Efficient high-order radial basis-function-based differential quadrature–finite volume method for incompressible flows on unstructured grids

Y. Y. Liu ●,[1] L. M. Yang,[2] C. Shu ●,[1,*] and H. W. Zhang[1]

[1]*Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260*
[2]*Department of Aerodynamics, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Yudao Street, Nanjing 210016, Jiangsu, China*

This paper presents an efficient high-order radial basis-function-based differential quadrature–finite volume method for incompressible flows on unstructured grids. In this method, a high-order polynomial based on the Taylor series expansion is applied within each control cell to approximate the solution. The derivatives in the Taylor series expansion are approximated by the mesh-free radial basis-function-based differential quadrature method. The recently proposed lattice Boltzmann flux solver is applied to simultaneously evaluate the inviscid and viscous fluxes at the cell interface by the local solution of the lattice Boltzmann equation. In the present high-order method, a premultiplied coefficient matrix appears in the time-dependent term, reflecting the implicit nature. The implicit time-marching techniques, i.e., the lower-upper symmetric Gauss-Seidel and the explicit first stage, singly diagonally implicit Runge-Kutta schemes, are incorporated to efficiently solve the resultant ordinary differential equations. Several numerical examples are tested to validate the accuracy, efficiency, and robustness of the present method on unstructured grids. Compared with the *k*-exact method, the present method enjoys higher accuracy and better computational efficiency.

I. INTRODUCTION

High-order numerical methods have drawn increasing attention in the computational fluid dynamics (CFD) community due to their high-order accuracy, low dissipation and dispersion, and high computational efficiency. They have shown better performance in predicting turbulence, aeroacoustics, and vortex-dominant flows than the conventional second-order methods [1–5]. Over the past three decades, various high-order methods have been developed. Among them, the high-order methods on the unstructured grids or overset meshes are active research topics since they can handle both complex physics and configurations. Typical examples in the literature include finite volume (FV) methods [1,2,4,6–14], finite difference methods using overset meshes [5,15,16], discontinuous Galerkin (DG) methods [17–21], spectral volume (SV) [22–24] methods, spectral difference (SD) [25,26] methods, correction procedure via reconstruction (CPR) method [27,28], etc.

This paper proposes a high-order FV method on unstructured grids for incompressible flows that addresses several challenging issues of the high-order FV methods, i.e., the high-order approximation of solution, viscous discretization, and time-marching approach. For the high-order spatial discretization, the essence lies in the high-order representation of flow variables within the control cell. One pioneering work is the *k*-exact method which was first proposed by Barth and Frederickson [10] and later was extensively applied in

computational fluid dynamics on unstructured grids [11,12]. The *k*-exact method is based on a modified Taylor series expansion which is determined by the mean values of the solution function at neighboring cells. In the process, the modified Taylor series expansion needs to be integrated over the current cell and its neighboring cells, respectively, resulting in an overdetermined equation system to compute the unknown coefficients by the least-squares technique. Apart from such an approximation approach based on the high-order polynomial, some mesh-free spatial approximation methods such as the moving least-squares (MLS) method [13] and radial basis function (RBF) method [14] have been applied to advance the development of unstructured high-order FV methods.

In this paper, a high-order approximation technique based on the mesh-free radial basis-function-based differential quadrature (RBFDQ) method [29] is introduced. We directly apply a Taylor series expansion within each control cell to approximate the solution with high-order accuracy. The unknown coefficients in the approximation polynomial, i.e., the spatial derivatives, are determined by the RBFDQ method using the solutions at centers of the current cell and its neighboring cells. The RBFDQ method [29] is a mesh-free method for derivative approximation at scattered points. It has been applied to various flow problems [30–32] owing to its merits such as high efficiency and good convergence. However, the existing applications of the RBFDQ method are limited to the finite difference (FD) framework whose numerical discretization cannot be guaranteed to be conservative. The FV discretization, unlike the FD method, can remain conservative at the cell and the global level. The foregoing facts suggest the incorporation of the RBFDQ method into the framework

*mpeshuc@nus.edu.sg

of the FV method, which provides a simple and effective high-order solver on unstructured grids, i.e., the RBFDQ-FV method. Compared with the k-exact method, the direct adoption of the Taylor series expansion as the approximation function in the RBFDQ-FV method is more straightforward and it is easier for the boundary treatment since no integral is involved. In addition, such a polynomial approximation based on the Taylor series expansion allows easier control of the order of accuracy than some function approximation methods.

