Journal of Computational Physics 401 (2020) 109019

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A high order least square-based finite difference-finite volume method with lattice Boltzmann flux solver for simulation of incompressible flows on unstructured grids

Y.Y. Liu, C. Shu *, H.W. Zhang, L.M. Yang

Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore

ARTICLE INFO

Article history:
Received 22 May 2019
Received in revised form 7 September 2019
Accepted 7 October 2019
Available online 21 October 2019

Keywords:
High order finite volume method
Least square-based finite difference
Lattice Boltzmann flux solver
Incompressible flow
Unstructured grids

ABSTRACT

This paper presents a high order least square-based finite difference-finite volume (LSFD-FV) method together with lattice Boltzmann flux solver for accurate simulation of incompressible flows on unstructured grids. Within each control cell, a high order polynomial, which is based on Taylor series expansion, is applied to approximate the solution function. The derivatives in the Taylor series expansion are approximated by the functional values at the centers of neighboring cells using the mesh-free least square-based finite difference (LSFD) method developed by Ding et al. (2004) [36]. In the high order finite volume method, the recently developed lattice Boltzmann flux solver (LBFS) is applied to evaluate the inviscid and viscous fluxes physically and simultaneously at the cell interface by local reconstruction of lattice Boltzmann solution. Compared with traditional *k*-exact high-order finite volume method, the present method is more accurate and efficient. Various benchmark examples are tested to validate the high-order accuracy, high computational efficiency and flexibility of the proposed method on unstructured grids.

1. Introduction

Development of high-order discretization methods on unstructured grids is becoming fashionable in computational fluid dynamics (CFD) due to their higher accuracy than low-order methods. Although the second-order schemes have played an important role in engineering applications due to their robust and reliable characteristics, high-order schemes outperform them on prediction of turbulence, aeroacoustics and many viscosity-dominated flows. In the past two decades, various high-order schemes [1–29] have been developed in the CFD community for simulation of incompressible and compressible flows. There are several differences and similarities among them. Firstly, the difference between high-order methods for compressible and incompressible flows is inherited from the physics of these two types of flows. For incompressible flows, the continuity equation contains only velocity components and there is no explicit link with the pressure. Thus, particular techniques, such as vorticity-stream function, artificial compressibility and pressure Poisson equation, have to be adopted to overcome the restriction. Then the high-order methods, such as discontinuous Galerkin (DG) methods [20–22], spectral methods [23] and finite difference (FD) methods [7,56], are used to discretize these particular equations. However, for compressible flows, the continuity and momentum equations are coupled. Therefore, the difficulty of the pressure-velocity coupling for incompressible flows does not exist. The high-order methods can discretize the compressible Navier-Stokes

* Corresponding author.
E-mail address: mpeshuc@nus.edu.sg (C. Shu).

https://doi.org/10.1016/j.jcp.2019.109019
0021-9991/

(N-S) equations directly. Secondly, for compressible flows, there are discontinuities in the flow field generally and the shock capturing techniques (e.g., the multi-dimensional limiter [8]) are usually required for the high-order methods to obtain the reasonable results. The incompressible flows, however, possess the smoother flow field where the shock capturing methods are not required basically. With regard to the similarities, since the fundamental of the high-order method is independent of the specific problems, the algorithm and formulation almost keep the same during the application of one high-order method for simulation of incompressible flows or compressible flows. In this work, we focus on the development of a new high-order method for incompressible flows.

Among the various high-order methods, the control volume-based high order schemes receive more and more attention due to their natural connection with physical conservation laws. In this category, the physical conservation laws (governing differential equation written in conservative form) are applied to a control cell. The conventional finite volume method (FVM) applies mean theorem to approximate the volume integral and surface integrals for each control cell. This treatment has the second-order of accuracy and the position of mean value is usually at the cell center. Thus, each cell has one unknown (functional value at cell center) and one equation (discrete form of governing equation by FVM), and the problem is well-posed. When a high order finite volume (FV) method is developed, the high order polynomial approximation for the solution function within each control cell is necessary. However, this may cause the problem to be not well-posed. For example, when the solution function within the control cell is approximated by a high order Taylor series expansion, the resultant integration of solution function over the control cell will give functional value and its spatial derivatives as unknowns but there is only one discrete equation for each cell. Obviously, the number of unknowns is larger than the number of equations, so the problem is not well-posed. To resolve this difficulty, various high order methods were presented. The widely used approach is the DG method [18–22,30–32]. To close the system, the DG method increases the number of equations. This can be achieved by weighted integrations of governing equation over each control cell. In the process, the number of weighting functions is taken the same as the number of unknowns. This can guarantee that the number of equations given from the integrations equals the number of unknowns. As compared with conventional FVM, the DG method needs to solve more discrete equations, and numerical stability may be affected. The major advantage of the DG method is that it is as compact as the conventional FVM since its discrete form of governing equations only involves information of current cell. Different from the DG method, the spectral volume method [25–27] uses the current cell and its neighboring cells to form a big cell, within which a high order polynomial approximation is established by using functional values at small cells. This polynomial is in turn applied to the current small cell for discretization of governing equations. Like the conventional FVM, the spectral volume method only uses one discrete equation for each control cell. Thus, its computational effort is lower than the DG method. However, the time-dependent term of its discrete form would involve functional values at neighboring cell centers, which is not as compact as the DG method. In addition, it is sensitive to the configuration of forming a big cell from the small neighboring cells. If it is not properly done, numerical computation may encounter instability [33–35]. This may cause inconvenience for numerical simulation, especially for the three-dimensional case. Apart from the DG and spectral volume methods, there is another high order method, called k-exact method [5]. In this method, the mean theorem is still applied in the finite volume discretization of governing equations. However, the mean value of the solution function may not be at the cell center, which is generally unknown. To keep high order approximation for the solution function within each cell, a modified Taylor series expansion is adopted, which is determined by the mean values of solution function at neighboring cells and used to interpolate the functional value at the cell interface. In the process, the modified Taylor series expansion needs to be integrated over the current cell and its neighboring cells respectively, resulting in an equation system to compute spatial derivatives at the cell center. Like the spectral volume method, the k-exact method also uses one discrete equation for each control cell, and its computational effort is less than that of the DG method. As the least square optimization is used for polynomial approximation, the k-exact method is usually more stable than the spectral volume method. However, its solution procedure seems to be more complicated as it needs to integrate the modified Taylor series expansion over the current and neighboring cells.

In this paper, we aim to present a more straightforward high order FV method. Like other high order methods, within each control cell, the solution function is approximated by a high order Taylor series expansion, which is then substituted into the discrete form of governing equations given by FVM. Since the Taylor series expansion involves the functional value and its spatial derivatives at the cell center, the discrete form of governing equations for each control cell would involve more than one unknowns (solution function and its spatial derivatives). To close the system, the least square-based finite difference (LSFD) scheme is used to approximate all the spatial derivatives by the functional values at centers of current cell and its neighboring cells. LSFD was proposed by Shu and his colleagues [36], which is a mesh-free method to approximate derivatives at randomly distributed points. It is an ideal approach to approximate derivatives on unstructured meshes. LSFD has been successfully applied to simulate various flow problems [37–40] with high-order of accuracy. However, due to the nature of finite difference discretization, there is no guarantee for its numerical discretization to be conservative if it is applied alone. In this work, FV method can guarantee conservative discretization and LSFD is a mesh-free method for approximation of derivatives. Thus, it is believed that the combination of LSFD and FV method can provide a simple and effective high order solver on unstructured mesh. On the other hand, similar to the spectral volume and k-exact methods, there is only one equation for each control cell in the present work, but the trade-off is that the time-dependent term resultant from volume integral of solution function over the control cell involves a pre-multiplied coefficient matrix. As shown in the paper, the resultant equation system can be solved by a point iterative method.

Apart from solution approximation within each cell, the discrete form of governing equations also involves evaluation of numerical fluxes at cell interface. Conventional methods usually evaluate inviscid flux and viscous flux separately by using smooth function interpolation or difference approximation, which is a mathematical way. For high order methods, the evaluation of viscous flux becomes more critical. For example, the DG method [41–43] needs to introduce additional degree of freedoms for evaluation of viscous flux. The process may increase complexity and computational effort. As compared to the mathematical approach, physical evaluation of numerical fluxes at the cell interface is more desirable as this way can assure that the solution at the cell interface also satisfies the governing equation. Lattice Boltzmann flux solver (LBFS) [44] is such a physical way. It uses local solution of lattice Boltzmann equation (LBE) to compute inviscid and viscous fluxes at the cell interface simultaneously. Through Chapman-Enskog (C-E) expansion analysis [44,45], it can be shown that the LBE solution can satisfy N-S equations. This means that the flow variables and fluxes at the cell interface given from LBFS also satisfy N-S equations. Thus, the mass and momentum conservation can be guaranteed effectively. So far, LBFS has been extensively applied to simulate various incompressible flows [46–50] with the weakly compressible assumption in the limit of low Mach number. In this work, LBFS is adopted to evaluate numerical fluxes at the cell interface in the high order LSFD-FV solver. Since inviscid and viscous fluxes are evaluated physically at the same time, the conventional difficulty of evaluating viscous flux in other high order methods is not appeared in the present work. To validate the present high order solver, several benchmark problems are solved. Numerical results demonstrate that the present high order method is slightly more accurate and efficient than conventional *k*-exact method.

2. High order least square-based finite difference-finite volume (LSFD-FV) method

2.1. Governing equations and high order finite volume discretization

This subsection describes the governing equations and the general framework for the cell-centered FVM on arbitrary grids. In Cartesian coordinate system, the governing equations for two-dimensional incompressible flows in the low Mach number limit, in the absence of source terms, can be cast as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{1}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u} + p\boldsymbol{I}) = \nu \nabla \cdot \left[\nabla \rho \boldsymbol{u} + (\nabla \rho \boldsymbol{u})^T\right], \tag{2}$$

where ρ, p and ν are respectively the density, pressure and kinematic viscosity of fluid flow. $\boldsymbol{u} = (u, v)$ is the velocity vector expressed in the global Cartesian coordinate system. $\boldsymbol{I}$ is the unit tensor. Eqs. (1) and (2) can be written in a unified way as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F} = 0, \tag{3}$$

where the vector of conservative variables $\boldsymbol{U}$ and the vector of flux $\boldsymbol{F}$ are given by

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho \boldsymbol{u} \end{bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} F_\rho \\ F_{\rho \boldsymbol{u}} \end{bmatrix}. \tag{4}$$

In this work, the cell-centered FVM is adopted to discretize the governing equations. The basic discretization stems from the integral form of Eq. (3) over a control cell Ω_i with application of the divergence theorem. Then Eq. (3) can be written as

$$\frac{\partial}{\partial t} \int_{\Omega_i} \boldsymbol{U} d\Omega = - \int_{\Gamma_i} \boldsymbol{F} \cdot \boldsymbol{n} d\Gamma, \tag{5}$$

where Γ_i denotes the boundary of the control cell Ω_i and $\boldsymbol{n} = (n_x, n_y)$ represents the unit normal vector of the cell interface in the global Cartesian coordinate system.

In the conventional FVM, the volume integral and the surface integral in Eq. (5) are approximated by the mean theorem, and the mean value is defined at the cell/surface center. This approximation has the second-order of accuracy. To achieve a high-order of accuracy for Eq. (5), the solution variable $\boldsymbol{U}$ should be approximated by a high order polynomial, which is also used to interpolate functional value at the surface center for evaluation of numerical fluxes. As an example, it is supposed that the solution variable $\boldsymbol{U}$ is approximated by the following polynomial given from Taylor series expansion,

$$\begin{aligned} \boldsymbol{U}(x, y) = \boldsymbol{U}_i &+ \left.\frac{\partial \boldsymbol{U}}{\partial x}\right|_i (x - x_i) + \left.\frac{\partial \boldsymbol{U}}{\partial y}\right|_i (y - y_i) + \left.\frac{\partial^2 \boldsymbol{U}}{\partial x^2}\right|_i \frac{(x - x_i)^2}{2} \\ &+ \left.\frac{\partial^2 \boldsymbol{U}}{\partial y^2}\right|_i \frac{(y - y_i)^2}{2} + \left.\frac{\partial^2 \boldsymbol{U}}{\partial x \partial y}\right|_i (x - x_i)(y - y_i) + \left.\frac{\partial^3 \boldsymbol{U}}{\partial x^3}\right|_i \frac{(x - x_i)^3}{6} \\ &+ \left.\frac{\partial^3 \boldsymbol{U}}{\partial y^3}\right|_i \frac{(y - y_i)^3}{6} + \left.\frac{\partial^3 \boldsymbol{U}}{\partial x^2 \partial y}\right|_i \frac{(x - x_i)^2 (y - y_i)}{2} + \left.\frac{\partial^3 \boldsymbol{U}}{\partial y^2 \partial x}\right|_i \frac{(y - y_i)^2 (x - x_i)}{2}, \end{aligned} \tag{6}$$

where the reference point (x_i, y_i) is the cell centroid of Ω_i. Note that from Taylor series expansion, the truncation error of Eq. (6) is in the order of $O(\Delta x^4, \Delta y^4)$. When Eq. (6) is substituted into Eq. (5), it involves 10 unknowns at the cell center (1 functional value, 2 first order derivatives, 3 second order derivatives and 4 third order derivatives). In fact, integrating Eq. (6) over the control cell Ω_i gives the following equation,

$$\int_{\Omega_i} \boldsymbol{U}(x, y)d\Omega = \Omega_i \boldsymbol{U}_i + d\boldsymbol{U}_i^T C_i, \tag{7}$$

with

$$\begin{aligned} d\boldsymbol{U}_i^T &= \left[\left.\frac{\partial \boldsymbol{U}}{\partial x}\right|_i, \left.\frac{\partial \boldsymbol{U}}{\partial y}\right|_i, \left.\frac{\partial^2 \boldsymbol{U}}{\partial x^2}\right|_i, \left.\frac{\partial^2 \boldsymbol{U}}{\partial y^2}\right|_i, \left.\frac{\partial^2 \boldsymbol{U}}{\partial x \partial y}\right|_i, \left.\frac{\partial^3 \boldsymbol{U}}{\partial x^3}\right|_i, \left.\frac{\partial^3 \boldsymbol{U}}{\partial y^3}\right|_i, \left.\frac{\partial^3 \boldsymbol{U}}{\partial x^2 \partial y}\right|_i, \left.\frac{\partial^3 \boldsymbol{U}}{\partial y^2 \partial x}\right|_i \right], \\ C_i^T &= \left[\overline{\overline{x^1 y^0}}_i, \overline{\overline{x^0 y^1}}_i, \frac{\overline{\overline{x^2 y^0}}_i}{2}, \frac{\overline{\overline{x^0 y^2}}_i}{2}, \overline{\overline{x^1 y^1}}_i, \frac{\overline{\overline{x^3 y^0}}_i}{6}, \frac{\overline{\overline{x^0 y^3}}_i}{6}, \frac{\overline{\overline{x^2 y^1}}_i}{2}, \frac{\overline{\overline{x^1 y^2}}_i}{2} \right], \end{aligned} \tag{8}$$

