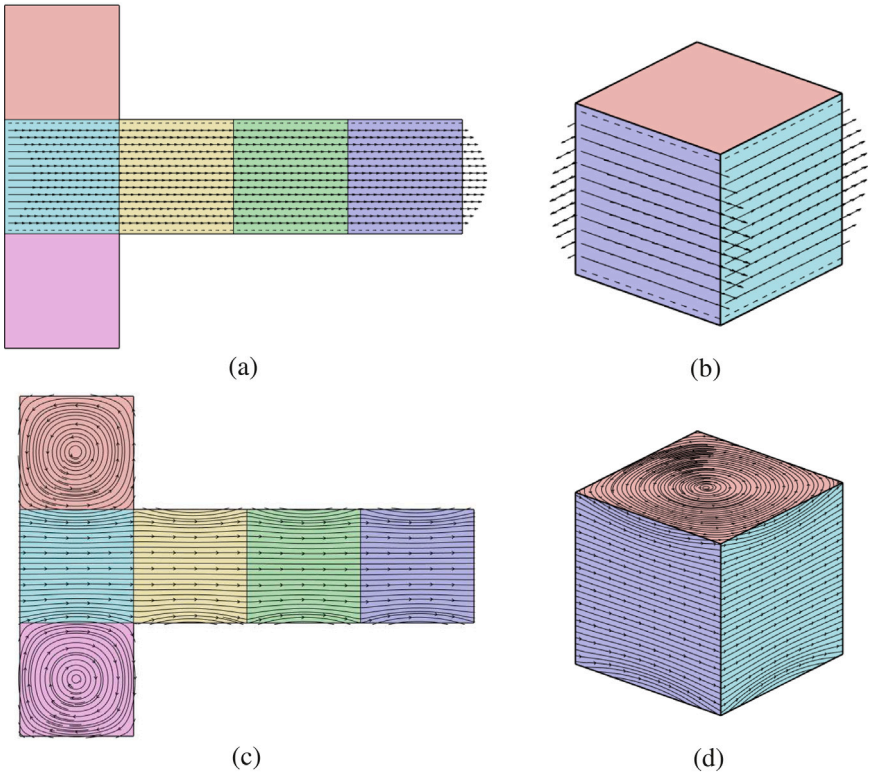


Fig. 5. Snapshot images of the parabolic-type flow at initial state (a) 2D and (b) 3D; (c) 2D and (d) 3D at time $t = 30$.

Table 1
Numerical errors and convergence rates of the proposed numerical algorithm for temporal step size.

Case	Δt	Rate	$\Delta t/2$	Rate	$\Delta t/4$
l_2-norm	1.0609e−5	1.10	4.9660e−6	1.28	2.0508e−6
l_1-norm	2.9180e−6	0.94	1.5166e−6	1.19	6.6418e-7

Table 2
Numerical errors and convergence rates of the proposed numerical algorithm for spatial step size.

Case	h	Rate	$h/2$	Rate	$h/4$
l_2-norm	1.3936e−4	2.03	3.4215e−5	1.82	9.6589e−6
l_1-norm	4.5733e−5	1.67	1.4485e−5	1.56	4.8823e-6

defined in the following way:

$$e_{ij}^h = u_{ij} - \Big(\tilde{u}_{2^p i-2^p,2^p j-2^p} + \tilde{u}_{2^p i-2^p+1,2^p j-2^p} + \tilde{u}_{2^p i-2^p,2^p j-2^p+1} + \tilde{u}_{2^p i-2^p+1,2^p j-2^p+1}\Big)/4,$$

where $p = 8, 4$ and 2 for tests using $h, h/2$ and $h/4$, respectively. Therefore, the l_1- and l_2-norms of the error for the spatial step size are defined as

$$\left\|e^h\right\|_1 = \frac{1}{6N_xN_y}\sum_{(x_i,y_j)\in\Omega_h}\left|e_{ij}^h\right| \quad \text{and} \quad \left\|e^h\right\|_2 = \sqrt{\frac{1}{6N_xN_y}\sum_{(x_i,y_j)\in\Omega_h}\left(e_{ij}^h\right)^2},$$

respectively. Numerical errors and spatial convergence rates are given in Table 2. The spatial convergence rates exceed one, although some remain below two. This is one of the limitations of current numerical schemes, and we will develop higher-order numerical schemes to overcome it. In this study, we focus on a new mathematical model for fluid flows on virtual cubic surfaces.

Next, the following example demonstrates the importance of time step restriction by applying an unstable time step $\Delta t = 1.01\Delta t_{\max}$. Fig. 6(a), (b) and (c) illustrate the snapshot images at initial state, $t = 50\Delta t$ and $t = 100\Delta t$, respectively. We can observe that the numerical solution at Fig. 6(c) does not agree with the stable solution shown in Fig. 5. It is also worth mentioning that the numerical solution blows up at finite time, $t = 104\Delta t$.

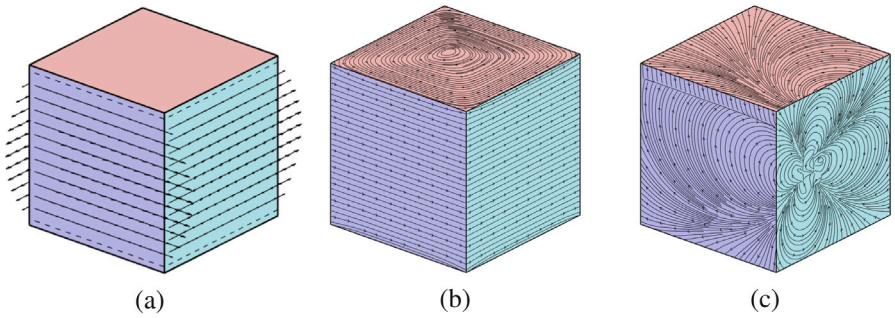


Fig. 6. 3D Snapshot images of the parabolic-type flow with unstable condition at (a) initial state, (b) $t = 50\Delta t$ and (c) $t = 100\Delta t$.

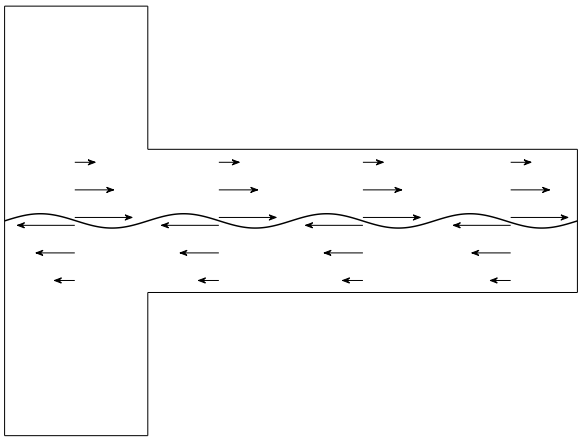


Fig. 7. Schematic illustration of the initial flow configuration.