The development of a high-order FV method on unstructured grids also involves the evaluation of numerical fluxes across the interface. In fact, it is a critical issue which significantly affects not only the accuracy and stability of the numerical method but also its computational efficiency. Normally, the inviscid and viscous fluxes are, respectively, evaluated by the Riemann solver and average approach. This seems to be simple and easy to implement. However, the average approach just takes a simple arithmetic average of the gradients from the left and right cells of the interface without considering a possible solution jump. Thus, this treatment is inconsistent and may result in an incorrect solution [33]. Some studies have been conducted to evaluate the viscous fluxes properly, such as the interior penalty approach [34], the local DG approach [35], and the second Bassi-Rebay scheme [3,36]. Nevertheless, continuous efforts are required to further reduce the computational cost and improve the convergence. This work adopts a physical way to evaluate the inviscid and viscous fluxes simultaneously by the lattice Boltzmann flux solver (LBFS) [37]. The LBFS utilizes the local reconstruction of the lattice Boltzmann solution to construct the numerical flux at the cell interface. Through the Chapman-Enskog (CE) expansion analysis, the lattice Boltzmann solution satisfies the Navier-Stokes (NS) equations. This means that the fluxes computed by the LBFS can satisfy the NS equations automatically. Furthermore, we apply the LBFS locally at the cell interface and thus the whole high-order FV method can be implemented easily on unstructured grids.

Another key ingredient of the high-order FV method is the time integration approach. Since the high-order methods are generally more complicated and stiffer than the conventional second-order methods, more attention should be paid to the time-marching strategy [38–41]. In this work, when the approximation polynomial is substituted into the volume integral over the control cell, a premultiplied coefficient matrix appears in the time-dependent term. This matrix implies the connection between the functional values at cell centers of the current cell and its neighboring cells, which reflects the implicit nature of the RBFDQ-FV method. In fact, the point iterative method could be used to obtain the steady-state solution and the dual time-stepping method can be incorporated for computing the time-accurate solution. Such an explicit time-marching strategy is simple but not efficient. In order to take full advantage of the implicit nature, we combine the coefficient matrix with the implicit lower-upper symmetric Gauss-Seidel (LU-SGS) scheme [38,42–44] and the explicit first stage, singly diagonally implicit Runge-Kutta (ESDIRK) method [39] as the time-marching strategy. As a result, the computational efficiency of the present high-order method is greatly improved. In order to validate the accuracy, computa-tional efficiency, and robustness of the current high-order FV method, a set of numerical cases for incompressible flows is studied.

The rest of this paper is organized as follows. In Sec. II, the high-order RBFDQ-FV method with LBFS for incompressible flows is described in detail. Then a series of incompressible benchmark tests is carried out in Sec. III to assess the performance of the proposed method on unstructured grids. We end this paper with some conclusions in Sec. IV.

II. HIGH-ORDER RADIAL BASIS-FUNCTION-BASED DIFFERENTIAL QUADRATURE–FINITE VOLUME (RBFDQ-FV) METHOD

A. Governing equations and high-order finite volume discretization

As stated in the Introduction, the LBFS [37] would be utilized to simultaneously evaluate the viscous and inviscid fluxes at the cell interface. In such a weakly compressible model, the incompressible flows can be simulated via the compressible NS equations under a low Mach number limit (Ma < 0.3) with explicit evaluation of pressure from density. For two-dimensional (2D) cases, the governing equations recovered by the lattice Boltzmann equation (LBE) in the continuum flow regime through the multiscale CE expansion analysis can be cast in the Cartesian coordinate system as

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

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

where ρ denotes the density; $\boldsymbol{u} = (u, v)$ is the velocity vector; p and ν are, respectively, the pressure and kinematic viscosity of the fluid flow. $\boldsymbol{I}$ is the unit tensor. Eqs. (1) and (2) can be rewritten in a conservative form as

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

where $\boldsymbol{U}$ is the vector of conservative variables and $\boldsymbol{F}$ represents the vector of fluxes. By incorporating the divergence theorem and Gaussian quadrature, the integral form of Eq. (3) over a control cell Ω_i is semidiscretized as

$$\frac{d}{dt}\left(\int_{\Omega_i} \boldsymbol{U} d\Omega\right) = -\sum_{j=1}^{n_f} \sum_{k=1}^{n_{GQp}} (\boldsymbol{F}_{n,k} A)_j w_k, \qquad (4)$$

where n_f is the number of cell edges of the cell Ω_i, n_{GQp} is the number of Gaussian quadrature points on each edge, and $n_{GQp} = 2$ in this work. A is the interface area and w denotes a quadrature weight. $\boldsymbol{F}_n = \boldsymbol{F} \cdot \boldsymbol{n}$ and $\boldsymbol{n}$ represents the outward unit vector normal to the cell edge.