where $\overline{\overline{x^n y^m}}_i = \int_{\Omega_i} (x - x_i)^n (y - y_i)^m d\Omega$. Equation (7) has 10 unknowns but Eq. (5) only provides one equation for each cell. Obviously, the problem is not well-posed. To close the system, the DG method modifies Eq. (5) by multiplying a weighting function to Eq. (3) and then performing integration over the control cell. The number of weighting functions is chosen to be the same as the number of unknowns (10 in Eq. (6)). Therefore, 10 equations for the control cell will be given. In this work, the unknown derivatives in Eq. (6) will be approximated by the meshless LSFD method, which will be described in Section 2.2. Apart from the volume integral, the surface integral in Eq. (5) also needs high order approximation. In this work, Gaussian quadrature is applied to approximate the surface integral. Generally, two Gaussian quadrature points will provide the fourth-order of accuracy. By applying the Gaussian quadrature rule, the surface integral in Eq. (5) can be approximated by

$$\int_{\Gamma_i} \boldsymbol{F} \cdot \boldsymbol{n} d\Gamma = \sum_{edge=1}^{nedge} \sum_{GQp=1}^{nGQp} (\boldsymbol{F}_{edge,GQp} \cdot \boldsymbol{n}_{edge}) A_{edge} w_{GQp}, \tag{9}$$

where *nedge* is the number of control surfaces for the control cell Ω_i, $nGQp$ represents the number of Gaussian quadrature points on each control surface, w_{GQp} denotes a quadrature weight and A is the surface area. The evaluation of numerical flux $\boldsymbol{F}$ in Eq. (9) will be addressed in Section 2.3. With Eqs. (7) and (9), Eq. (5) can be reduced to a set of ordinary differential equations. The solution of these equations will be discussed in Section 2.4.

2.2. Derivative approximation by least square-based finite difference (LSFD) scheme

In this work, the least square-based finite difference (LSFD) scheme [36,37] is applied to approximate spatial derivatives in Eq. (6). In fact, LSFD is also based on two-dimensional Taylor series expansion as shown in Eq. (6), where derivatives are considered as unknowns. Therefore, the approximation form has 9 unknowns. By applying Eq. (6) at 9 neighboring points, the following equation system can be obtained:

$$\boldsymbol{S} d\boldsymbol{U} = \Delta \boldsymbol{U}, \tag{10}$$

where the matrix $\boldsymbol{S}$ contains all the geometric information about the distribution of the supporting points. The details of the matrix $\boldsymbol{S}$ and $\Delta \boldsymbol{U}$ are

$$\begin{aligned} \boldsymbol{S} &= \begin{pmatrix} \Delta x_1 & \Delta y_1 & \frac{\Delta x_1^2}{2} & \frac{\Delta y_1^2}{2} & \Delta x_1 \Delta y_1 & \frac{\Delta x_1^3}{6} & \frac{\Delta y_1^3}{6} & \frac{\Delta x_1^2 \Delta y_1}{2} & \frac{\Delta y_1^2 \Delta x_1}{2} \\ \Delta x_2 & \Delta y_2 & \frac{\Delta x_2^2}{2} & \frac{\Delta y_2^2}{2} & \Delta x_2 \Delta y_2 & \frac{\Delta x_2^3}{6} & \frac{\Delta y_2^3}{6} & \frac{\Delta x_2^2 \Delta y_2}{2} & \frac{\Delta y_2^2 \Delta x_2}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta x_9 & \Delta y_9 & \frac{\Delta x_9^2}{2} & \frac{\Delta y_9^2}{2} & \Delta x_9 \Delta y_9 & \frac{\Delta x_9^3}{6} & \frac{\Delta y_9^3}{6} & \frac{\Delta x_9^2 \Delta y_9}{2} & \frac{\Delta y_9^2 \Delta x_9}{2} \end{pmatrix}, \\ \Delta \boldsymbol{U}^T &= [\boldsymbol{U}_{1i} - \boldsymbol{U}_i, \boldsymbol{U}_{2i} - \boldsymbol{U}_i, \ldots, \boldsymbol{U}_{9i} - \boldsymbol{U}_i], \end{aligned} \tag{11}$$

where $(\Delta x, \Delta y) = (x - x_i, y - y_i)$, $\boldsymbol{U}_{ki}$ is the functional value at the center of the kth neighboring cell to the current cell i. By solving equation system (10), the expression of the derivative vector $d\boldsymbol{U}$ can be obtained in terms of $\Delta \boldsymbol{U}$. However, due to unstructured cell distribution for a general case, the equation system may be ill-conditioned or even singular. To overcome this difficulty, the local scaling technique and least square optimization were introduced, where Eq. (6) is applied at more than 9 neighboring points. The details of these techniques can be referred to the work of Ding et al. [36]. As a consequence, the derivative vector $d\boldsymbol{U}$ can be approximated by the following matrix form

$$d\boldsymbol{U} = \boldsymbol{W}\Delta\boldsymbol{U}, \tag{12}$$

where $\boldsymbol{W}$ is the $9 \times N$ dimensional weighting coefficient matrix, whose components are uniquely determined by the coordinates of mesh points (centers of current cell and its neighboring cells), and N is the number of neighboring cells. The details of matrix $\boldsymbol{W}$ can be referred to the appendix. With matrix $\boldsymbol{W}$, the first, second and third order derivatives at the center of current cell i can be approximated by the following forms,

$$\left.\frac{\partial \boldsymbol{U}}{\partial x}\right|_i = \sum_{j=1}^{N} W_{1,j}\Delta\boldsymbol{U}_{ij}, \tag{13a}$$

$$\left.\frac{\partial \boldsymbol{U}}{\partial y}\right|_i = \sum_{j=1}^{N} W_{2,j}\Delta\boldsymbol{U}_{ij}, \tag{13b}$$

$$\left.\frac{\partial^2 \boldsymbol{U}}{\partial x^2}\right|_i = \sum_{j=1}^{N} W_{3,j}\Delta\boldsymbol{U}_{ij}, \tag{13c}$$

$$\left.\frac{\partial^2 \boldsymbol{U}}{\partial y^2}\right|_i = \sum_{j=1}^{N} W_{4,j}\Delta\boldsymbol{U}_{ij}, \tag{13d}$$

$$\left.\frac{\partial^2 \boldsymbol{U}}{\partial x\partial y}\right|_i = \sum_{j=1}^{N} W_{5,j}\Delta\boldsymbol{U}_{ij}, \tag{13e}$$

$$\left.\frac{\partial^3 \boldsymbol{U}}{\partial x^3}\right|_i = \sum_{j=1}^{N} W_{6,j}\Delta\boldsymbol{U}_{ij}, \tag{13f}$$

$$\left.\frac{\partial^3 \boldsymbol{U}}{\partial y^3}\right|_i = \sum_{j=1}^{N} W_{7,j}\Delta\boldsymbol{U}_{ij}, \tag{13g}$$

$$\left.\frac{\partial^3 \boldsymbol{U}}{\partial x^2\partial y}\right|_i = \sum_{j=1}^{N} W_{8,j}\Delta\boldsymbol{U}_{ij}, \tag{13h}$$

$$\left.\frac{\partial^3 \boldsymbol{U}}{\partial x\partial y^2}\right|_i = \sum_{j=1}^{N} W_{9,j}\Delta\boldsymbol{U}_{ij}, \tag{13i}$$

where j is the jth neighboring cell of the current cell i, ij is the global index of jth neighboring cell. As indicated above, the weighting coefficient matrix $\boldsymbol{W}$ is uniquely determined by the mesh point distribution. So, when a stationary problem is considered, the computational mesh is fixed. Therefore, we only need to compute $\boldsymbol{W}$ once and store the data for the following computation. This can save the computational effort.

2.3. Evaluation of numerical fluxes at cell interface by lattice Boltzmann flux solver

The above subsection discusses how to use LSFD method to approximate derivatives in the high order Taylor series expansion for the solution function in the FV discretization. This subsection will address another important issue, that is, evaluation of numerical flux at the cell interface. As indicated in the introduction, the lattice Boltzmann flux solver (LBFS) [44] is adopted in this work to fulfill the job. The LBFS is proposed recently for simulation of incompressible inviscid and viscous flows, which evaluates the inviscid and viscous fluxes simultaneously by local reconstruction of lattice Boltzmann solution from macroscopic flow variables at two cell centers. Clearly, it is a physical reconstruction of solution at the cell interface. In this work, only 2D cases are considered. In order to facilitate simulations of flows on arbitrary grids, a local-coordinate system is introduced at the cell interface, in which the two directions are the outward normal direction and the tangential direction. Some details of LBFS in the local-coordinate system for the 2D case are shown below.

Through multiscale Chapman-Enskog expansion analysis [45], Eqs. (1) and (2) can be recovered by expressing the fluxes as follows

$$F_\rho = \sum_{\alpha=0}^{Nd} f_\alpha^{eq}(\boldsymbol{e}_\alpha)_1, \tag{14}$$

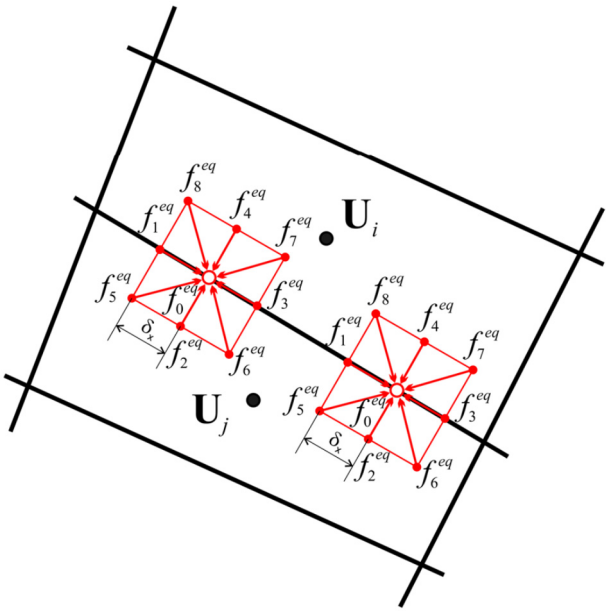


Fig. 1. Local reconstruction of lattice Boltzmann solution at cell interface. δ_x denotes the streaming distance. f_α^{eq} denotes the equilibrium distribution function at the α-th lattice point.

$$F_{\rho u_1} = \sum_{\alpha=0}^{Nd} (\boldsymbol{e}_\alpha)_1 (\boldsymbol{e}_\alpha)_1 \left[f_\alpha^{eq} + \left(1 - \frac{1}{2\tau}\right) f_\alpha^{neq} \right], \tag{15a}$$

$$F_{\rho u_2} = \sum_{\alpha=0}^{Nd} (\boldsymbol{e}_\alpha)_1 (\boldsymbol{e}_\alpha)_2 \left[f_\alpha^{eq} + \left(1 - \frac{1}{2\tau}\right) f_\alpha^{neq} \right], \tag{15b}$$

where τ is the single relaxation parameter and $\boldsymbol{e}_\alpha$ is the lattice velocity vector. The subscripts "1" and "2" denote the outward normal and tangential direction of the cell interface, respectively. Nd is the number of discrete particle velocities in the lattice Boltzmann method. For the most popular D2Q9 model, $Nd = 9$. f_α^{eq} denotes the equilibrium distribution function along the α direction and f_α^{neq} is the corresponding non-equilibrium distribution function.

The equilibrium distribution function f_α^{eq} reads

$$f_\alpha^{eq} = \rho w_\alpha \left[1 + \frac{\boldsymbol{e}_\alpha \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})^2 - (c_s|\boldsymbol{u}|)^2}{2c_s^4} \right]. \tag{16}$$

For D2Q9 model defined in a square lattice as shown in Fig. 1, the sound speed c_s and the coefficients w_α are given as $c_s = c/\sqrt{3}$, $w_0 = 4/9$, $w_1 = w_2 = w_3 = w_4 = 1/9$ and $w_5 = w_6 = w_7 = w_8 = 1/36$. $c = \delta_x/\delta_t$. δ_x and δ_t denote the lattice spacing and the streaming time step, respectively. Generally, c is taken as 1.

The kinematic viscosity ν can be estimated from the relaxation parameter τ with

$$\nu = \left(\tau - \frac{1}{2}\right) c_s^2 \delta_t. \tag{17}$$

The pressure can be calculated from the equation of state,

$$p = \rho c_s^2. \tag{18}$$

The non-equilibrium distribution function f_α^{neq} in Eq. (15) can be approximated by

$$f_\alpha^{neq}(\boldsymbol{r}, t) = -\tau \left[f_\alpha^{eq}(\boldsymbol{r}, t) - f_\alpha^{eq}(\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t, t - \delta_t) \right] + O\left(\delta_t^2\right), \tag{19}$$

where $\boldsymbol{r}$ is physical location and t is time. $f_\alpha^{eq}(\boldsymbol{r}, t)$ and $f_\alpha^{eq}(\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t, t - \delta_t)$ are the equilibrium distribution functions at the Gaussian quadrature point $\boldsymbol{r}$ along the cell interface and its surrounding nodes $\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t$, respectively. Note that the streaming time step δ_t equals the lattice spacing δ_x, since $c = 1$. It is indicated that LBFS has the second-order of accuracy, but this order of accuracy is in terms of lattice spacing δ_x rather than the mesh spacing Δx. In practical implementation, δ_x is chosen to be much smaller than Δx in order to keep the global accuracy of the solution.