3.2. Shear flow

We move on to the next problem, where we test the shear flow on a $1 \times 1 \times 1$ cube. Shear flow is fluid motion that arises between two superposed fluids with a relative horizontal velocity. In this study, the shear flow is simulated between two vertically separated fluids. The problem is defined on the unit cubic domain as follows: Initially, two fluids are separated vertically with a sinusoidal interface.

$$u(x, y, 0) = \begin{cases} 0.2\tanh(2 - y) & \text{for } y - 1.5 - 0.05\sin(2\pi x) \geq 0, \\ 0.2\tanh(y - 1) & \text{otherwise,} \end{cases}$$
$$v(x, y, 0) = 0.$$

The velocity of each flow is zero at the top and bottom faces and increases as it gets closer to the sinusoidal interface. Fig. 7 shows a schematic illustration of the initial flow configuration, where we can observe a velocity shear across the sinusoidal interface between the two fluids.

We set $h = 1/32$ and $\Delta t = 0.01$. A large Reynolds number, $Re = 1000$, is used for this section to observe the vortex flow at the final time $t = 30$. The computational results are illustrated in Fig. 8.

3.3. One rotating vortex

We consider the evolution of one rotating vortex on a $(0, 2) \times (0, 2) \times (0, 2)$ cubic domain. The initial conditions of velocity components are as follows:

$$u(x, y, 0) = -4(y - 5)e^{0.3(1-l_1^2)} \quad \text{and} \quad v(x, y, 0) = 4(x - 1)e^{0.3(1-l_1^2)},$$

where $l_1 = \sqrt{(x-1)^2 + (y-5)^2}/0.08$. We set $h = 1/32$, $\Delta t = 0.1$, and $Re = 10\,000$. The vorticity on the two-dimensional plane is calculated by $\omega = v_x - u_y$. The vorticity snapshots and streamlines at $t = 0, 20, 40$, and 60 are shown in Fig. 9. Vortex separation is not observed in this case.

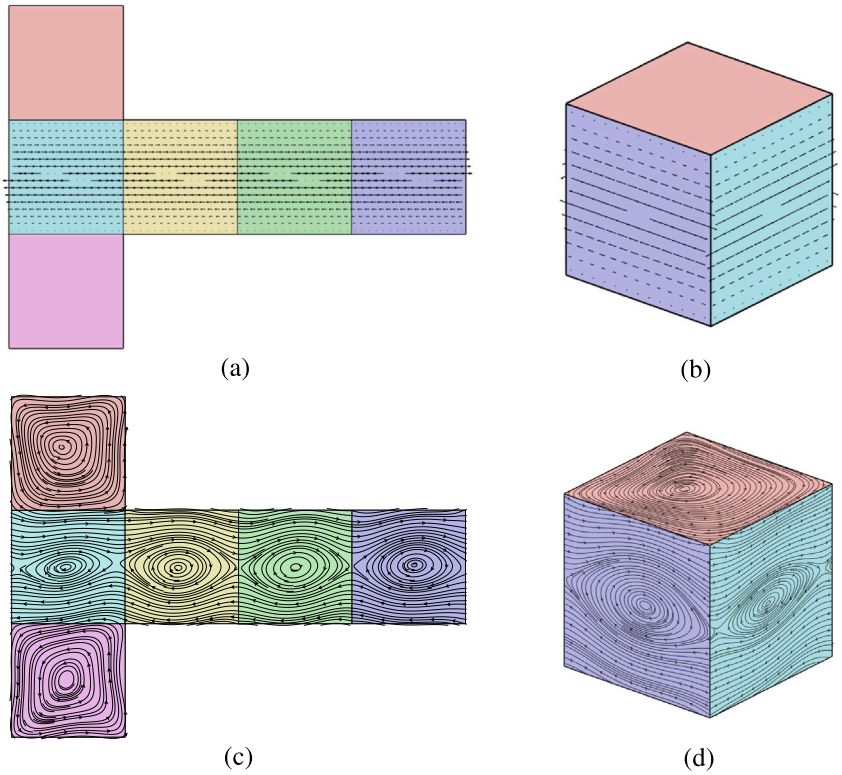


Fig. 8. Snapshot images of shear flow at initial state (a) 2D, (b) 3D and $t = 30$ (c) 2D, (d) 3D.

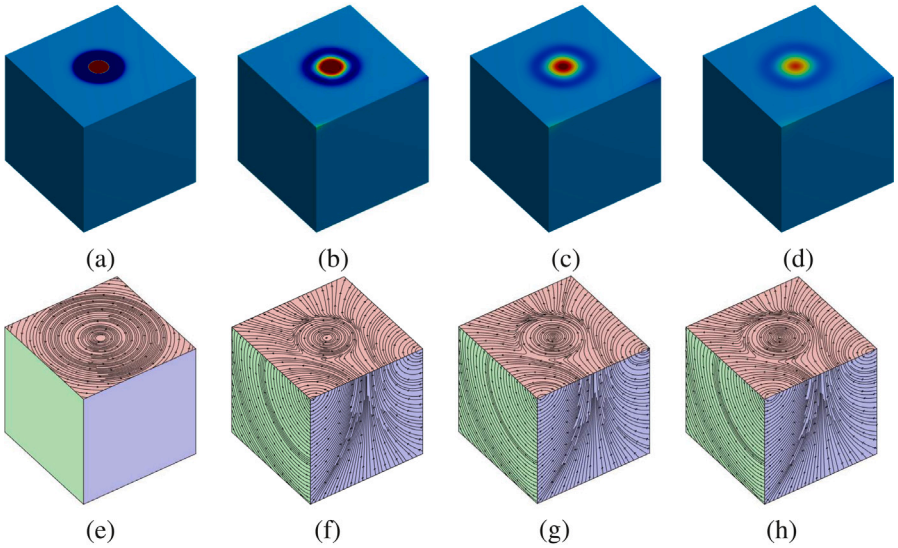


Fig. 9. Evolution of single vortex on a cube. Vorticity snapshots (a)–(d) and streamlines (e)–(h) are at $t = 0, 20, 40$, and 60.