In this work, the following high-order polynomial based on the Taylor series expansion to the cell center of the control cell Ω_i is used to approximate the functional value with

fourth-order accuracy.

$$
\boldsymbol{U}(x, y) = \boldsymbol{U}_i + \frac{\partial \boldsymbol{U}}{\partial x}\bigg|_i (x - x_i) + \frac{\partial \boldsymbol{U}}{\partial y}\bigg|_i (y - y_i) + \frac{\partial^2 \boldsymbol{U}}{\partial x^2}\bigg|_i \frac{(x - x_i)^2}{2} + \frac{\partial^2 \boldsymbol{U}}{\partial y^2}\bigg|_i \frac{(y - y_i)^2}{2} + \frac{\partial^2 \boldsymbol{U}}{\partial x \partial y}\bigg|_i (x - x_i)(y - y_i)
$$
$$
+ \frac{\partial^3 \boldsymbol{U}}{\partial x^3}\bigg|_i \frac{(x - x_i)^3}{6} + \frac{\partial^3 \boldsymbol{U}}{\partial y^3}\bigg|_i \frac{(y - y_i)^3}{6} + \frac{\partial^3 \boldsymbol{U}}{\partial x^2 \partial y}\bigg|_i \frac{(x - x_i)^2 (y - y_i)}{2} + \frac{\partial^3 \boldsymbol{U}}{\partial y^2 \partial x}\bigg|_i \frac{(y - y_i)^2 (x - x_i)}{2}, \tag{5}
$$

where (x_i, y_i) is the location of the cell center of Ω_i and $\boldsymbol{U}_i$ is the corresponding function value. Integrating Eq. (5) over the control cell Ω_i gives

$$
\int_{\Omega_i} \boldsymbol{U}(x, y) d\Omega = \Omega_i \boldsymbol{U}_i + d\boldsymbol{U}_i^T \boldsymbol{C}_i, \tag{6}
$$

with

$$
d\boldsymbol{U}_i^T = \left[\frac{\partial \boldsymbol{U}}{\partial x}\bigg|_i, \; \frac{\partial \boldsymbol{U}}{\partial y}\bigg|_i, \; \frac{\partial^2 \boldsymbol{U}}{\partial x^2}\bigg|_i, \; \frac{\partial^2 \boldsymbol{U}}{\partial y^2}\bigg|_i, \; \frac{\partial^2 \boldsymbol{U}}{\partial x \partial y}\bigg|_i, \; \frac{\partial^3 \boldsymbol{U}}{\partial x^3}\bigg|_i, \; \frac{\partial^3 \boldsymbol{U}}{\partial y^3}\bigg|_i, \; \frac{\partial^3 \boldsymbol{U}}{\partial x^2 \partial y}\bigg|_i, \; \frac{\partial^3 \boldsymbol{U}}{\partial y^2 \partial x}\bigg|_i \right],
$$
$$
\boldsymbol{C}_i^T = \left[\overline{\overline{x^1 y_i^0}}, \; \overline{\overline{x^0 y_i^1}}, \; \frac{\overline{\overline{x^2 y_i^0}}}{2}, \; \frac{\overline{\overline{x^0 y_i^2}}}{2}, \; \overline{\overline{x^1 y_i^1}}, \; \frac{\overline{\overline{x^3 y_i^0}}}{6}, \; \frac{\overline{\overline{x^0 y_i^3}}}{6}, \; \frac{\overline{\overline{x^2 y_i^1}}}{2}, \; \frac{\overline{\overline{x^1 y_i^2}}}{2} \right], \tag{7}
$$

where $\overline{\overline{x^n y_i^m}} = \int_{\Omega_i} (x - x_i)^n (y - y_i)^m d\Omega$. The coefficients $\boldsymbol{C}_i$ represent the volume integrals of the distance relations with respect to the cell center; they could vary for different cells in the whole flow domain. Clearly, Eq. (6) involves nine unknown spatial derivatives. They are approximated by the mesh-free RBFDQ method based on the functional values at the centers of the N neighboring cells as follows:

$$
d\boldsymbol{U}_k^i = \sum_{j=1}^{N} W_{k,j}^i \boldsymbol{U}_{ij}, \quad k = 1, \ldots, 9, \tag{8}
$$

where W_k^i denote the corresponding weighting coefficients for the kth spatial derivative $d\boldsymbol{U}_k^i$ and $\boldsymbol{U}_{ij}$ is the function value at the cell center of the jth neighboring cell to the reference cell Ω_i. The details of determining W_k^i by the RBFDQ approach are presented in Sec. II B.