In order to calculate $f_\alpha^{neq}(\boldsymbol{r},t)$, we have to obtain $f_\alpha^{eq}(\boldsymbol{r},t)$ and $f_\alpha^{eq}(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t,t-\delta_t)$ firstly. For $f_\alpha^{eq}(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t,t-\delta_t)$, the corresponding fluid density ρ and velocity $\boldsymbol{u}$ can be given from those at cell centroid via the high order polynomial approximation form (6). Using the information at cell centroids $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ and according to Eq. (6), any variable $\boldsymbol{\varphi}$ at the location $(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t)$ can be given as

$$\boldsymbol{\varphi}(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t)=\begin{cases}\boldsymbol{\varphi}(\boldsymbol{r}_i)+\nabla\boldsymbol{\varphi}(\boldsymbol{r}_i)\Delta\boldsymbol{X}_i+\frac{1}{2}\Delta\boldsymbol{X}_i^T\boldsymbol{H}(\boldsymbol{X}_i)\Delta\boldsymbol{X}_i & (\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t)\in\Omega_i\\ \quad+\frac{1}{6}\Delta^2\boldsymbol{X}_i^T G(\boldsymbol{X}_i)\Delta\boldsymbol{X}_i+O(\boldsymbol{X}^4) & \\ \boldsymbol{\varphi}(\boldsymbol{r}_j)+\nabla\boldsymbol{\varphi}(\boldsymbol{r}_j)\Delta\boldsymbol{X}_j+\frac{1}{2}\Delta\boldsymbol{X}_j^T\boldsymbol{H}(\boldsymbol{X}_j)\Delta\boldsymbol{X}_j & (\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t)\in\Omega_j\\ \quad+\frac{1}{6}\Delta^2\boldsymbol{X}_j^T G(\boldsymbol{X}_j)\Delta\boldsymbol{X}_j+O(\boldsymbol{X}^4) & \end{cases} \tag{20}$$

where $\boldsymbol{\varphi}=\begin{pmatrix}\rho\\ \boldsymbol{u}\end{pmatrix}$, $\Delta\boldsymbol{X}_i=\begin{pmatrix}(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t-\boldsymbol{r}_i)_x\\ (\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t-\boldsymbol{r}_i)_y\end{pmatrix}$,

$\Delta^2\boldsymbol{X}_i^T=\left((\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t-\boldsymbol{r}_i)_x^2,(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t-\boldsymbol{r}_i)_y^2\right)$.

The gradient $\nabla\boldsymbol{\varphi}$, the matrices $\boldsymbol{H}$ and $\boldsymbol{G}$ are

$$\nabla\boldsymbol{\varphi}=\left(\frac{\partial\boldsymbol{\varphi}}{\partial x},\frac{\partial\boldsymbol{\varphi}}{\partial y}\right),$$
$$\boldsymbol{H}=\begin{bmatrix}\frac{\partial^2\boldsymbol{\varphi}}{\partial x^2} & \frac{\partial^2\boldsymbol{\varphi}}{\partial x\partial y}\\ \frac{\partial^2\boldsymbol{\varphi}}{\partial y\partial x} & \frac{\partial^2\boldsymbol{\varphi}}{\partial y^2}\end{bmatrix},\qquad \boldsymbol{G}=\begin{bmatrix}\frac{\partial^3\boldsymbol{\varphi}}{\partial x^3} & 3\frac{\partial^3\boldsymbol{\varphi}}{\partial x^2\partial y}\\ 3\frac{\partial^3\boldsymbol{\varphi}}{\partial y^2\partial x} & \frac{\partial^3\boldsymbol{\varphi}}{\partial y^3}\end{bmatrix}, \tag{21}$$

where the derivatives are calculated by the LSFD method. Once the required fluid density ρ and velocity $\boldsymbol{u}$ have been obtained, $f_\alpha^{eq}(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t,t-\delta_t)$ can be calculated from Eq. (16). In addition, following the derivations in [44], we can obtain

$$\rho(\boldsymbol{r},t)=\sum_{\alpha=0}^{Nd}f_\alpha^{eq}(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t,t-\delta_t), \tag{22}$$

$$\rho(\boldsymbol{r},t)\boldsymbol{u}(\boldsymbol{r},t)=\sum_{\alpha=0}^{Nd}f_\alpha^{eq}(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t,t-\delta_t)\boldsymbol{e}_\alpha. \tag{23}$$

Thus, $f_\alpha^{eq}(\boldsymbol{r},t)$ can be further calculated with Eq. (16). Once $f_\alpha^{eq}(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t,t-\delta_t)$ and $f_\alpha^{eq}(\boldsymbol{r},t)$ are available, f_α^{neq} can be calculated with Eq. (19). Finally, F_ρ in Eq. (14) and $F_{\rho u_1}$ and $F_{\rho u_2}$ in Eq. (15) can be computed.

Practically, the fluxes in Eqs. (14) and (15) cannot be used in Eq. (3) directly since they are defined in the local coordinate system. In order to calculate fluxes $\boldsymbol{F}$ in the global Cartesian coordinate system, the coordinate transformation should be conducted. Then $\boldsymbol{F}$ in the global Cartesian coordinate system can be evaluated from the local coordinate system via

$$\boldsymbol{F}=(F_\rho,F_{\rho u_1}n_x-F_{\rho u_2}n_y,F_{\rho u_2}n_x+F_{\rho u_1}n_y)^T. \tag{24}$$

It is noted that, since LBE solution is reconstructed physically and locally at every Gaussian quadrature point along each cell interface, different δ_t could be chosen for different interfaces. This provides a great flexibility for applications on unstructured grids. Moreover, the time marching step Δt is independent of the streaming time step δ_t (δ_t is only used in the solution reconstruction), which gives flexibility for the choice of time evolution scheme in the present solver.

2.4. Solution of resultant ordinary differential equations

As shown in Eq. (7), when Taylor series expansion form (6) is substituted into the volume integral of solution function over the control cell i, we have

$$\int_{\Omega_i}\boldsymbol{U}(x,y)d\Omega=\Omega_i\boldsymbol{U}_i+d\boldsymbol{U}_i^TC_i=\Omega_i\boldsymbol{U}_i+\sum_{k=1}^{9}C_k d\boldsymbol{U}_k. \tag{25}$$

Eq. (25) involves functional value $\boldsymbol{U}_i$ and its spatial derivatives. As shown in Eq. (13), the spatial derivatives can be approximated by LSFD in terms of differences between the functional values at the centers of current cell i and its neighboring cells. Thus, we have

$$\int_{\Omega_i}\boldsymbol{U}(x,y)d\Omega=\Omega_i\boldsymbol{U}_i+\sum_{k=1}^{9}C_k\sum_{j=1}^{N}W_{k,j}\Delta\boldsymbol{U}_{ij}=\Omega_i\boldsymbol{U}_i+\sum_{k=1}^{9}C_k\sum_{j=1}^{N}W_{k,j}(\boldsymbol{U}_{ij}-\boldsymbol{U}_i). \tag{26}$$

Eq. (26) is applied to each control cell. Note that for different cells, the coefficients C_k and $W_{k,j}$ could be different. To show the difference, we use C_k^i and $W_{k,j}^i$ to replace C_k and $W_{k,j}$ in Eq. (26) in the following derivations. With new notations, Eq. (26) can be written as

$$\int_{\Omega_i} \boldsymbol{U}(x,y)d\Omega = \left(\Omega_i - \sum_{k=1}^{9} C_k^i \sum_{j=1}^{N} W_{k,j}^i\right)\boldsymbol{U}_i + \sum_{j=1}^{N}\left(\sum_{k=1}^{9} C_k^i W_{k,j}^i\right)\boldsymbol{U}_{ij}. \tag{27}$$

If we use variable $\boldsymbol{R}_i$ to represent the flux contributions as given in Eq. (9), then Eq. (5) can be written as

$$\left(\Omega_i - \sum_{k=1}^{9} C_k^i \sum_{j=1}^{N} W_{k,j}^i\right)\frac{\partial \boldsymbol{U}_i}{\partial t} + \sum_{j=1}^{N}\left(\sum_{k=1}^{9} C_k^i W_{k,j}^i\right)\frac{\partial \boldsymbol{U}_{ij}}{\partial t} = -\boldsymbol{R}_i. \tag{28}$$

As compared to the conventional FVM form, Eq. (28) involves not only the time derivative of functional value at the current cell i but also the time derivatives of functional value at the neighboring cells. To simplify the solution process, in this work, we will use point iterative method to solve Eq. (28). For simplicity, when Eq. (28) is applied to all control cells, we can have the following matrix form,

$$\boldsymbol{M}\frac{\partial \boldsymbol{U}}{\partial t} = -\boldsymbol{R}, \tag{29}$$

where $\boldsymbol{U}$, $\boldsymbol{R}$ is respectively the solution vector and the vector of flux contribution for all control cells, and $\boldsymbol{M}$ is a sparse matrix formed by coefficients on the left hand side of Eq. (28). Although the elements of $\boldsymbol{M}$ depend on the ordering of the vector $\boldsymbol{U}$, and it is not easy to write out all elements of $\boldsymbol{M}$, the diagonal elements of $\boldsymbol{M}$ can be easily given from Eq. (28) as

$$M_{i,i} = \Omega_i - \sum_{k=1}^{9} C_k^i \sum_{j=1}^{N} W_{k,j}^i. \tag{30}$$

Eq. (30) will be very useful for the point iterative solution of Eq. (29) when a steady state problem is considered. For this case, the time derivative is zero, and Eq. (29) is used to march in time until a steady state solution is reached. Since the accuracy of temporal discretization is not important for the steady problem, we can use the first order Euler explicit scheme to solve Eq. (29), which gives

$$\boldsymbol{M}\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t} = -\boldsymbol{R}^n. \tag{31}$$

For the point iterative method to solve Eq. (31), only at the current cell, $\boldsymbol{U}_i^{n+1}$ is written as

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n + \Delta \boldsymbol{U}_i^{n+1}. \tag{32}$$

For all the neighboring cells, $\boldsymbol{U}_{ij}^{n+1}$ is approximated by $\boldsymbol{U}_{ij}^n$. As a consequence, Eq. (31) can be simplified to

$$\boldsymbol{M}_{i,i}\frac{\Delta \boldsymbol{U}_i^{n+1}}{\Delta t} = -\boldsymbol{R}_i^n, \tag{33}$$

where $\boldsymbol{M}_{i,i}$ is given by Eq. (30). It can be seen from Eq. (33) that the solution for $\Delta \boldsymbol{U}$ does not need to form matrix $\boldsymbol{M}$. This process is as convenient as the explicit method utilized in the conventional FVM. Once $\Delta \boldsymbol{U}$ is obtained for all the control cells, the solution at new time level $(n+1)$ can be updated by using Eq. (32).

For unsteady flow problems, the use of Euler explicit method may not be able to provide accurate time evolution solution. For this case, the dual time stepping (DTS) [51] method is adopted in this work. By adding a pseudo-time derivative to Eq. (29), we have

$$\boldsymbol{M}\frac{\partial \boldsymbol{U}}{\partial \tau^*} + \boldsymbol{M}\frac{\partial \boldsymbol{U}}{\partial t} = -\boldsymbol{R}, \tag{34}$$

where τ^* is the pseudo time. By defining a new residual vector $\boldsymbol{R}^*$ as

$$\boldsymbol{R}^* = \boldsymbol{R} + \boldsymbol{M}\frac{\partial \boldsymbol{U}}{\partial t}. \tag{35}$$

Eq. (34) can be reduced to

$$\boldsymbol{M}\frac{\partial \boldsymbol{U}}{\partial \tau^*} = -\boldsymbol{R}^*. \tag{36}$$

Eq. (36) is similar to Eq. (29). In the pseudo-time domain, it is a steady state problem. Thus the point iterative method described above can be used to solve Eq. (36). To keep time accuracy, the time derivative in $\boldsymbol{R}^*$ is approximated by the second order finite difference scheme.

2.5. Computational sequence

The computational sequence for the present high-order LSFD-FV method is summarized as below:

(1) Select the supporting cells for every cell. Calculate and store the weighting coefficient matrix $\boldsymbol{W}$ in Eq. (12) for every cell.
(2) Specify a streaming time step δ_t which should ensure that the location of the virtual lattice velocity point of D2Q9 model is within either the left or the right cell of the interface. Then compute the single relaxation parameter τ by Eq. (17).
(3) Calculate the derivatives of conservative variables with Eq. (12) based on the functional values at the cell center.
(4) For the considered interface, reconstruct the macroscopic flow variables at the lattice velocity point of D2Q9 with Eq. (20). Calculate $f_\alpha^{eq}(\boldsymbol{r}-\boldsymbol{e}_\alpha\delta_t, t-\delta_t)$ and $f_\alpha^{eq}(\boldsymbol{r}, t)$ using Eq. (16). Then, use Eq. (19) to calculate $f_\alpha^{neq}(\boldsymbol{r}, t)$ and further compute the fluxes at every Gaussian quadrature point by Eqs. (14) and (15). Then convert fluxes with Eq. (24) and further apply the Gaussian quadrature rule to calculate total fluxes across the considered interface.
(5) Solve resultant ordinary differential equations by Eq. (33) for steady problems and Eq. (36) for unsteady problems.
(6) Repeat steps (2)–(5) until the converged solution or the specified time is reached.

3. Numerical examples

In this section, a series of benchmark cases are tested to validate the accuracy, efficiency, stability and flexibility of present high-order LSFD-FV method. In the simulation, the small constant physical time step of $\Delta t = 10^{-4}$ is chosen and the pseudo time step in DTS is taken as $\Delta\tau^* = 10^{-4}$. The maximum number of pseudo iterations is set as $k_{\max} = 20$ and the criterion for pseudo steady state convergence per physical time step requires that the unsteady residual $\boldsymbol{R}^*$ drops by five orders of magnitude.

3.1. Accuracy test

3.1.1. Accuracy test of high order LSFD-FV method

Firstly, the accuracy of the present high-order LSFD-FV method without LBFS is tested by the isentropic vortex transport problem [1]. This test problem involves convection of an isentropic vortex in the inviscid flow, where the inviscid fluxes are calculated by the Roe scheme [52]. The free-stream conditions are $(\rho, u, v, P) = (1, 1, 1, 1)$. The following perturbations are added to the free stream with no entropy gradient in the flow field,

$$(\delta u, \delta v) = \frac{\varepsilon}{2\pi} e^{0.5(1-r^2)}(-\overline{y}, \overline{x}), \tag{37}$$

while the temperature and entropy are

$$T = 1 - \frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2}, \quad S = 1, \tag{38}$$

where $(\overline{x}, \overline{y}) = (x-5, y-5), r^2 = \overline{x}^2 + \overline{y}^2$ and the vortex strength $\varepsilon = 5$. The density ρ and pressure p are calculated from the temperature T and the entropy S by

$$T = \frac{p}{\rho}, \qquad S = \frac{p}{\rho^\gamma}. \tag{39}$$

The computational domain is $[0, 10] \times [0, 10]$. Periodic boundary condition is applied to all boundaries. The regular triangular grids are used and the grid spacing ranges from 1 to 1/8. The accuracy tests of the k-exact FV method are also performed for the purpose of comparison.