3.4. Co-rotating vortex pairs

For an initially co-rotating vortex pair, the low viscosity leads to the vortex separation over time. By adopting a practical closest point-based FDM, Yang et al. [24] conducted the computation of vortex separation on a sphere. In this subsection, we simulate this benchmark test on the top surface of a cube. The domain size of a cube is $(0, 2) \times (0, 2) \times (0, 2)$. The initial conditions of velocity components are as follows:

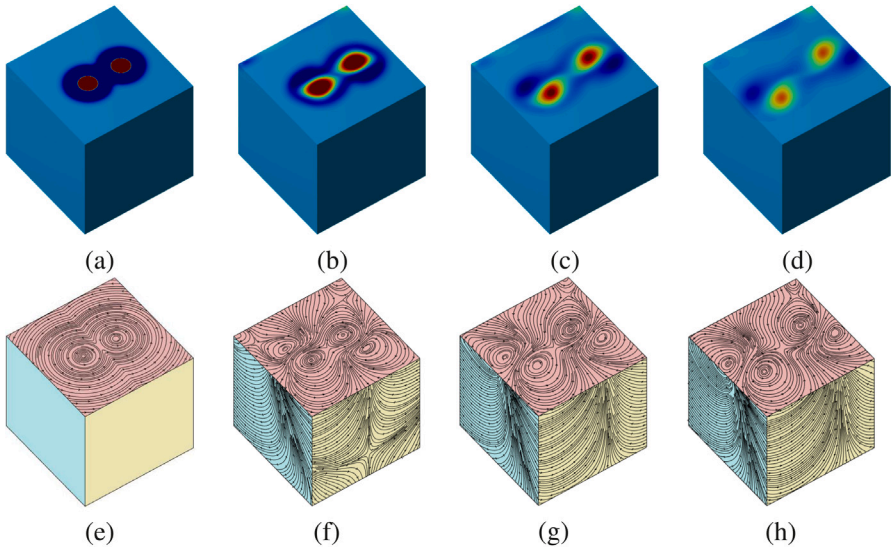


Fig. 10. Evolution of co-rotating vortex pair on a cube. Vorticity snapshots (a)–(d) and streamlines (e)–(h) are at $t = 0, 20, 40, 60$.

$$u(x,y,0) = \begin{cases} -4(y-4.7)e^{0.3(1-l_1^2)} & \text{if } y < 5, \\ -4(y-5.3)e^{0.3(1-l_2^2)} & \text{if } y > 5, \\ 0 & \text{otherwise} \end{cases}$$

and

$$v(x,y,0) = \begin{cases} 4(x-1)e^{0.3(1-l_1^2)} & \text{if } y < 5, \\ 4(x-1)e^{0.3(1-l_2^2)} & \text{if } y > 5, \\ 0 & \text{otherwise,} \end{cases}$$

where $l_1 = \sqrt{(x-1)^2 + (y-4.7)^2}/0.08$ and $l_2 = \sqrt{(x-1)^2 + (y-5.3)^2}/0.08$. The other conditions and parameter values are the same as the one vortex simulation. Fig. 10(a)–(d) show the vorticity at different moments. It can be observed that the vortices rotate away from each other in time. In Fig. 10(e)–(h), streamlines at $t = 0, 20, 40$, and 60 are plotted.

Next, we consider the case where co-rotating vortex pair has unequal strengths, i.e., unequal velocities. One vortex has twice the velocity of the other. In order to simulate this benchmark test, the following initial conditions of velocity components are applied:

$$u(x,y,0) = \begin{cases} -4(y-4.7)e^{0.3(1-l_1^2)} & \text{if } y < 5, \\ -8(y-5.3)e^{0.3(1-l_2^2)} & \text{if } y > 5, \\ 0 & \text{otherwise} \end{cases}$$

and

$$v(x,y,0) = \begin{cases} 4(x-1)e^{0.3(1-l_1^2)} & \text{if } y < 5, \\ 8(x-1)e^{0.3(1-l_2^2)} & \text{if } y > 5, \\ 0 & \text{otherwise.} \end{cases}$$

Other conditions are the same as former tests. Fig. 11 illustrates the computational results of velocity and streamline in different times. In Fig. 11, the stronger vortex is on the upper-right side and the weak vortex is on the lower-left side. The results show that vortices move away from each other in a manner consistent with the former test. However, the stronger vortex tends to transform and rotate more widely than the weak vortex. The trajectories in Figs. 10 and 11 match with the figures given in [37].

3.5. Counter-rotating vortex pairs

When studying vortex pair flows, interpreting the vorticity dynamics of the counter-rotating vortex model is as important as the co-rotating vortex model. Here, we consider counter-rotating equal and unequal vortex pairs with a simple adjustment in initial velocity components. Counter-rotating equal vortex pairs are simulated using the following initial condition:

$$u(x,y,0) = \begin{cases} -4(y-4.7)e^{0.3(1-l_1^2)} & \text{if } y < 5, \\ 4(y-5.3)e^{0.3(1-l_2^2)} & \text{if } y > 5, \\ 0 & \text{otherwise} \end{cases}$$

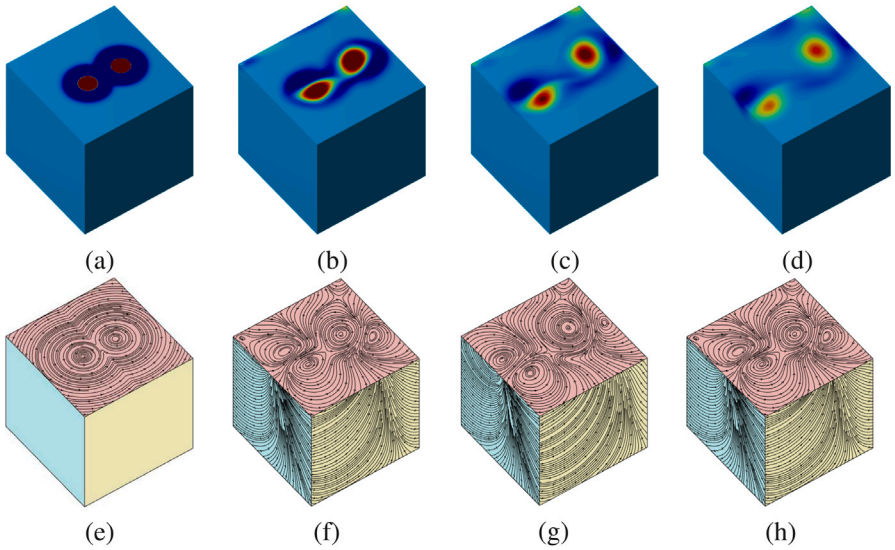


Fig. 11. Evolution of co-rotating unequal vortex pair on a cube. Vorticity snapshots (a)–(d) and streamlines (e)–(h) are at $t = 0, 20, 40$, and 60.

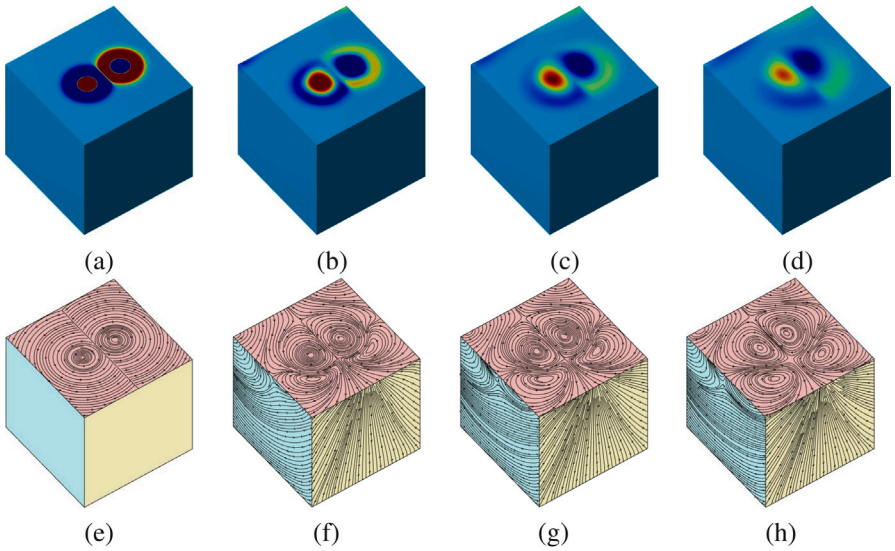


Fig. 12. Evolution of counter-rotating vortex pair on a cube. Vorticity snapshots (a)–(d) and streamlines (e)–(h) are at $t = 0, 20, 40$, and 60.

$$\text{and} \quad v(x,y,0) = \begin{cases} 4(x-1)e^{0.3(1-l_1^2)} & \text{if } y < 5, \\ -4(x-1)e^{0.3(1-l_2^2)} & \text{if } y > 5, \\ 0 & \text{otherwise.} \end{cases}$$

Unequal vortex pairs with twice different velocities are simulated using the following initial conditions:

$$u(x,y,0) = \begin{cases} -4(y-4.7)e^{0.3(1-l_1^2)} & \text{if } y < 5, \\ 8(y-5.3)e^{0.3(1-l_2^2)} & \text{if } y > 5, \\ 0 & \text{otherwise} \end{cases}$$

$$\text{and} \quad v(x,y,0) = \begin{cases} 4(x-1)e^{0.3(1-l_1^2)} & \text{if } y < 5, \\ -8(x-1)e^{0.3(1-l_2^2)} & \text{if } y > 5, \\ 0 & \text{otherwise.} \end{cases}$$

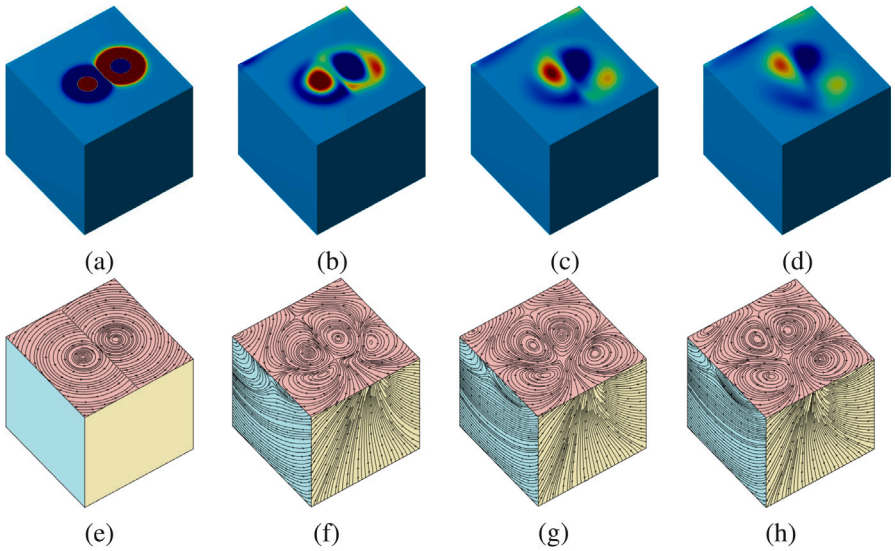


Fig. 13. Evolution of counter-rotating unequal vortex pair on a cube. Vorticity snapshots (a)–(d) and streamlines (e)–(h) are at $t = 0, 20, 40$, and 60.

The other conditions are the same as the test done in Section 3.4. Figs. 12 and 13 show the vorticity results at $t = 0, 20, 40$, and 60 for counter-rotating equal vortex pairs and counter-rotating unequal pairs, respectively. We can see that counter-rotating is more unstable and promotes vorticity decay during destruction. The trajectories in Figs. 12 and 13 match with the figures given in [37].

3.6. Three pairs of rotating vortices

Next, we study the dynamics at three pairs of rotating vortices. The following initial conditions are used

$$u(x,y,0) = \begin{cases} -4(y-4.7)e^{0.3(1-l_1^2)} & \text{if } y<5,\ x<1.1,\\ -4(y-5.3)e^{0.3(1-l_2^2)} & \text{if } y>5,\ x<1.1,\\ -4(y-5)e^{0.3(1-l_3^2)} & \text{if } x>1.1,\\ 0 & \text{otherwise} \end{cases}$$

$$\text{and}\quad v(x,y,0) = \begin{cases} 4(x-0.7)e^{0.3(1-l_1^2)} & \text{if } y<5,\ x<1.1,\\ 4(x-0.7)e^{0.3(1-l_2^2)} & \text{if } y>5,\ x<1.1,\\ 4(x-1.45)e^{0.3(1-l_3^2)} & \text{if } x>1.1,\\ 0 & \text{otherwise,} \end{cases}$$

where $l_1 = \sqrt{(x-0.7)^2+(y-4.7)^2}/0.08$, $l_2 = \sqrt{(x-0.7)^2+(y-5.3)^2}/0.08$, and $l_3 = \sqrt{(x-1.45)^2+(y-5)^2}/0.08$. Other parameters keep unchanged. The vorticities and streamlines at $t = 0, 20, 40$, and 60 are shown in the top and bottom rows in Fig. 14, respectively. We can clearly see vortex separation. These simulations indicate that the initial vortex pairs are essential to lead to the evolution of vortex separation.

The case involving a counter-rotating vortex is also tested. In this case, the values of l_1, l_2, and l_3 are identical to the previous test and the initial conditions are given as follows:

$$u(x,y,0) = \begin{cases} -4(y-4.7)e^{0.3(1-l_1^2)} & \text{if } y<5,\ x<1.1,\\ -4(y-5.3)e^{0.3(1-l_2^2)} & \text{if } y>5,\ x<1.1,\\ 4(y-5)e^{0.3(1-l_3^2)} & \text{if } x>1.1,\\ 0 & \text{otherwise} \end{cases}$$

$$\text{and}\quad v(x,y,0) = \begin{cases} 4(x-0.7)e^{0.3(1-l_1^2)} & \text{if } y<5,\ x<1.1,\\ 4(x-0.7)e^{0.3(1-l_2^2)} & \text{if } y>5,\ x<1.1,\\ -4(x-1.45)e^{0.3(1-l_3^2)} & \text{if } x>1.1,\\ 0 & \text{otherwise.} \end{cases}$$

Vorticity and streamline snapshots at $t = 0, 20, 40$, and 60 are illustrated in Fig. 15. The computational results are similar to the case where two vortices are counter-rotating. The vorticity is smaller and the results are unstable.