Table 1
Accuracy test results for the isotropic vortex problem on regular triangular grids.

Schemes	Grid size	L_∞ errors	Order	L_1 errors	Order
4th order *k*-exact FV	1/20	3.59E-02		1.27E-03	
	1/40	3.08E-03	3.544	1.04E-04	3.599
	1/60	5.76E-04	4.133	1.89E-05	4.210
	1/80	1.52E-04	4.619	5.61E-06	4.228
4th order LSFD FV	1/20	3.52E-02		1.24E-03	
	1/40	2.87E-03	3.619	9.82E-05	3.657
	1/60	5.21E-04	4.204	1.75E-05	4.255
	1/80	1.25E-04	4.956	5.49E-06	4.027

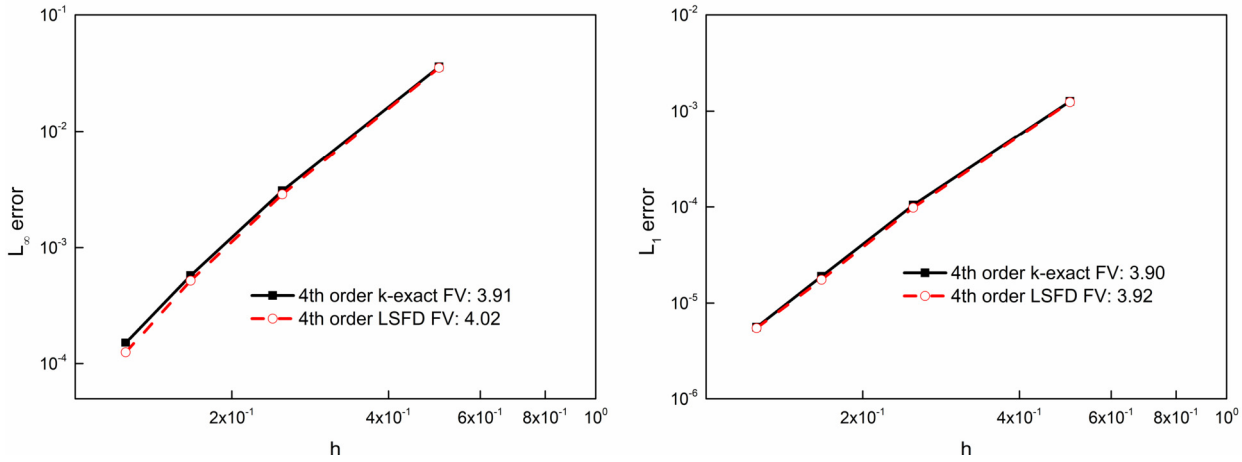


Fig. 2. Accuracy comparison between the present high-order LSFD-FV method and the high-order k-exact FV method for the isentropic vortex problem on regular triangular grids. The numbers in the figures denote the slopes of the linearly fitted lines. For the L_∞ error, the slopes of the LSFD-FV and k-exact FV methods are 4.02 and 3.91, respectively. For the L_1 error, the slopes of the LSFD-FV and k-exact FV methods are 3.92 and 3.90, respectively.

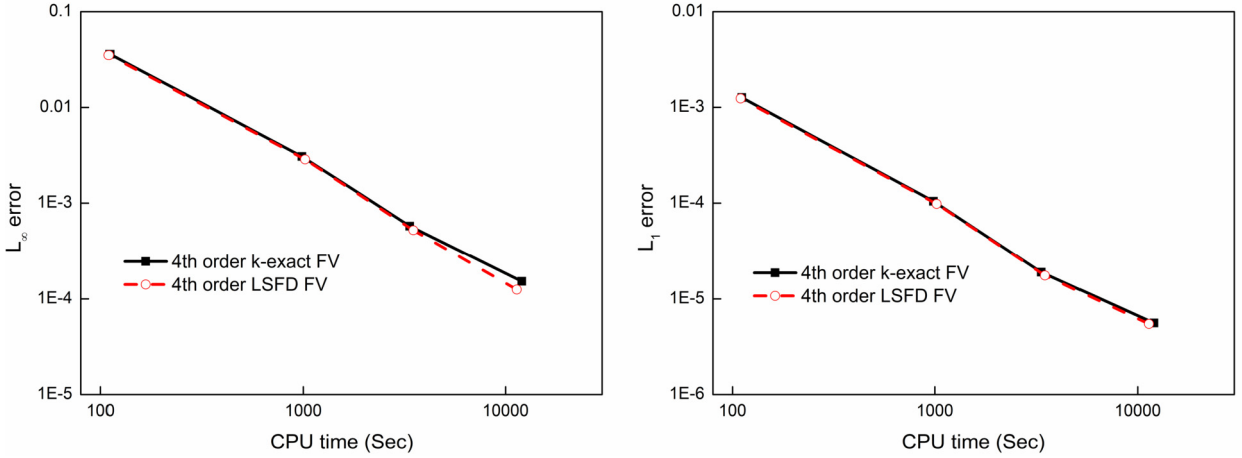


Fig. 3. Efficiency comparison between the present high-order LSFD-FV method and the high-order k-exact FV method for the isentropic vortex problem on regular triangular grids.

Results at $t = 2.0$ are extracted. The errors of density in terms of L_1 and L_∞ norms and the convergence rates are listed in Table 1. Fig. 2 shows the accuracy comparison. These results verify the fourth-order accuracy of both FV methods using LSFD approximation and k-exact approximation. Furthermore, the slopes of linearly fitted lines in Fig. 2 reveal that the accuracy of LSFD-FV method is slightly higher than the k-exact FV method. Additionally, from the relative errors listed in Table 1, the solutions computed by the fourth-order LSFD-FV method are more accurate than those obtained by the fourth-order k-exact FV method. The efficiency comparison in Fig. 3 shows that, for this unsteady simulation, the efficiency of LSFD-FV method is almost the same as the k-exact FV method. Moreover, for the grid of $h = 1/80$, the LSFD-FV method takes a little less CPU time (around 5%) than the k-exact FV method.

3.1.2. Accuracy test of high-order LSFD-FV method with LBFS

For incompressible flow simulations, the high-order LSFD-FV method uses LBFS to evaluate the inviscid and viscous fluxes simultaneously. From the first test case, the fourth-order accuracy of the LSFD-FV method has been validated. However, the LBFS only has the second-order accuracy in solution reconstruction. Naturally, one may doubt the overall accuracy of the whole method. Nevertheless, it is noteworthy that the second-order accuracy of LBFS is the local reconstruction accuracy in terms of δ_x or δ_t rather than the grid spacing h. Theoretically, if h is far larger than δ_x, the global or overall accuracy of the whole method which is assessed in terms of h will not be affected significantly by the LBFS. For the present high-order LSFD-FV method, two Gauss-Legendre quadrature points are chosen on every interface of the control cell to ensure the fourth-order accuracy of the surface integral. To avoid the extrapolation, generally, δ_x must satisfy the limitation that the virtual streaming nodes should be inside the two control cells as depicted in Fig. 1. Thus, δ_x is restricted by the distribution of the Gauss-Legendre quadrature points and the control cell. Specifically, for unstructured grids, one effective and practical

Table 2
Accuracy test results for decaying vortex flow on regular triangular grids.

Schemes	Grid size	L_1 errors	Order	L_2 errors	Order
4th order *k*-exact FV	1/10	2.22E-02		2.51E-02	
	1/20	2.30E-03	3.272	2.57E-03	3.287
	1/30	4.64E-04	3.948	5.19E-04	3.948
	1/40	1.45E-04	4.050	1.62E-04	4.050
	1/80	8.28E-06	4.129	9.26E-06	4.127
4th order LSFD FV	1/10	1.99E-02		2.24E-02	
	1/20	2.00E-03	3.313	2.24E-03	3.323
	1/30	4.06E-04	3.933	4.55E-04	3.933
	1/40	1.27E-04	4.028	1.43E-04	4.032
	1/80	7.34E-06	4.117	8.23E-06	4.115
2nd order *k*-exact/LSFD FV	1/30	6.78E-03		6.09E-03	
	1/40	3.95E-03	1.881	3.55E-03	1.876
	1/80	1.02E-03	1.951	9.20E-04	1.949
	1/100	6.57E-04	1.982	5.91E-04	1.981
	1/160	2.58E-04	1.991	2.32E-04	1.990

Note: The 2nd order k-exact FV and 2nd order LSFD-FV methods are degraded to the conventional 2nd order FV method. Thus, they give the totally same results.

limitation for δ_x is that $\delta_x = (1/2 - \sqrt{3}/2) l_{\min} \sin\theta \approx 0.2113 l_{\min} \sin\theta$, where $l_{\min}$ and θ are the minimum length of the edge and the minimum interior angle among the left and right cells of the interface. Furthermore, due to the independence between the streaming time step and the time marching step, the small streaming distance does not affect the convergence rate.

Based on the discussion above, the decaying vortex flow problem which has an analytic solution is used to test the accuracy of the LSFD-FV method with LBFS numerically. The analytical solution of the problem satisfying the 2D incompressible N-S equations reads

$$\begin{aligned} u(x,y,t) &= -U\cos(\pi x/L)\sin(\pi y/L)e^{-2\pi^2 Ut/(\mathrm{Re}L)}, \\ v(x,y,t) &= U\sin(\pi x/L)\cos(\pi y/L)e^{-2\pi^2 Ut/(\mathrm{Re}L)}, \\ \rho(x,y,t) &= \rho_0 - \frac{\rho_0 U^2}{4c_s^2}\left[\cos(2\pi x/L) + \cos(2\pi y/L)\right]e^{-4\pi^2 Ut/(\mathrm{Re}L)}. \end{aligned} \tag{40}$$

Numerical simulations are conducted on the computational domain of $[-L, L] \times [-L, L]$ at a Reynolds number of $Re = UL/\nu = 10$. The relaxation parameter τ is set as 0.8 and ρ_0 is taken as 1. The streaming distance δ_x is selected as $0.2 l_{\min} \sin\theta$. The solutions at $t = L/U = 1$ are computed and the relative errors of velocity component u are measured using the L_2 norm and L_1 norm, which are defined as

$$L_1(u) = \frac{1}{Ncell}\sum_{i=1}^{Ncell}\left(\left|\frac{u_i - u_i^e}{U}\right|\right), \qquad L_2(u) = \left(\frac{1}{Ncell}\sum_{i=1}^{Ncell}\left(\frac{u_i - u_i^e}{U}\right)^2\right)^{\frac{1}{2}}, \tag{41}$$

where u_i and u_i^e represent the numerical result and the exact solution, respectively. *Ncell* is the number of the cells.

For the convergence study, the regular triangular grids with the spacing of $h = 1/10$ to $1/160$ are used. L_1 and L_2 norms of the relative errors of velocity component u and the rates of convergence are shown in Table 2. Fig. 4 shows the accuracy comparison of the high-order and second-order LSFD-FV method. As can be seen in Table 2 and Fig. 4, both the k-exact FV method and LSFD-FV method with LBFS achieve the theoretical accuracy, which is consistent with the above discussion of accuracy. In addition, the solutions calculated by both high-order methods are much more accurate than those obtained by the second-order methods. It is noteworthy that the relative errors obtained by high-order LSFD-FV method are slightly smaller than those computed by the high-order k-exact FV method. This proves that the high-order LSFD-FV method is slightly more accurate than the high-order k-exact FV method again. Furthermore, the efficiency comparison between the fourth- and the second-order LSFD-FV method is shown in Fig. 5, which indicates that the fourth-order LSFD-FV method with LBFS requires less CPU time than the second-order counterpart to achieve the same accuracy. This means that the high-order LSFD-FV method is more efficient than the second-order one.

3.2. Steady plane Poiseuille flow

The plane Poiseuille flow which is driven by a pressure gradient is a classic benchmark test for incompressible viscous flows. The analytical solution of this problem can be easily derived, which provides an ideal test case to investigate the accuracy of the high-order LSFD-FV method with LBFS and compare it with the second-order counterpart. The physical

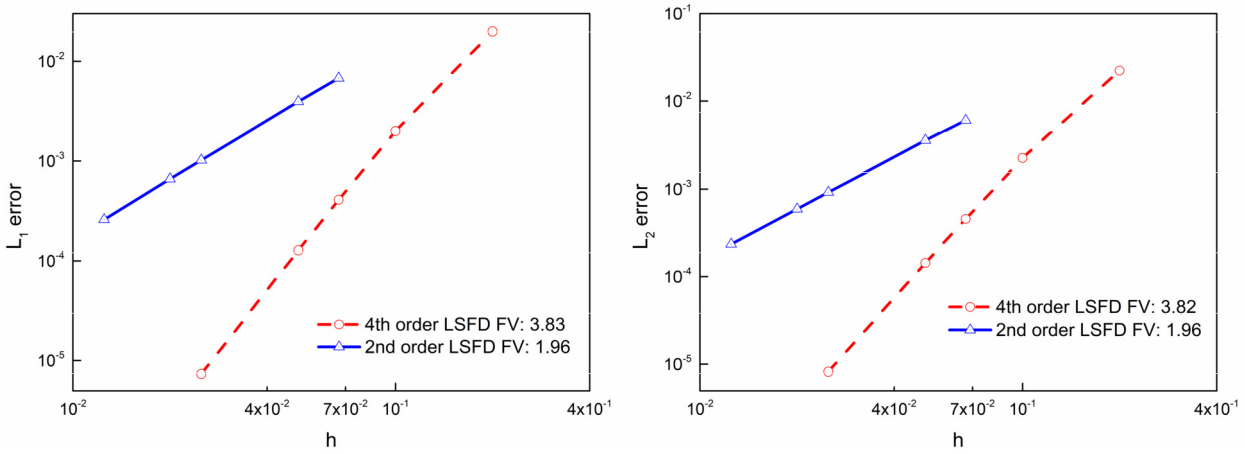


Fig. 4. Convergence studies for 2D decaying vortex flow at $t = 1$. The numbers in the figures denote the slopes of the linearly fitted lines. For the L_1 error, the slopes of the fourth- and second-order LSFD-FV methods are 3.83 and 1.96, respectively. For the L_2 error, the slopes of the fourth- and second-order LSFD-FV methods are 3.82 and 1.96, respectively.