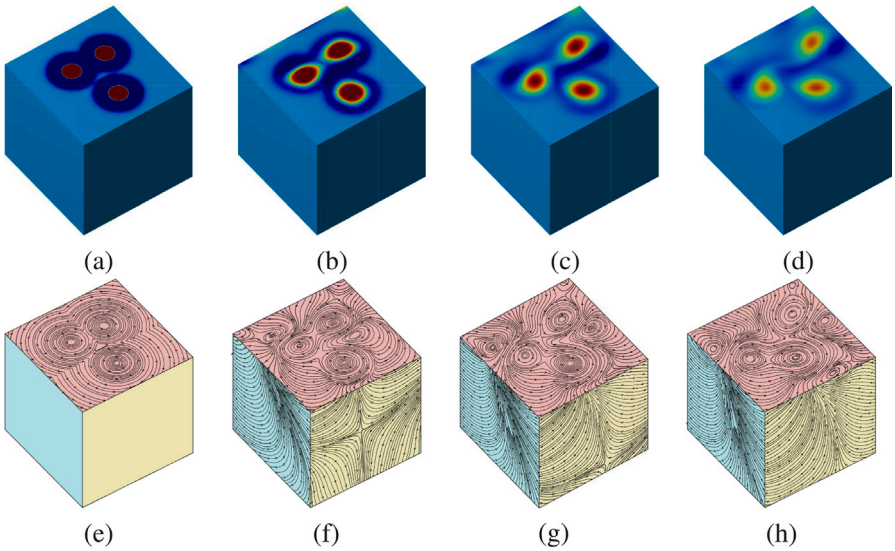


Fig. 14. Evolution of three rotating vortex pair on a cube. Vorticity snapshots (a)–(d) and streamlines (e)–(h) are at $t = 0, 20, 40$, and 60.

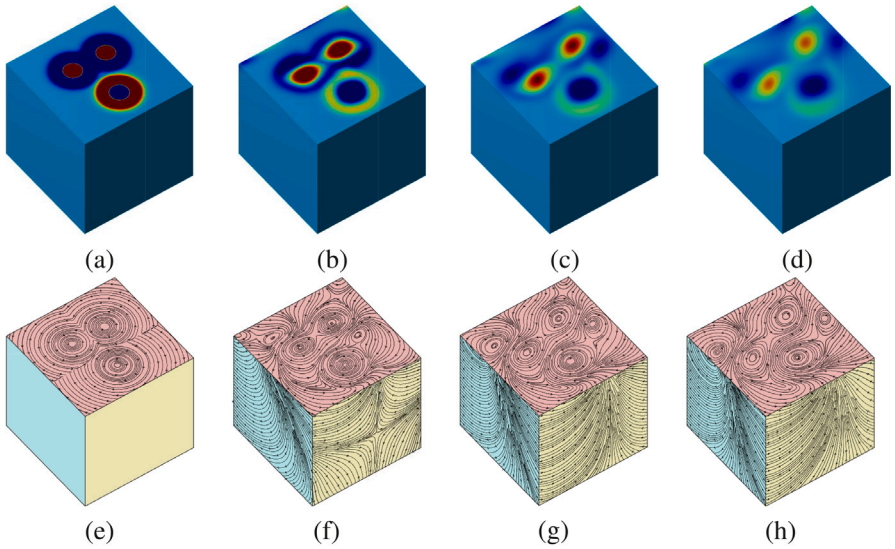


Fig. 15. Evolution of three rotating vortex pair with one counter-rotating vortex on a cube. Vorticity snapshots (a)–(d) and streamlines (e)–(h) are at $t = 0, 20$, 40, and 60.

3.7. Flow past a circular cylinder

In this subsection, we investigate the flow past a circular cylinder on the surface of a cube. The cube size is $(0, 1)\times(0, 1)\times(0, 1)$. To simulate the incompressible flow in the presence of a solid obstacle, we adopt the penalty treatment approach as outlined in [38]. This involves a reformulation of the NS equations and we can effectively account for the interaction between the fluid and the obstacle as follows:

$$\begin{aligned} &u_t(x,y,t) + u(x,y,t)u_x(x,y,t) + v(x,y,t)u_y(x,y,t) \\ &\quad = -p_x(x,y,t) + \frac{1}{Re}\Delta u(x,y,t) + f_s + \frac{\phi(x,y)}{\kappa}\left(u_s(x,y) - u(x,y,t)\right), \end{aligned} \tag{18}$$

$$\begin{aligned} &v_t(x,y,t) + u(x,y,t)v_x(x,y,t) + v(x,y,t)v_y(x,y,t) \\ &\quad = -p_y(x,y,t) + \frac{1}{Re}\Delta v(x,y,t) + \frac{\phi(x,y)}{\kappa}\left(v_s(x,y) - v(x,y,t)\right), \end{aligned} \tag{19}$$

$$u_x(x,y,t) + v_y(x,y,t) = 0, \tag{20}$$

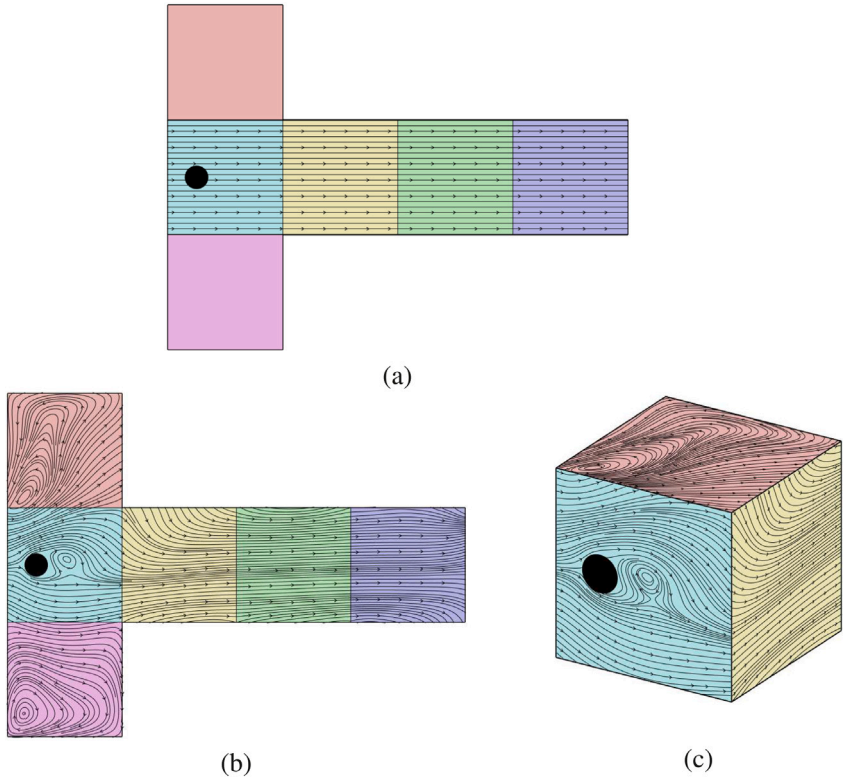


Fig. 16. (a) is an initial state of velocity field for the flow past a circular cylinder. (b) and (c) are 2D and 3D snapshots of flow past a circular cylinder at $t = 6$, respectively.

where $\phi(x, y)$ is an indicator function which takes the value 0 or 1 in the exterior or interior of the solid, κ is a small positive constant.