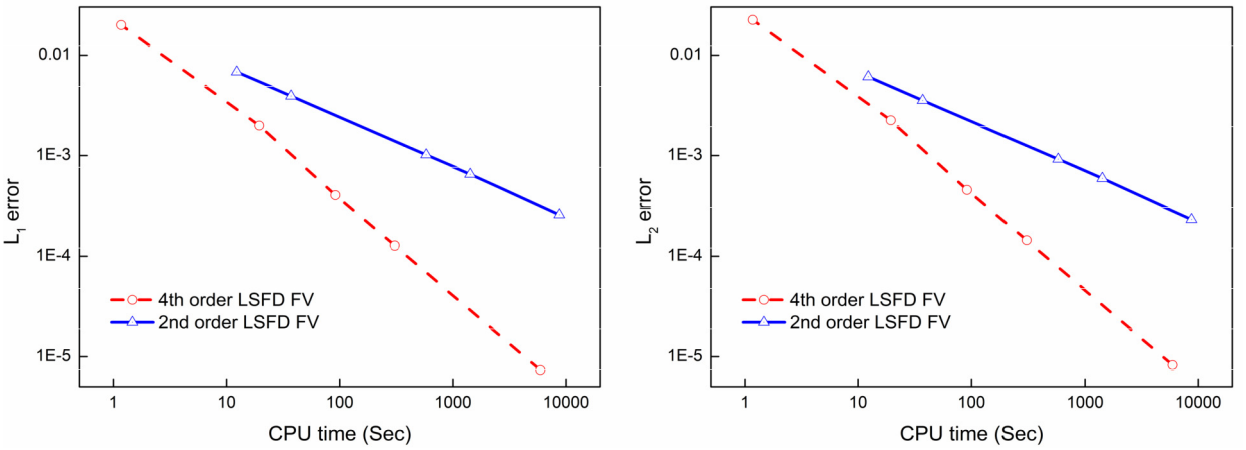


Fig. 5. Efficiency comparison between the fourth- and second-order LSFD-FV methods for 2D decaying vortex flow at $t = 1$.

configuration of this problem is that the viscous flow between 2 parallel plates is driven by a uniform pressure gradient. From the Navier-Stokes equations, the analytical solution of the velocity profile for this problem can be given as

$$u(y) = \frac{(y^2 - Hy)}{2\mu}\frac{\partial p}{\partial x}, \tag{42}$$

where $\partial p/\partial x$ is the pressure gradient. In the present simulation, the computational domain is set as $[0, 1] \times [0, 1]$, $H = 1$, the dynamic viscosity of the fluid is $\mu = 0.05$ and $\partial p/\partial x = -0.001$. The uniform quadrilateral grids with the spacing of $h = 1/10$, $1/20$ and $1/40$ are applied. No-slip boundary condition is implemented on the upper and lower solid boundaries, while uniform pressure boundary condition is applied on the inlet and outlet of the channel.

Fig. 6 shows the computed u-component profiles by the high-order LSFD-FV method with different mesh points ($N = 10$, 20 and 40). Clearly, good mesh convergence of the solution is achieved and the velocity profile is well captured even with the coarse mesh by the current high-order method. This indicates that the velocity distribution can be accurately predicted by the high-order LSFD-FV method with LBFS. Furthermore, as can be seen in Fig. 7, on the same coarse mesh of $h = 1/20$, the high-order LSFD-FV method gives more accurate results than the second-order one. Such outcomes evidently prove the better performance of the high-order LSFD-FV method than the second-order one in terms of numerical accuracy.

3.3. Lid-driven cavity flow

The case of plane Poiseuille flow can be regarded as a simple one-dimensional problem with negligible velocity in the vertical direction. To comprehensively evaluate the performance of the high-order LSFD-FV method with LBFS, the two-dimensional lid-driven flow in a square cavity is tested. As one of the most important benchmark cases for validating new

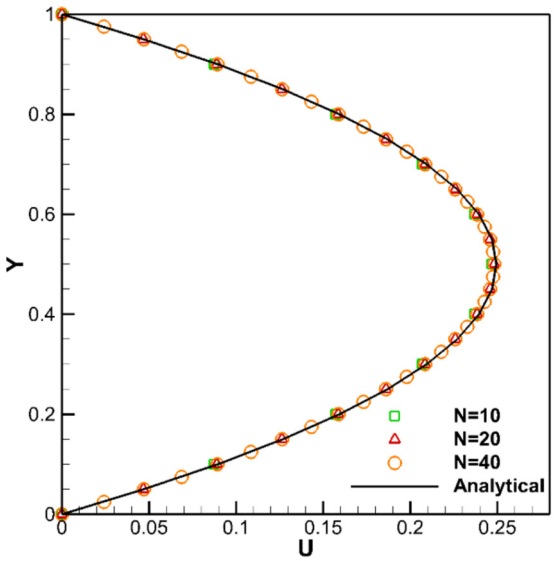


Fig. 6. u-component profiles given by the fourth-order LSFD-FV method on the uniform quadrilateral grids. The solid line denotes the analytical solution. Symbols are computed with different meshes.

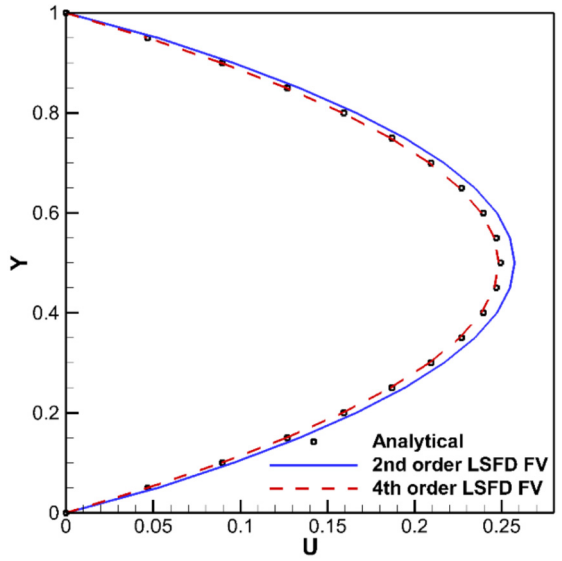


Fig. 7. Comparison of u-component profile on uniform quadrilateral grids of $h = 1/20$.

numerical methods in the simulation of incompressible viscous fluid flow problems, this steady case is also used to compare the performance of the high order LSFD-FV and k-exact FV methods. Furthermore, the memory cost and computational efficiency between the high-order LSFD-FV method and the second-order one are compared.

The Reynolds number for the problem is defined as $Re = UL/\nu$, where U is the velocity of the top lid and L is the length of the square cavity. The computational domain of $[0, 1] \times [0, 1]$ with unstructured grids is presented in Fig. 8, where the non-uniform mesh spacings are respectively $h = 1/20$ for the cells in the middle and $h = 1/45$ for the cells near the walls, which has 2734 cells in total. The streamlines at $Re = 1000$ given by the high-order LSFD-FV method are shown in Fig. 8. For three Reynolds numbers of $Re = 1000$, 3200 and 5000, the u-velocity and v-velocity profiles along the vertical and horizontal central lines obtained by the high-order and second-order LSFD-FV method, and benchmark data of Ghia et al. [53] are presented in Fig. 9. Furthermore, to compare the computational efficiency and memory cost of the second- and high-order schemes, the mesh is refined to obtain comparable results for the second-order scheme. For internal cells, the mesh spacing is taken as $h = 1/60$, while for cells near the walls, h is reduced to $h = 1/100$. The total number of cells is 11032. The corresponding results are shown in Fig. 10 and Table 3. From the results, it is clear that on the coarse mesh of 2734 cells, the high-order LSFD-FV method can give more accurate results than the second-order one. Meanwhile, for

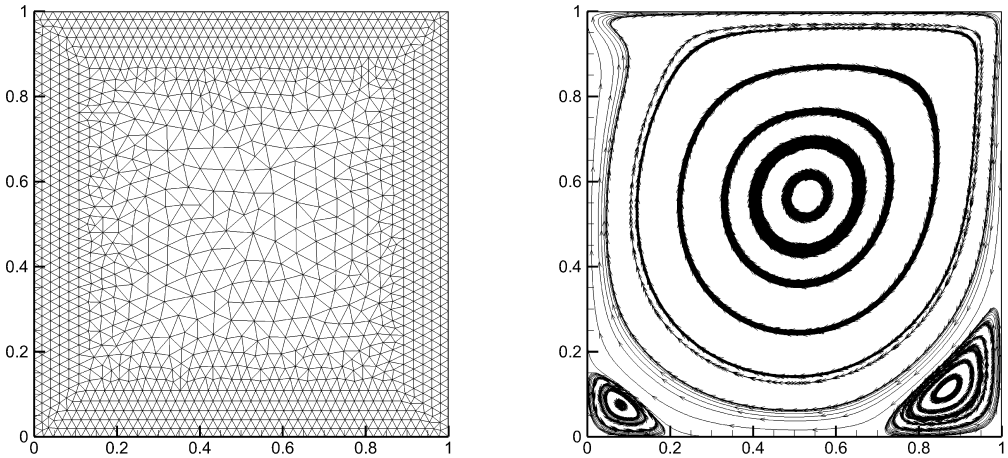


Fig. 8. Lid-driven cavity flow: mesh and streamlines for the high-order LSFD-FV method at $Re = 1000$.

the similar accuracy achieved on different meshes, the high-order LSFD-FV method is almost three times faster than the second-order method with less memory cost. Such outcomes provide strong evidence that the high-order LSFD-FV method performs better than the second-order LSFD-FV method in terms of numerical accuracy and computational efficiency.

In order to further evaluate the performance of the high-order LSFD-FV method, the comparison between the high-order LSFD-FV and k-exact FV methods for this steady case is conducted. The results of $Re = 3200$ and 5000 on the mesh of 2734 cells are shown in Fig. 11 and Fig. 12, respectively. Although the two methods can give comparable results, the high-order LSFD-FV method converges faster than the high-order k-exact FV method with less iterations. Furthermore, for the case of $Re = 5000$, the high-order LSFD-FV method uses only 85% of the CPU time spent by the high-order k-exact FV method. Such results clearly demonstrate the better efficiency of the high-order LSFD-FV method than the k-exact FV method with the same accuracy.

3.4. Viscous flow past a stationary circular cylinder

To further illustrate the capability of the present LSFD-FV method for problems with curved boundary, the viscous flow past a stationary circular cylinder at various Reynolds numbers ($Re = 20$, 40 and 100) is simulated. Physically, the simulation models the incoming viscous fluid with a constant free-stream velocity flowing over a fixed circular cylinder. To meet the incompressible limit, the free-stream velocity is chosen as $U_0 = 0.1$ and the Reynolds number, which affects the flow pattern, is defined as $Re = U_0 L/\nu$, where L is the diameter of the circular cylinder. The pressure coefficient C_p, lift coefficient C_l, drag coefficient C_d and Strouhal number S_t are used to quantify the numerical results. They are defined as

$$C_p = \frac{p_w - p_0}{\rho_0 U_0^2/2}, \qquad C_l = \frac{F_l}{\rho_0 U_0^2/2}, \qquad C_d = \frac{F_d}{\rho_0 U_0^2/2}, \qquad S_t = \frac{f_q L}{U_0}, \tag{43}$$

where p_0 is the free-stream pressure, p_w is the pressure on the cylinder surface, F_l is the lift force, F_d is the drag force and f_q is the vortex shedding frequency. In addition, the geometrical quantities of the eddies, such as the recirculation length L_s and the separation angle θ_s, are also measured for steady flows.

No-slip boundary condition is implemented on the cylinder surface, whereas far field free-stream condition is enforced on the outer boundary. Unstructured grid (see Fig. 13) with 65 points on the cylinder surface and 5908 cells in total is used for the simulation of steady flow at $Re = 20$, where the outer boundary is taken $25L$ away from the center of the cylinder. For the steady simulation of $Re = 40$ and the unsteady simulation of $Re = 100$, due to the smaller physical viscosity, a refined mesh (see Fig. 14) with 65 points on the cylinder surface and 11626 cells in total is used with the far-field boundary at $55L$ away from the center of the cylinder. For quantitative comparisons, the drag coefficient C_d, the length of the recirculation zone L_s and the separation angle θ_s in steady cases are tabulated and compared with the reference data [44,54–58] as shown in Table 4. The computed dynamic parameters for $Re = 100$ and the corresponding reference data [44,54,59,60] are presented in Table 5. As can be seen, on the same unstructured grids, the results of LSFD-FV methods are all within the range of reference data, while the high-order method outperforms the second-order method, which validates the accuracy and flexibility of the present high-order method on unstructured grids.