The circular solid with radius 0.1 is located at $(0.25, 1.5)$. The velocities of the solid are $u_s(x, y)$ and $v_s(x, y)$. To simulate the flow past a fixed cylinder, we set $u_s(x, y) = v_s(x, y) = 0$. Inspired by [39], we adopt $f_s = 0.5v(x, y, t)$ as a force term which drives the fluid flow. In numerical calculation, the last terms in Eqs. (18) and (19) are implicitly treated. The initial state is shown in Fig. 16(a). We perform simulations with $h = 1/64$, $\Delta t = 0.01$, $\kappa = 1e{-}8$, and $Re = 500$. In Fig. 16(b) and (c), a snapshot at $t = 6$ is plotted, it can be observed that the small vortex appears behind the solid cylinder. By referring Fig. 17, we can observe the formation and progression of a shedding vortex phenomenon as time progresses.

3.8. Comparison with other methodology

Fluid flow on spherical surfaces has been studied by numerous researchers over the past decades. For a comparison study, we consider a recent study [25] in which the authors proposed a finite volume lattice Boltzmann method for fluid flow on 3D curved surfaces. The reason for choosing the comparison between the LBM and the current method is that this study proposes a mathematical equation and its numerical solution algorithm designed to approximate fluid flows on spherical surfaces with small curvature embedded in a three-dimensional space. The spherical surface is approximated by a virtual cubic domain represented as a two-dimensional domain, on which the computation can be performed more efficiently and rapidly. For example, the Earth's surface can be locally approximated as flat for simulations due to its large radius, which results in a small curvature. This approximation is valid because the local deviation from a flat plane over relatively small areas is negligible compared to the scale of the Earth. Mathematically, the curvature of a sphere with a large radius approaches zero. As a result, the error introduced by approximating the surface as flat is negligible for many practical applications, particularly when the region of interest in the simulation is small relative to the Earth's overall size.

The authors in [25] used triangular meshes for spatial discretization and a fully explicit form for calculation, which improves the efficiency of the algorithm. We compare the CPU time between the previous method and the present method. Because the previous method and present method are solved on a spherical surface and a cubic surface, respectively, we set up a similar environment for an unbiased comparison. The spherical flow is simulated with 2562 points, 5120 triangular faces, a radius of $R = 1$, $\Delta t = 0.001$, and an initial velocity field $\mathbf{u}(\mathbf{x}, 0) = (0.1y, -0.1x, 0)$. The cubic flow is simulated with 24 576 points, a $1 \times 1 \times 1$ cubic surface, a time step of $\Delta t = 0.01$, and an initial velocity field defined as $u(x, y, 0) = 0.1(1 - 2|1.5 - y|)$ for $1 \le y \le 2$, $u(x, y, 0) = 0$ otherwise, with $v(x, y, 0) = 0$.

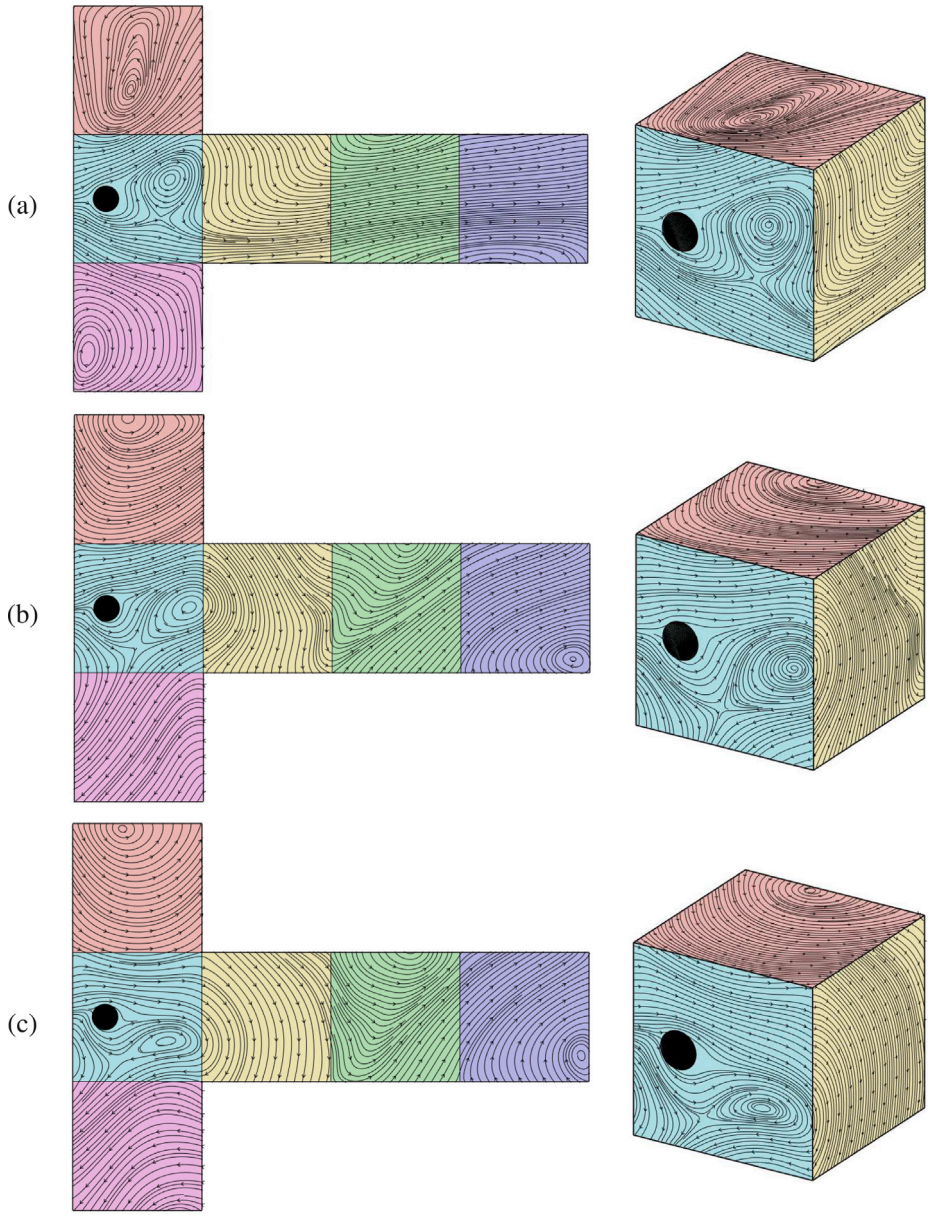


Fig. 17. Snapshots of flow past a circular cylinder at (a) $t = 9$, (b) $t = 15$, and (c) $t = 18$.

Both simulations are performed until the final time, $t = 1$. Fig. 18(a) and (b) illustrate the initial and computational results of the previous method, respectively. Fig. 18(c) and (d) display the initial and numerical results of the present method, respectively. The CPU times for the previous method and the present proposed scheme are 122.09 and 12.41, respectively. Despite the fact that the proposed method calculates 10 times more grid points compared to the previous method, its CPU time is significantly reduced due to the capability of using larger time steps and its superior computational efficiency.

We successfully conducted numerical experiments on virtual cubic surfaces, which serve as a good approximation of spherical surfaces when the surface curvature is small. Hence, there is a natural limitation when the domain size is small and the surface curvature is large, and the approximated spherical surface has a large curvature.

4. Conclusion

In conclusion, this study has successfully introduced a numerical method for simulating incompressible fluid flows on a cubic surface. By omitting the influence of gravitational force, we have achieved the elimination of its impact on the system. As a result,

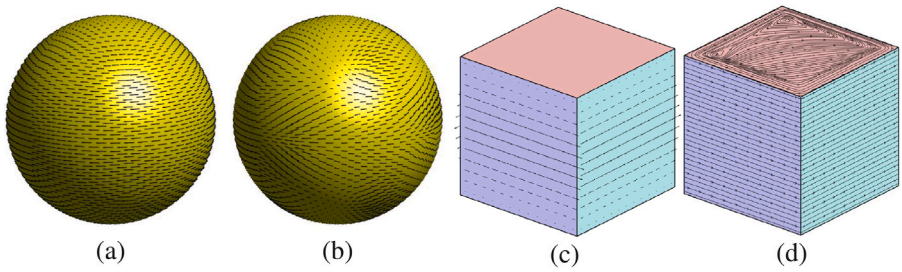


Fig. 18. Snapshot images of the simulations. (a) Initial condition of the previous method, (b) result at $t = 1$ for the previous method, (c) initial condition of the proposed method, and (d) result at $t = 1$ for the proposed method.

the dynamics of the fluid flow can be accurately characterized as primarily two-dimensional and confined to a specific plane. The combination of a projection method and an FDM has proven effective in numerically solving the governing equation that describes the fluid flow on the cubic surface. This innovative approach contributes to the field by providing a robust numerical framework for analyzing and understanding incompressible fluid flows on cubic surfaces. The findings from this study have significant implications for various applications such as engineering design, medical devices, and animation. By enabling reliable simulations, this numerical method opens doors for further advancements in these areas and contributes to a deeper understanding of fluid dynamics in complex systems. For future research, we will extend the current algorithm to investigate multiphase flows on cubic surfaces by using the Allen–Cahn equations [40].

CRediT authorship contribution statement

Junxiang Yang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Seungyoon Kang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Investigation, Formal analysis, Data curation. **Sangkwon Kim:** Writing – review & editing, Writing – original draft, Visualization, Validation, Investigation, Data curation. **Youngjin Hwang:** Writing – review & editing, Writing – original draft, Visualization, Investigation, Formal analysis, Data curation. **Soobin Kwak:** Writing – review & editing, Validation, Software, Investigation, Formal analysis, Data curation. **Seokjun Ham:** Writing – review & editing, Validation, Software, Investigation, Formal analysis, Data curation. **Junseok Kim:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

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

Acknowledgments

Junxiang Yang is supported by the Science and Technology Development Fund (FDCT) (No. FDCT-24-080-SCSE) of Macao SAR and the Macau University of Science and Technology Faculty Research Grants (FRG) (No. FRG-24-026-FIE). Sangkwon 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). We sincerely thank the reviewers for their valuable comments and suggestions, which have significantly improved the quality of this manuscript.

Data availability

No data was used for the research described in the article.

References

[1] Yang J, Li Y, Kim J. A structure-preserving projection method with formal second-order accuracy for the incompressible Navier–Stokes equations. Commun Nonlinear Sci Numer Simul 2024;133:107963. http://dx.doi.org/10.1016/j.cnsns.2024.107963.

[2] Antuono M. Tri-periodic fully three-dimensional analytic solutions for the Navier–Stokes equations. J Fluid Mech 2020;890:A23. http://dx.doi.org/10.1017/jfm.2020.126.

[3] Abramov RV. Turbulence via intermolecular potential: Uncovering the origin. Commun Nonlinear Sci Numer Simul 2024;130:107727. http://dx.doi.org/10.1016/j.cnsns.2023.107727.