3.5. Inviscid flow past a stationary circular cylinder

In the present LSFD-FV method with LBFS, when τ is set as 0.5, the contribution from the non-equilibrium distribution function is vanished, and only the equilibrium distribution function has contribution to the flux calculation. In this way, we

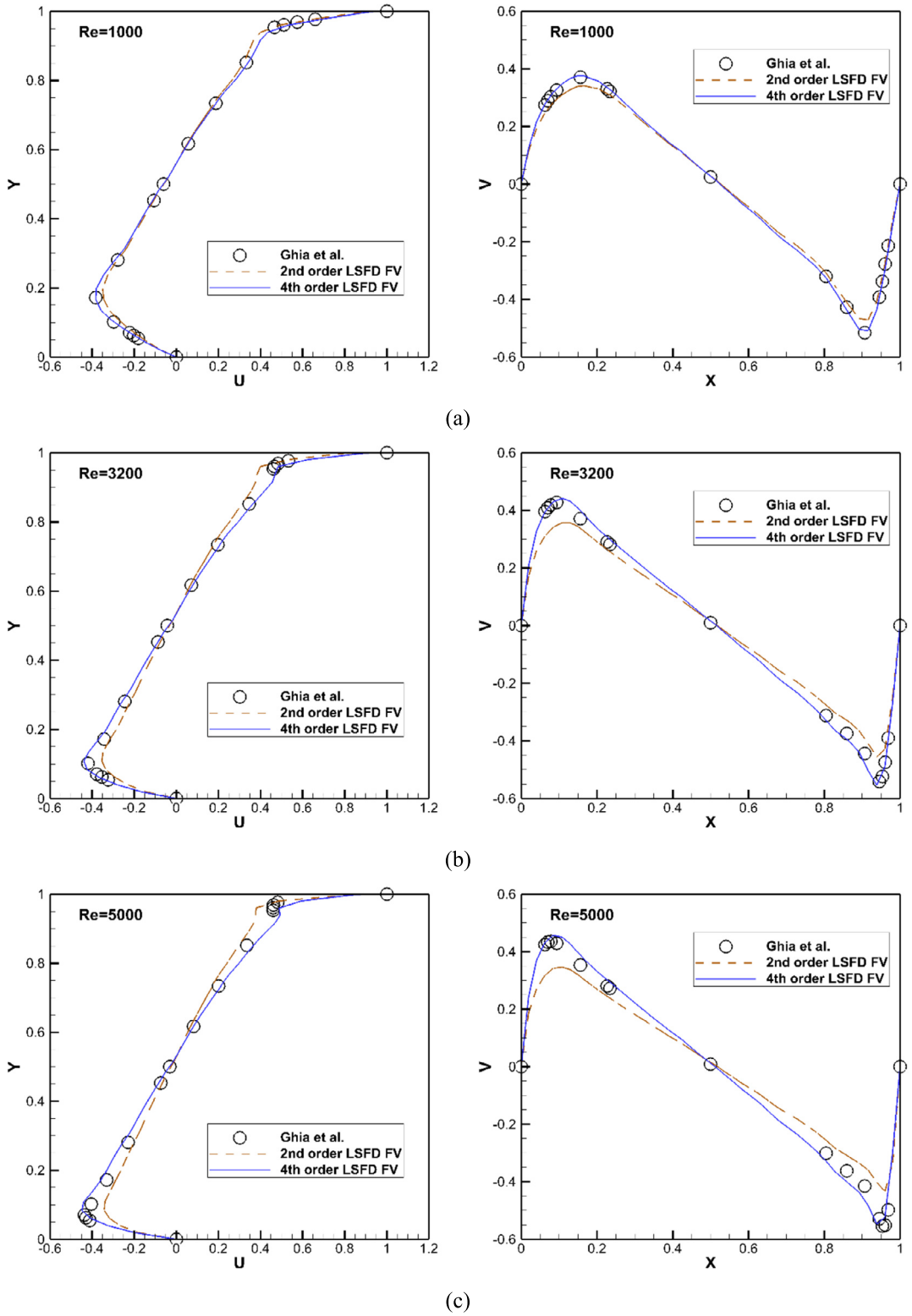
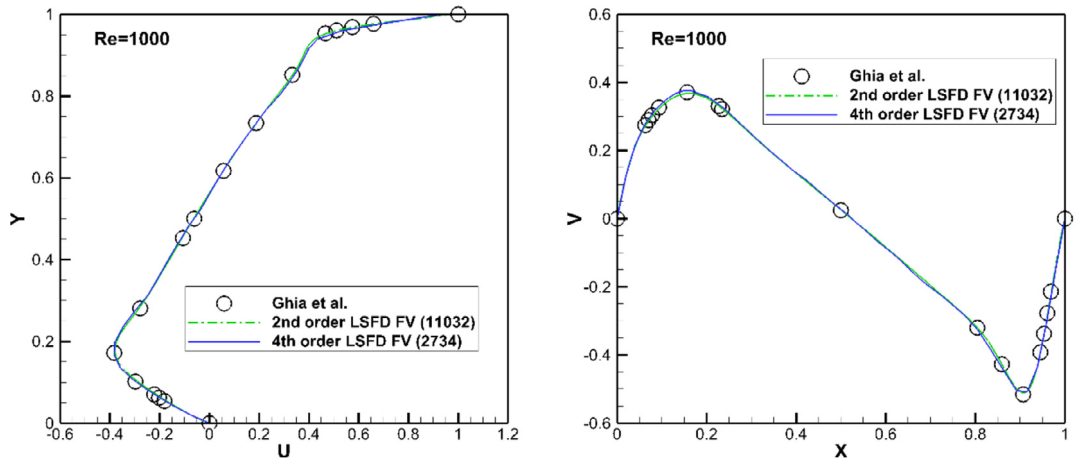
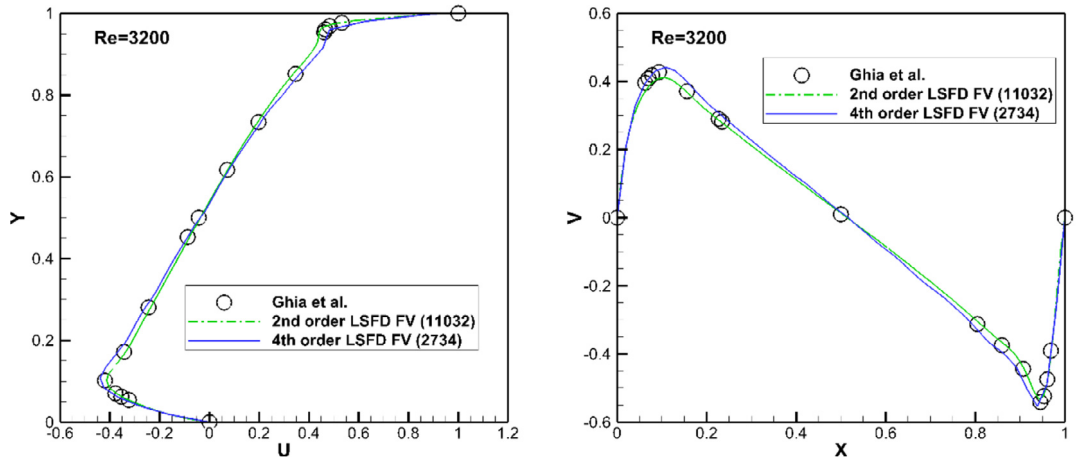


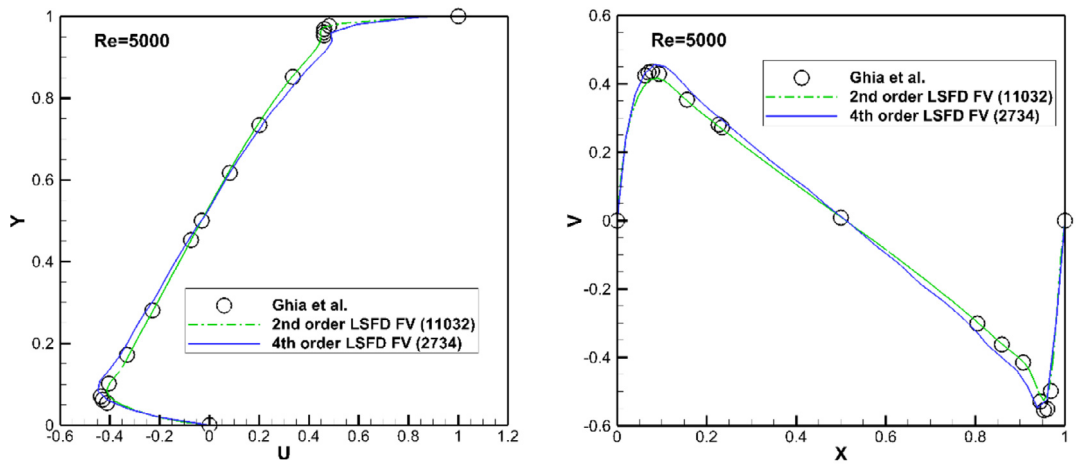
Fig. 9. Comparison of the u-velocity (left) and v-velocity (right) profiles along vertical and horizontal central lines at (a) $Re = 1000$, (b) $Re = 3200$ and (c) $Re = 5000$ on the mesh of 2734 cells.



(a)



(b)



(c)

Fig. 10. Comparison of the u-velocity (left) and v-velocity (right) profiles along vertical and horizontal central lines at (a) $Re = 1000$, (b) $Re = 3200$ and (c) $Re = 5000$ on different grids.

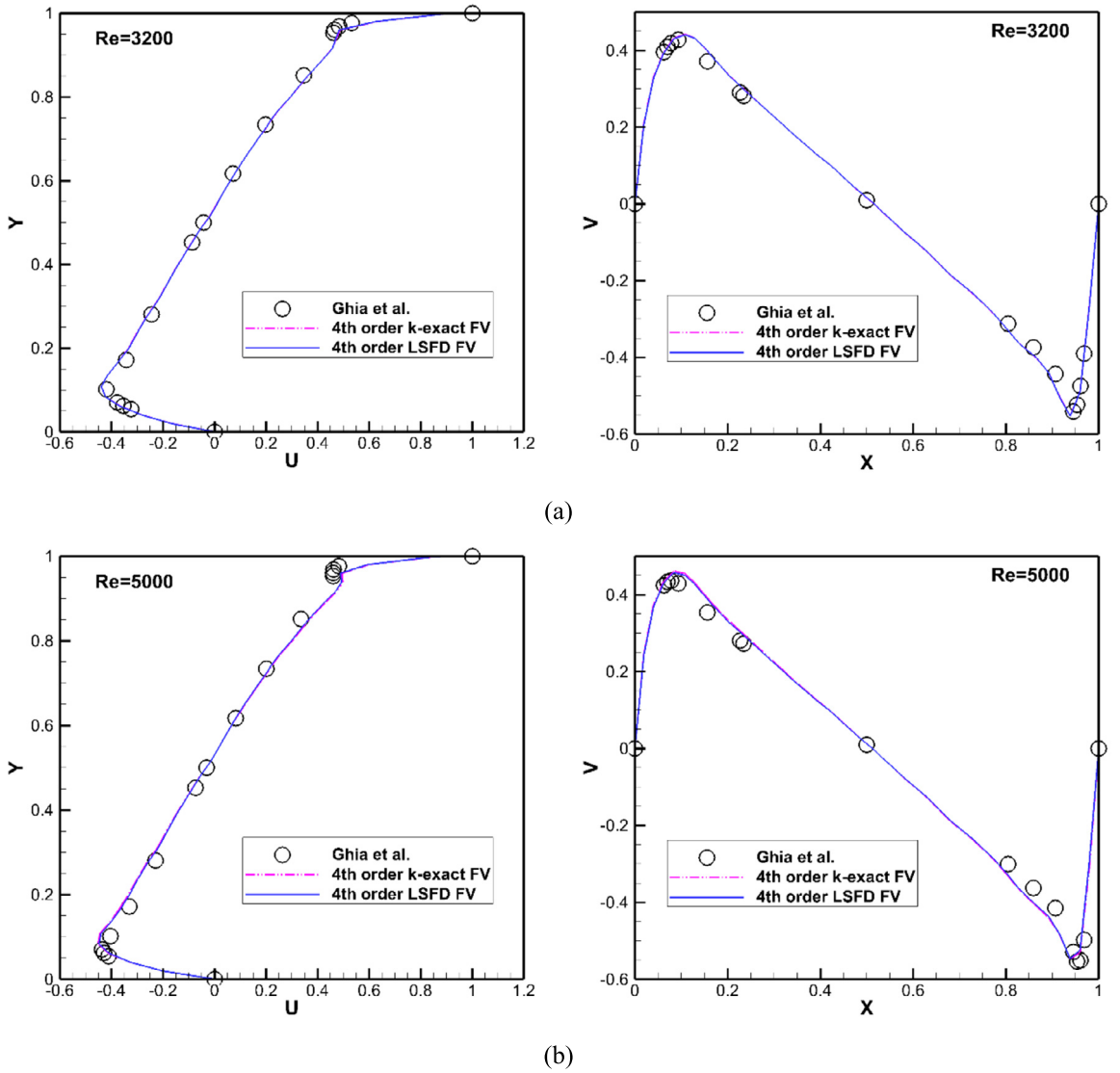
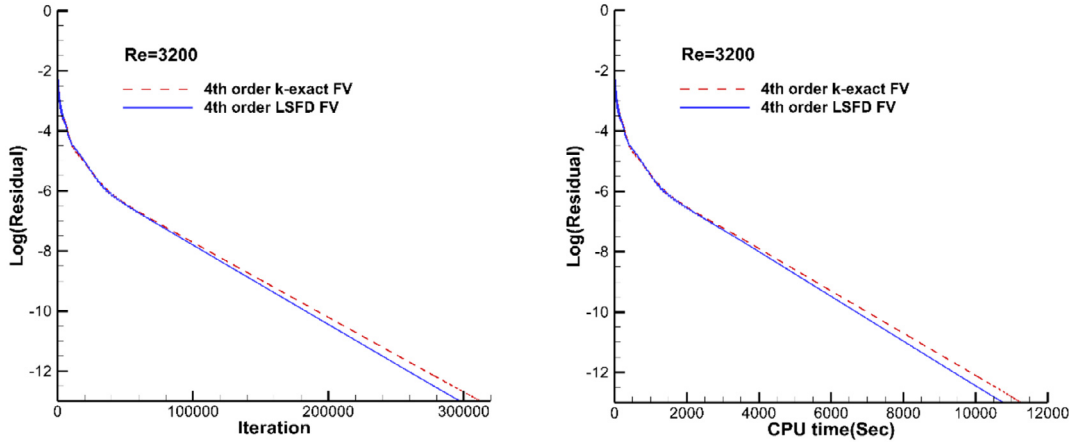


Fig. 11. Comparison of the u-velocity (left) and v-velocity (right) profiles along vertical and horizontal central lines at (a) $Re = 3200$ and (b) $Re = 5000$ between the high-order LSFD-FV method and high-order k-exact FV method. The mesh has 2734 cells.

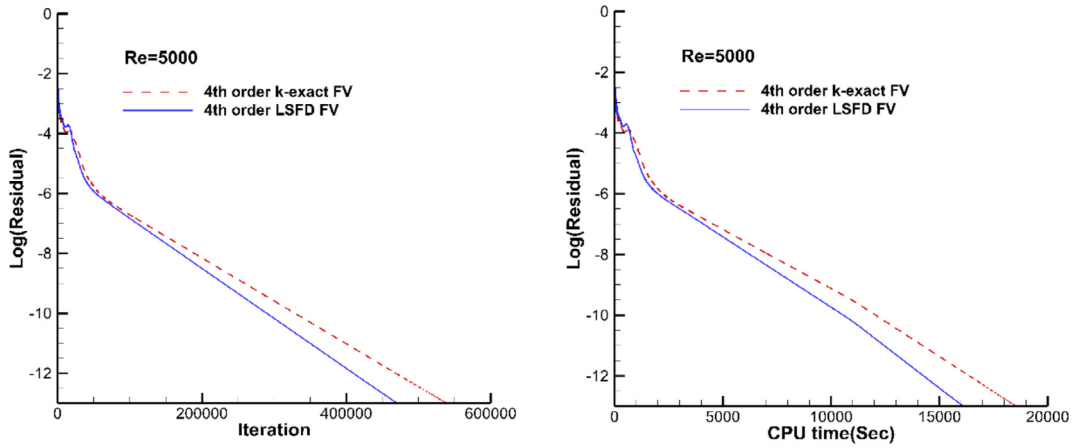
Table 3
Comparison of the memory cost and computational time for lid-driven cavity flow.