[4] Yakoubi D. Enhancing the viscosity-splitting method to solve the time-dependent Navier–Stokes equations. Commun Nonlinear Sci Numer Simul 2023;123:107264. http://dx.doi.org/10.1016/j.cnsns.2023.107264.
[5] Yadav VS, Ganta N, Mahato B, Rajpoot MK, Bhumkar YG. New time-marching methods for compressible Navier–Stokes equations with applications to aeroacoustics problems. Appl Math Comput 2022;419:126863. http://dx.doi.org/10.1016/j.amc.2021.126863.
[6] Ebrahimijahan A, Dehghan M, Abbaszadeh M. Simulation of the incompressible Navier–Stokes via integrated radial basis function based on finite difference scheme. Eng Comput 2022;1–22. http://dx.doi.org/10.1007/s00366-021-01543-z.
[7] Hu T, Leng Y, Gomez H. A novel method to impose boundary conditions for higher-order partial differential equations. Comput Methods Appl Mech Engrg 2022;391:114526. http://dx.doi.org/10.1016/j.cma.2021.114526.
[8] Wassim E, Zheng B, Shang Y. A parallel two-grid method based on finite element approximations for the 2D/3D Navier–Stokes equations with damping. Eng Comput 2023;1–14. http://dx.doi.org/10.1007/s00366-023-01807-w.
[9] Costa R, Clain S, Machado GJ, Nóbrega JM. Very high-order accurate finite volume scheme for the steady-state incompressible Navier–Stokes equations with polygonal meshes on arbitrary curved boundaries. Comput Methods Appl Mech Engrg 2022;396:115064. http://dx.doi.org/10.1016/j.cma.2022.115064.
[10] Fisher N. ADI+MDA orthogonal spline collocation for the pressure Poisson reformulation of the Navier–Stokes equation in two space variables. Math Comput Simulation 2023;208:351–65. http://dx.doi.org/10.1016/j.matcom.2023.01.020.
[11] Li J, Zhu S. Shape optimization of Navier–Stokes flows by a two-grid method. Comput Methods Appl Mech Engrg 2022;400:115531. http://dx.doi.org/10.1016/j.cma.2022.115531.
[12] Zhang Y, Lin J, Reutskiy S, Rabczuk T, Lu J. A Gaussian–cubic backward substitution method for the four-order pure stream function formulation of two-dimensional incompressible viscous flows. Eng Comput 2023;1–18. http://dx.doi.org/10.1007/s00366-023-01896-7.
[13] Cho H, Park Y, Kang M. Solving incompressible Navier–Stokes equations on irregular domains and quadtrees by monolithic approach. J Comput Phys 2022;463:111304. http://dx.doi.org/10.1016/j.jcp.2022.111304.
[14] Yang X. A novel fully decoupled scheme with second-order time accuracy and unconditional energy stability for the Navier–Stokes equations coupled with mass-conserved Allen–Cahn phase-field model of two-phase incompressible flow. Internat J Numer Methods Engrg 2021;122(5):1283–306. http://dx.doi.org/10.1002/nme.6578.
[15] Anselmann M, Markus B. A geometric multigrid method for space–time finite element discretizations of the Navier–Stokes equations and its application to 3D flow simulation. ACM Trans Math Software 2023;49(1):1–25. http://dx.doi.org/10.1145/3582492.
[16] Gao Y, Li R, He X, Lin Y. A fully decoupled numerical method for Cahn–Hilliard–Navier–Stokes–Darcy equations based on auxiliary variable approaches. J Comput Appl Math 2024;436:115363. http://dx.doi.org/10.1016/j.cam.2023.115363.
[17] Liu C, Frank F, Thiele C, Alpak FO, Berg S, Chapman W, et al. An efficient numerical algorithm for solving viscosity contrast Cahn–Hilliard–Navier–Stokes system in porous media. J Comput Phys 2020;400:108948. http://dx.doi.org/10.1016/j.jcp.2019.108948.
[18] Marynets K. The modeling of the equatorial undercurrent using the Navier–Stokes equations in rotating spherical coordinates. Appl Anal 2021;100(10):2069–77. http://dx.doi.org/10.1080/00036811.2019.1673375.
[19] Weyn JA, Durran DR, Caruana R. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. J Adv Model Earth Syst 2020;12(9):e2020MS002109. http://dx.doi.org/10.1029/2020MS002109.
[20] Kang HG, Cheong HB. An efficient implementation of a high-order filter for a cubed-sphere spectral element model. J Comput Phys 2017;332:66–82. http://dx.doi.org/10.1016/j.jcp.2016.12.001.
[21] Xiao Z, Yu P, Ouyang H, Zhang J. A parallel high-order compact scheme for the pure streamfunction formulation of the 3D unsteady incompressible Navier–Stokes equation. Commun Nonlinear Sci Numer Simul 2021;95:105631. http://dx.doi.org/10.1016/j.cnsns.2020.105631.
[22] Lee D, Palha A. A mixed mimetic spectral element model of the rotating shallow water equations on the cubed sphere. J Comput Phys 2018;375:240–62. http://dx.doi.org/10.1016/j.jcp.2018.08.042.
[23] Bellet JB. A discrete Funk transform on the cubed sphere. J Comput Appl Math 2023;429:115205. http://dx.doi.org/10.1016/j.cam.2023.115205.
[24] Yang J, Li Y, Kim J. A practical finite difference scheme for the Navier–Stokes equation on curved surfaces in R3. J Comput Phys 2020;411:109403. http://dx.doi.org/10.1016/j.jcp.2020.109403.
[25] Yang J, Tan Z, Kim S, Lee C, Kwak S, Kim J. Finite volume scheme for the lattice Boltzmann method on curved surfaces in 3D. Eng Comput 2022;38(6):5507–18. http://dx.doi.org/10.1007/s00366-022-01671-0.
[26] Shashkin VV, Goyman GS, Tolstykh MA. Summation-by-parts finite-difference shallow water model on the cubed-sphere grid. Part I: Non-staggered grid. J Comput Phys 2023;474:111797. http://dx.doi.org/10.1016/j.jcp.2022.111797.
[27] Margenberg N, Hartmann D, Lessig C, Richter T. A neural network multigrid solver for the Navier–Stokes equations. J Comput Phys 2022;460:110983. http://dx.doi.org/10.1016/j.jcp.2022.110983.
[28] Landau L, Lifshitz EM. Fluid mechanics, course of theoretical physics. 2nd revised ed.. vol. 6, Pergamon Press; 1987.
[29] Hwang Y, Ham S, Lee C, Lee G, Kang S, Kim J. A simple and efficient numerical method for the Allen–Cahn equation on effective symmetric triangular meshes. Electron Res Arch 2023;31(8):4557–78. http://dx.doi.org/10.3934/era.2023233.
[30] Hwang Y, Kim I, Kwak S, Ham S, Kim S, Kim J. Unconditionally stable Monte Carlo simulation for solving the multi-dimensional Allen–Cahn equation. Electron Res Arch 2023;31(8):5104–23. http://dx.doi.org/10.3934/era.2023261.
[31] Kim J, Kwak S, Lee HG, Hwang Y, Ham S. A maximum principle of the Fourier spectral method for diffusion equations. Electron Res Arch 2023;31(9):5396–405. http://dx.doi.org/10.3934/era.2023273.
[32] Kwak S, Kang S, Ham S, Hwang Y, Lee G, Kim J. An unconditionally stable difference scheme for the two-dimensional modified Fisher–Kolmogorov–Petrovsky–Piscounov equation. J Math 2023;2023(1):5527728. http://dx.doi.org/10.1155/2023/5527728.
[33] Lee C, Kim S, Kwak S, Hwang Y, Ham S, Kang S, et al. Semi-automatic fingerprint image restoration algorithm using a partial differential equation. AIMS Math 2023;8(11):27528–41. http://dx.doi.org/10.3934/math.20231408.
[34] Deville MO, Fischer PF, Mund EH. High-order methods for incompressible fluid flow. Cambridge University Press; 2002.
[35] Tome MF, McKee S. GENSMAC: A computational marker and cell method for free surface flows in general domains. J Comput Phys 1994;110(1):171–86. http://dx.doi.org/10.1006/jcph.1994.1013.
[36] Griebel M, Dornseifer T, Neunhoeffer T. Numerical simulation in fluid dynamics: a practical introduction. Soc Ind Appl Math 1998.
[37] Leweke T, Le Dizes S, Williamson CH. Dynamics and instabilities of vortex pairs. Annu Rev Fluid Mech 2016;48:507–41. http://dx.doi.org/10.1146/annurev-fluid-122414-034558.
[38] Bergmann M, Hovnanian J, Iollo A. An accurate Cartesian method for incompressible flows with moving boundaries. Commun Comput Phys 2014;15:1266–90. http://dx.doi.org/10.4208/cicp.220313.111013a.
[39] Liu X, Dong S, Xie Z. An unconditionally energy-stable scheme for the convective heat transfer equation. Internat J Numer Methods Heat Fluid Flow 2023;33(8):2982–3024. http://dx.doi.org/10.1108/HFF-08-2022-0477.
[40] Cheng J, Xia Q, Kim J, Li Y. An efficient linear and unconditionally stable numerical scheme for the phase field sintering model. Commun Nonlinear Sci Numer Simul 2023;127:107529. http://dx.doi.org/10.1016/j.cnsns.2023.107529.