Cells of mesh	*Re*	4th order LSFD-FV method		2nd order LSFD-FV method	
		Memory (Mb)	CPU time (s)	Memory (Mb)	CPU time (s)
2734	1000	**8.0**	**742.06**	3.2	302.38
	3200	**8.0**	**2175.25**	3.2	923.41
	5000	**8.0**	**3328.03**	3.1	1357.14
11032	1000			**9.7**	**2126.75**
	3200			**9.7**	**5464.59**
	5000			**9.6**	**9093.02**

further test the LSFD-FV method for inviscid flows by setting $\tau = 0.5$. Inviscid flow past a circular cylinder has analytical solution in fluid mechanics, which is usually used to verify the accuracy of high-order methods. Since there is no boundary layer around the cylinder surface for this case, the computational mesh used is much coarser than that for the viscous flow. In the present simulation, the computational domain is divided into 2456 triangular grids and 40 points are distributed on



(a)



(b)

Fig. 12. Comparison of the convergence history (left) and efficiency (right) between the high-order LSFD-FV and k-exact FV methods at (a) $Re = 3200$ and (b) $Re = 5000$. The mesh has 2734 cells.

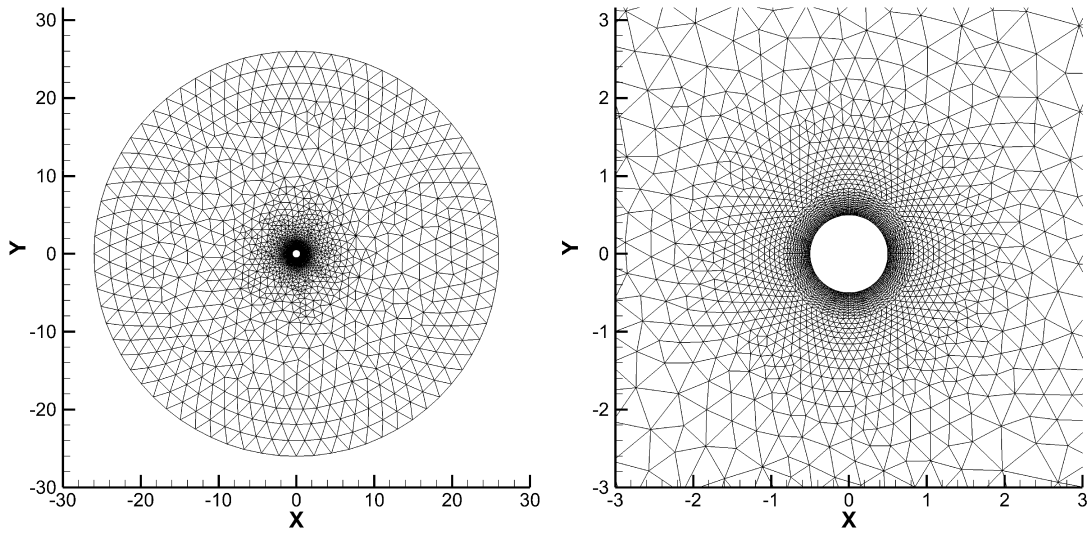


Fig. 13. Mesh of the viscous flow past a circular cylinder at $Re = 20$. 65 grids on the cylinder; 5908 cells.

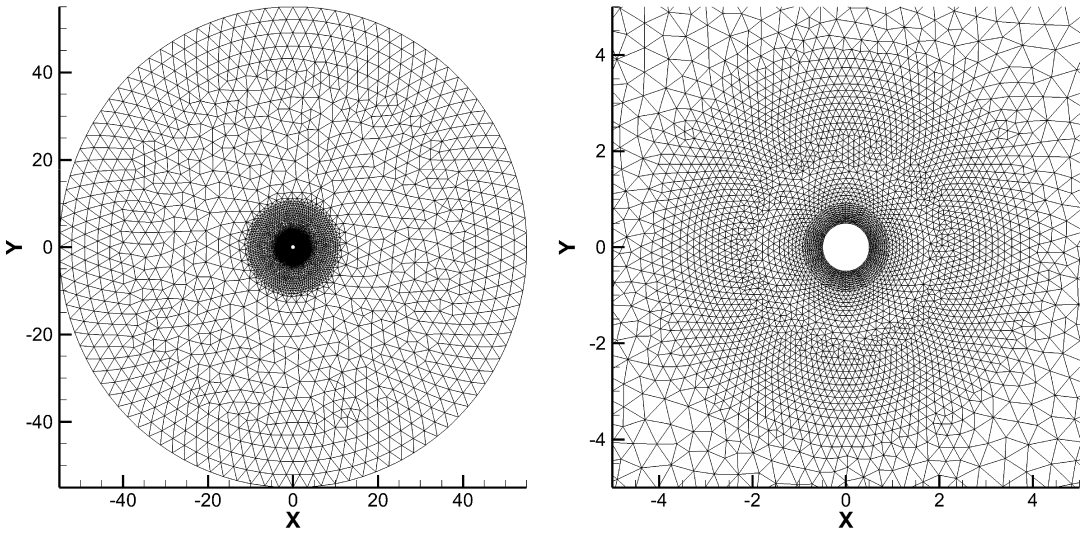


Fig. 14. Mesh of the viscous flow past a circular cylinder at $Re = 40$ and 100. 65 grids on the cylinder; 11626 cells.

Table 4
Comparison of drag coefficient, recirculation length and separation angle for steady flow past a stationary circular cylinder at $Re = 20$ and 40.

Re	References	C_d	L_s/D	θ_s □
20	Dennis and Chang [55]	2.05	0.94	43.7
	Shukla et al. [56]	2.07	0.92	43.3
	Wu and Shu et al. [57]	2.091	0.93	–
	2nd order LSFD-FV method	**2.08**	**0.848**	**40.995**
	4th order LSFD-FV method	**2.072**	**0.92**	**43.78**
40	He and Doolen [58]	1.499	2.245	52.84
	Pellerin et al. [54]	1.505	2.259	53.64
	Shu et al. [44]	1.53	2.24	52.69
	2nd order LSFD-FV method	**1.524**	**2.114**	**55.08**
	4th order LSFD-FV method	**1.508**	**2.249**	**53.63**

Table 5
Comparison of dynamic parameters for unsteady flow past a stationary circular cylinder at $Re = 100$.

Re	References	C_l	C_d	S_t
100	Braza et al. [59]	± 0.30	1.28 ± 0.02	0.16
	Liu et al. [60]	± 0.339	1.350 ± 0.02	0.164
	Shu et al. [44]	± 0.33	1.334 ± 0.02	0.164
	Pellerin et al. [54]	± 0.325	1.325	0.164
	2nd order LSFD-FV method	$\mathbf{\pm 0.327}$	1.338 ± 0.011	**0.158**
	4th order LSFD-FV method	$\mathbf{\pm 0.332}$	1.337 ± 0.011	**0.164**

the cylinder surface. The far-field boundary is taken 30 diameters away from the geometrical center of the cylinder. The mesh is shown in Fig. 15. On the cylinder surface, no-penetration condition is applied.

The pressure contours and the streamlines near the rear stagnation point obtained by the high-order LSFD-FV method are shown in Fig. 16. As can be seen, the symmetry of the solutions is well captured. The streamlines pass through the cylinder smoothly and no vortex emerges. From the comparison of pressure coefficient computed by the high- and second-order LSFD-FV methods shown in Fig. 17, it is clear that the high-order method outperforms the second-order one and the results of the high-order method match well with the analytical solution. This test case well demonstrates the good capability of the present high-order LSFD-FV method for simulation of inviscid flows with curved boundary.

4. Conclusions

In this paper, a high-order least square-based finite difference-finite volume method is developed on arbitrary grids. This high-order method is based on the polynomial approximation with a Taylor series expansion. Different from the high-order k-exact FV method, the LSFD-FV method directly uses the Taylor series expansion as the approximation function

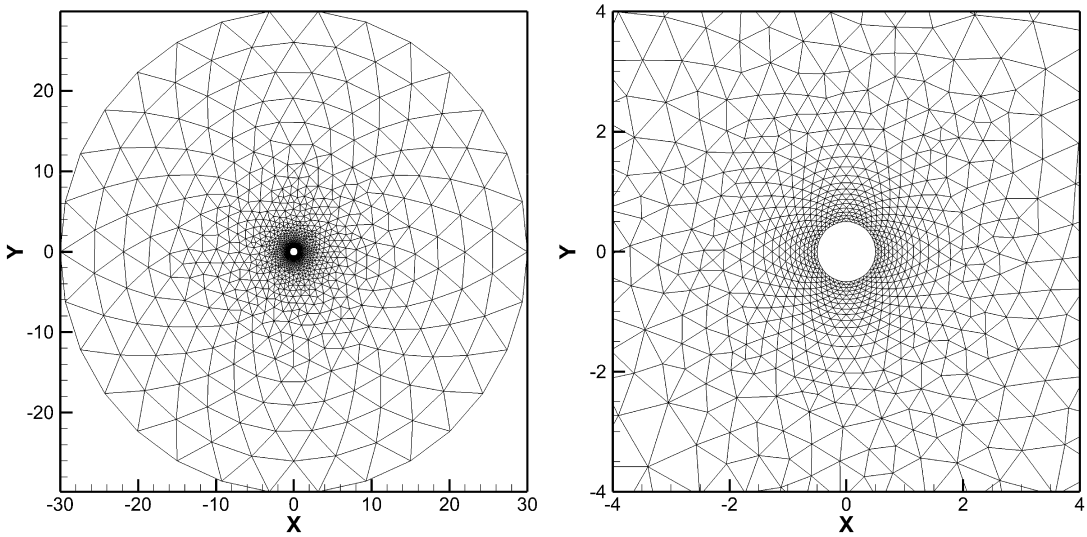


Fig. 15. Mesh of the inviscid flow past a circular cylinder.

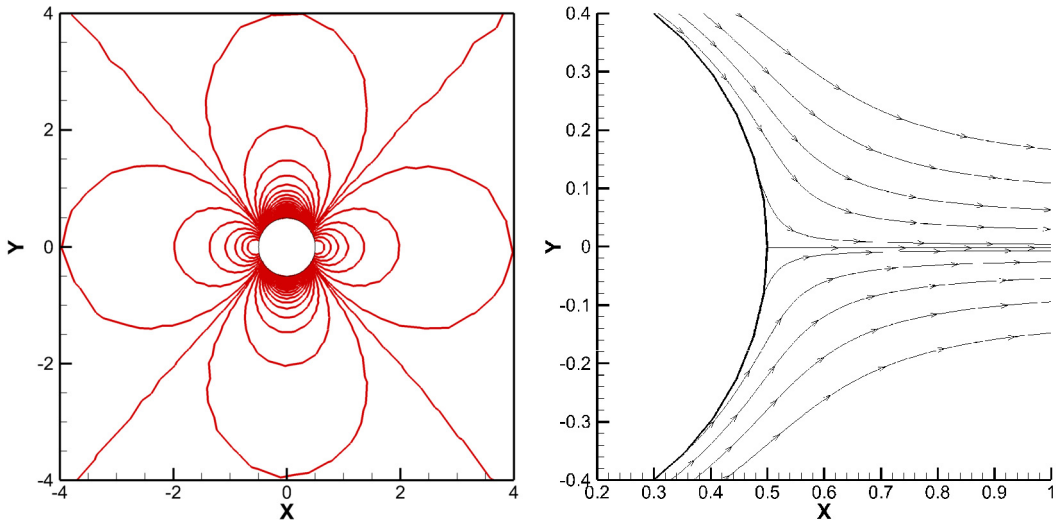


Fig. 16. Pressure contours (left) and streamlines (right) given by the high-order LSFD-FV method.

within the control cell instead of constructing the modified Taylor series expansion for its application on the current and neighboring cells. This provides the high-order LSFD-FV method with the advantage of straightforward algorithm and easy comprehension. For simulating incompressible flow problems, the LBFS is used to evaluate the inviscid and viscous fluxes simultaneously without introducing new degrees of freedom, which makes the LSFD-FV method competitive in terms of the viscous discretization. Moreover, in comparison with existing high-order solvers, the present high-order method is simpler and easier for implementation.

Representative simulations validate the high order accuracy and high computational efficiency of present LSFD-FV method as well as its capability of handling the problems with curve boundary on unstructured grids. Numerical results show that, at the same mesh resolution, the high-order LSFD-FV method has smaller absolute errors and better efficiency compared with the traditional k-exact FV method of the same order of accuracy. In addition, compared with the second-order LSFD-FV method, the high-order method can obtain more precise results with less computational cost. Such outcomes indicate that the high-order LSFD-FV method with LBFS is very promising in practical problems of engineering interest. Since this high-order LSFD-FV method can be extended to three-dimensional simulation straightforwardly, the application for three-dimensional incompressible flow problems will be presented in the future work.

Declaration of competing interest

We declare that we do not have relevant interests to disclose.

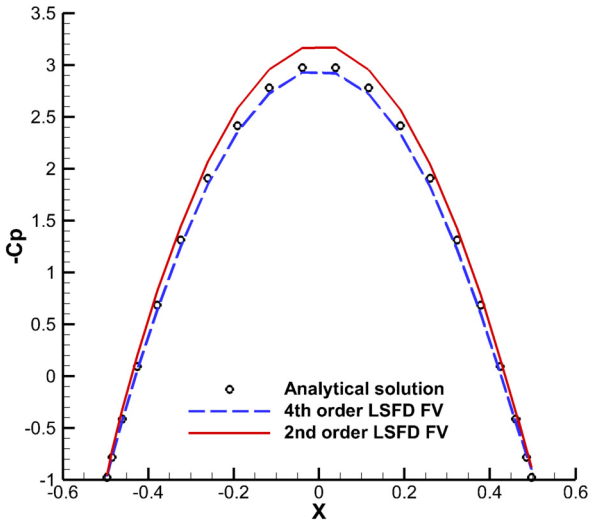


Fig. 17. Comparison of pressure coefficients given by the high- and second-order LSFD-FV methods.

Appendix A. Computation of weighting coefficient matrix *W*

As reported in [36], the local scaling technique and least square optimization were used to prevent the coefficient matrix $\boldsymbol{S}$ in Eq. (10) from being ill-conditioned. By applying Eq. (6) at N ($N > 9$) neighboring points, a least square optimal approximation of the derivative vector in succinct matrix form is obtained,

$$\boldsymbol{S}d\boldsymbol{U} = \Delta\boldsymbol{U}, \tag{A.1}$$

where the matrix $\boldsymbol{S}$ and $\Delta\boldsymbol{U}$ are

$$\begin{aligned}
&\boldsymbol{S}^T = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N],\\
&\boldsymbol{s}_j = \left[\Delta x_j, \Delta y_j, \frac{\Delta x_j^2}{2}, \frac{\Delta y_j^2}{2}, \Delta x_j \Delta y_j, \frac{\Delta x_j^3}{6}, \frac{\Delta y_j^3}{6}, \frac{\Delta x_j^2 \Delta y_j}{2}, \frac{\Delta y_j^2 \Delta x_j}{2}\right], \quad j = 1, 2, \ldots, N,\\
&\Delta\boldsymbol{U}^T = [\boldsymbol{U}_1 - \boldsymbol{U}_i, \boldsymbol{U}_2 - \boldsymbol{U}_i, \ldots, \quad \boldsymbol{U}_j - \boldsymbol{U}_i].
\end{aligned} \tag{A.2}$$

Note that $\boldsymbol{S}$ is not a square matrix but a $N \times 9$ matrix.

For Eq. (A.1), applying the local scaling technique by scaling the local distance $(\Delta x, \Delta y)$ with the radius of the support domain d_0. Eq. (A.1) is then written as

$$\boldsymbol{S}\boldsymbol{D}\boldsymbol{D}^{-1}d\boldsymbol{U} = \overline{\boldsymbol{S}}\boldsymbol{D}^{-1}d\boldsymbol{U} = \Delta\boldsymbol{U}, \tag{A.3}$$

with

$$\begin{aligned}
&\overline{\boldsymbol{S}}^T = [\bar{\boldsymbol{s}}_1, \bar{\boldsymbol{s}}_2, \ldots, \bar{\boldsymbol{s}}_N],\\
&\bar{\boldsymbol{s}}_j = \left[\Delta\bar{x}_j, \Delta\bar{y}_j, \frac{\Delta\bar{x}_j^2}{2}, \frac{\Delta\bar{y}_j^2}{2}, \Delta\bar{x}_j \Delta\bar{y}_j, \frac{\Delta\bar{x}_j^3}{6}, \frac{\Delta\bar{y}_j^3}{6}, \frac{\Delta\bar{x}_j^2 \Delta\bar{y}_j}{2}, \frac{\Delta\bar{y}_j^2 \Delta\bar{x}_j}{2}\right], \quad j = 1, 2, \ldots, N,
\end{aligned} \tag{A.4}$$

where $(\Delta\bar{x}, \Delta\bar{y}) = (\Delta x/d_0, \Delta y/d_0)$ and the scaling matrix $\boldsymbol{D}$ is given as

$$\boldsymbol{D} = \begin{pmatrix}
d_0^{-1} & & & & & & & & \\
& d_0^{-1} & & & & & & & \\
& & d_0^{-2} & & & & & & \\
& & & d_0^{-2} & & & & & \\
& & & & d_0^{-2} & & & & \\
& & & & & d_0^{-3} & & & \\
& & & & & & d_0^{-3} & & \\
& & & & & & & d_0^{-3} & \\
& & & & & & & & d_0^{-3}
\end{pmatrix}. \tag{A.5}$$

As a result, the condition number of resultant scaled coefficient matrix $\bar{\boldsymbol{S}}$ can be improved greatly. The solution of Eq. (A.3) gives,

$$d\boldsymbol{U} = \boldsymbol{D}\left(\overline{\boldsymbol{S}}^T\overline{\boldsymbol{S}}\right)^{-1}\overline{\boldsymbol{S}}^T\Delta\boldsymbol{U}. \tag{A.6}$$

On the other hand, as shown in [36], the distance-related weighting functions are introduced to reduce the influence of data farther from the reference point. In this work, the weighting function used is $W_{ej} = 1/\sqrt{\Delta x_j^2 + \Delta y_j^2}$. By applying the weighting function at N supporting points, the $N \times N$ diagonal matrix $\boldsymbol{W}_e$ is formed as

$$\boldsymbol{W}_e = \begin{pmatrix} W_{e1} & & & \\ & \cdot & & \\ & & \cdot & \\ & & & W_{eN} \end{pmatrix}. \tag{A.7}$$

Then the derivatives are calculated by

$$d\boldsymbol{U} = \boldsymbol{D}\left(\overline{\boldsymbol{S}}^T\boldsymbol{W}_e\overline{\boldsymbol{S}}\right)^{-1}\overline{\boldsymbol{S}}^T\boldsymbol{W}_e\Delta\boldsymbol{U} = \boldsymbol{W}\Delta\boldsymbol{U}, \tag{A.8}$$

Thus, the weighting coefficient matrix $\boldsymbol{W}$ in Eq. (12) is

$$\boldsymbol{W} = \boldsymbol{D}\left(\overline{\boldsymbol{S}}^T\boldsymbol{W}_e\overline{\boldsymbol{S}}\right)^{-1}\overline{\boldsymbol{S}}^T\boldsymbol{W}_e. \tag{A.9}$$

References

[1] C. Hu, C.W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.
[2] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.
[3] M. Dumbser, U. Iben, C.D. Munz, Efficient implementation of high order unstructured WENO schemes for cavitating flows, Comput. Fluids 86 (2013) 141–168.
[4] O. Friedrich, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, J. Comput. Phys. 144 (1998) 194–212.
[5] T.J. Barth, P. Frederichson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, in: 28th Aerospace Sciences Meeting, vol. 90, 1990, 0013.
[6] T.J. Barth, Recent Developments in High-Order k-Exact Reconstruction on Unstructured Meshes, AIAA Paper, 93-0668, 1993.
[7] T. Wang, T.G. Liu, A consistent fourth-order compact finite difference scheme for solving vorticity-stream function form of incompressible Navier-Stokes equations, Numer. Math. Theor. Methods Appl. 12 (2019) 312–330.
[8] Y.L. Liu, W.W. Zhang, C.N. Li, A novel multi-dimensional limiter for high-order finite volume methods on unstructured grids, Commun. Comput. Phys. 22 (2017) 1385–1412.
[9] C.F. Ollivier-Gooch, Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-square reconstruction, J. Comput. Phys. 133 (1997) 6–17.
[10] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, J. Comput. Phys. 181 (2002) 729–752.
[11] L. Cueto-Felgueroso, I. Colominas, X. Nogueira, F. Navarrina, M. Casteleiro, Finite volume solvers and moving least-squares approximations for the compressible Navier-Stokes equations on unstructured grids, Comput. Methods Appl. Mech. Eng. 196 (2007) 4712–4736.
[12] L. Cueto-Felgueroso, I. Colominas, High-order finite volume methods and multiresolution reproducing kernels, Arch. Comput. Methods Eng. 15 (2008) 185–228.
[13] X. Nogueira, L. Cueto-Felgueroso, I. Colominas, H. Gómez, F. Navarrina, M. Casteleiro, On the accuracy of finite volume and discontinuous Galerkin discretizations for compressible flow on unstructured grids, Int. J. Numer. Methods Eng. 78 (2009) 1553–1584.
[14] Q. Wang, Y.X. Ren, W. Li, Compact high order finite volume method on unstructured grids II: extension to two-dimensional Euler equations, J. Comput. Phys. 314 (2016) 883–908.
[15] Q. Wang, Y.X. Ren, J. Pan, W. Li, Compact high order finite volume method on unstructured grids III: variational reconstruction, J. Comput. Phys. 337 (2017) 1–26.
[16] Y. Liu, W. Zhang, Y. Jiang, Z. Ye, A high-order finite volume method on unstructured grids using RBF reconstruction, Comput. Math. Appl. 72 (2016) 1096–1117.
[17] L. Pan, K. Xu, A third-order compact gas-kinetic scheme on unstructured meshes for compressible Navier-Stokes solutions, J. Comput. Phys. 318 (2016) 327–348.
[18] F. Bassi, S. Rebay, A higher-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, J. Comput. Phys. 131 (1997) 267–279.
[19] F. Bassi, S. Rebay, A higher-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138 (1997) 251–285.
[20] F. Zhang, H.S. Tang, J. Cheng, T.G. Liu, A simplified artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier-Stokes equations, Commun. Comput. Phys. 25 (2019) 988–1009.
[21] J.L. Lou, X.D. Liu, H. Luo, H. Nishikawa, Reconstructed discontinuous Galerkin methods for hyperbolic diffusion equations on unstructured grids, Commun. Comput. Phys. 25 (2019) 1302–1327.
[22] B. Cockburn, C.W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (2001) 173–261.
[23] X.L. Liu, F.Y. Song, C.J. Xu, An efficient spectral method for the inextensible immersed interface in incompressible flows, Commun. Comput. Phys. 25 (2019) 1071–1096.
[24] Z.J. Wang, Y. Liu, G. May, A. Jameson, Spectral difference method for unstructured grids II: extension to the Euler equations, J. Sci. Comput. 32 (2007) 45–71.

[25] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two dimensional systems, J. Comput. Phys. 194 (2004) 716–741.
[26] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids VI: extension to viscous flow, J. Comput. Phys. 215 (2006) 41–58.
[27] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210–251.
[28] Z.J. Wang, H. Gao, A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids, J. Comput. Phys. 228 (2009) 8161–8186.
[29] H. Gao, Z.J. Wang, A conservative correction procedure via reconstruction formulation with the Chain-Rule divergence evaluation, J. Comput. Phys. 232 (2013) 7–13.
[30] B. Cockburn, G. Kanschat, I. Perugia, D. Schötzau, Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids, SIAM J. Numer. Anal. 39 (2011) 264–285.
[31] B. Cockburn, G.E. Karniadakis, C.W. Shu, Discontinuous Galerkin methods, in: Theory, Computation and Applications, in: Lecture Notes in Computational Science and Engineering, vol. 11, Springer, 2000.
[32] R. Hartmann, Numerical analysis of higher order discontinuous Galerkin finite element methods, in: VKI LS, vol. 8, 2008, pp. 13–17.
[33] K. Van den Abeele, G. Ghorbaniasl, M. Parsani, C. Lacor, A stability analysis for the spectral volume method on tetrahedral grids, J. Comput. Phys. 228 (2009) 257–265.
[34] K. Van den Abeele, C. Lacor, An accuracy and stability study of the 2D spectral volume method, J. Comput. Phys. 226 (2007) 1007–1026.
[35] K. Van den Abeele, C. Lacor, Z.J. Wang, On the connection between the spectral volume and the spectral difference method, J. Comput. Phys. 227 (2007) 877–885.
[36] H. Ding, C. Shu, K.S. Yeo, Development of least square-based two-dimensional finite difference schemes and their application to simulate natural convection in a cavity, Comput. Fluids 33 (2004) 137–154.
[37] H. Ding, C. Shu, K.S. Yeo, D. Xu, Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method, Comput. Methods Appl. Mech. Eng. 193 (2004) 727–744.
[38] C. Shu, W.X. Wu, H. Ding, C.M. Wang, Free vibration analysis of plates using least-square-based finite difference method, Comput. Methods Appl. Mech. Eng. 196 (2007) 1330–1343.
[39] H. Ding, C. Shu, K.S. Yeo, D. Xu, Numerical simulation of flows around two circular cylinders by mesh-free least square-based finite difference methods, Int. J. Numer. Methods Fluids 53 (2007) 305–332.
[40] W.X. Wu, C. Shu, C.M. Wang, Mesh-free least-squares-based finite difference method for large-amplitude free vibration analysis of arbitrarily shaped thin plates, J. Sound Vib. 317 (2008) 955–974.
[41] J.J. Douglas, T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods, Lecture Notes in Phys., vol. 58, Springer, Berlin, 1976.
[42] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, M. Savini, A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows, in: Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Antwerpen, Belgium, 1997, pp. 99–108.
[43] B. Cockburn, C.W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[44] C. Shu, Y. Wang, C.J. Teo, J. Wu, Development of lattice Boltzmann flux solver for simulation of incompressible flows, Adv. Appl. Math. Mech. 6 (2014) 436–460.
[45] Z. Guo, C. Shu, Lattice Boltzmann method and its applications in engineering, World Scientific, 2013.
[46] Y. Wang, C. Shu, L.M. Yang, An improved multiphase lattice Boltzmann flux solver for three-dimensional flows with large density ratio and high Reynolds number, J. Comput. Phys. 302 (2015) 41–58.
[47] Y. Wang, C. Shu, C.J. Teo, Development of LBGK and incompressible LBGK-based lattice Boltzmann flux solvers for simulation of incompressible flows, Int. J. Numer. Methods Fluids 75 (2014) 344–364.
[48] L.M. Yang, C. Shu, W.M. Yang, Y. Wang, A simplified circular function-based gas kinetic scheme for simulation of incompressible flows, Int. J. Numer. Methods Fluids 85 (2017) 583–598.
[49] Y. Sun, C. Shu, C.J. Teo, Y. Wang, L.M. Yang, Explicit formulations of gas-kinetic flux solver for simulation of incompressible and compressible viscous flows, J. Comput. Phys. 300 (2015) 492–519.
[50] Y. Wang, C. Shu, C.J. Teo, Thermal lattice Boltzmann flux solver and its application for simulation of incompressible thermal flows, Comput. Fluids 94 (2014) 98–111.
[51] A. Jameson, Time dependent calculations using multigrid with applications to unsteady flows past airfoils and wings, in: AIAA 10th Comput. Fluid Dyn. Conf., Honolulu, Hawaii, 1991, 1596.
[52] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[53] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.
[54] N. Pellerin, S. Leclaire, M. Reggio, Solving incompressible fluid flows on unstructured meshes with the lattice Boltzmann flux solver, Eng. Appl. Comput. Fluid Mech. 11 (2017) 310–327.
[55] S.C.R. Dennis, G.Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, J. Fluid Mech. 42 (1970) 471–489.
[56] R.K. Shukla, M. Tatineni, X. Zhong, Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier-Stokes equations, J. Comput. Phys. 224 (2007) 1064–1094.
[57] J. Wu, C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys. 228 (2009) 1963–1979.
[58] X. He, G. Doolen, Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder, J. Comput. Phys. 134 (1997) 306–315.
[59] M. Braza, P. Chassaing, H.H. Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, J. Fluid Mech. 165 (1986) 79–130.
[60] C. Liu, X. Zheng, C.H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, J. Comput. Phys. 139 (1998) 35–57.