With Eqs. (6) and (8), we can rewrite Eq. (4) as follows:

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

where $-\boldsymbol{R}_i$ is the discrete form of the right-hand side of Eq. (4). In this work, the numerical fluxes in $\boldsymbol{R}_i$ are evaluated by the LBFS, which is described in Sec. II C. When Eq. (9) is applied to all the control cells in the flow domain, we obtain the following matrix form,

$$
\mathbf{M} \frac{d\mathbf{U}}{dt} = -\mathbf{R}, \tag{10}
$$

where $\mathbf{U}$ represents the solution vector composed by all control cells. $\mathbf{M}$ is a sparse coefficient matrix formed by coefficients on the left-hand side of Eq. (9). By solving Eq. (10), the cell-centered solution would be obtained. The detailed time-marching strategy is described in Sec. II D.

B. Derivative approximation by radial basis-function-based differential quadrature (RBFDQ) scheme

In this subsection, the details of derivative approximation by the RBFDQ scheme are presented. In fact, the RBFDQ

method stems from the combination of the function approximation by RBF and derivative approximation by the differential quadrature (DQ) technique [45–47]. Basically, the DQ method approximates the partial derivative by a linear weighted sum of functional values at all discrete points in the supporting domain. Although DQ enjoys appealing merits such as a simple algorithm and high accuracy, its one-dimensional essence restricts its application to problems with complex geometries. The RBFDQ method resolves this difficulty by using the RBF instead of the high-order polynomial as the base function. As a result, the RBFDQ method inherits the advantages of the RBF such as the mesh-free property and multidimensional nature and remains a simple algorithm of the DQ scheme. In the development of the RBFDQ method, the multiquadric (MQ) RBF is commonly used since it outperforms the other RBFs in terms of accuracy and convergence. The expression of the MQ RBF is

$$
\varphi(r) = \sqrt{r^2 + c^2}, \quad c > 0, \tag{11}
$$

where r is the Euclidean distance and the positive constant c denotes the shape parameter. Hereinafter, we will briefly introduce the derivative approximation by the RBFDQ method with MQ in the 2D case and for the detailed derivation, refer to Ref. [29]. Note that other RBFs can be implemented in the same manner.

For illustrative purposes, the approximation of the mth-order derivative of $\boldsymbol{U}(x, y)$ with respect to x at (x_i, y_i) by the DQ method is introduced here, i.e.,

$$
\frac{\partial^m \boldsymbol{U}(x, y)}{\partial x^m}\bigg|_{(x_i, y_i)} = \sum_{j=1}^{N} W_j^{(m)} \boldsymbol{U}_{ij}, \tag{12}
$$

where $W_j^{(m)}$ denotes the corresponding weighting coefficients for the mth-order derivative. Clearly, the key issue of approximating the derivative lies in the calculation of $W_j^{(m)}$. Suppose that $\boldsymbol{U}(x, y)$ is approximated by a set of MQ RBFs locally, i.e.,

$$
\boldsymbol{U}(x, y) \cong \sum_{j=1}^{N} \lambda_j \varphi(\|\boldsymbol{x} - \boldsymbol{x}_j\|_2) = \sum_{j=1}^{N} \lambda_j \sqrt{r_j^2 + c^2}, \tag{13}
$$

where $r_j = \|\boldsymbol{x} - \boldsymbol{x}_j\|_2$ and λ_j is the weight coefficient of each RBF. As analyzed in Ref. [46], if all the MQ base functions $\varphi(\|\boldsymbol{x} - \boldsymbol{x}_j\|_2)$ satisfy the linear equation (12), so does $\boldsymbol{U}(x, y)$. Therefore, $W_j^{(m)}$ in Eq. (12) can be directly computed by substituting these MQ base functions into Eq. (12), i.e.,

$$\frac{\partial^m \varphi_k(x_i, y_i)}{\partial x^m} = \sum_{j=1}^{N} W_j^{(m)} \varphi_k(x_j, y_j), \quad k = 1, 2, \ldots, N, \quad (14)$$

where $\varphi_k(x_j, y_j)$ is the value of the kth MQ RBF at the jth supporting point of (x_i, y_i). The above equations can be further written in the matrix form as

$$\begin{bmatrix} \frac{\partial^m \varphi_1(x_i, y_i)}{\partial x^m} \\ \frac{\partial^m \varphi_2(x_i, y_i)}{\partial x^m} \\ \vdots \\ \frac{\partial^m \varphi_N(x_i, y_i)}{\partial x^m} \end{bmatrix} = \begin{bmatrix} \varphi_1(x_1, y_1) & \varphi_1(x_2, y_2) & \cdots & \varphi_1(x_N, y_N) \\ \varphi_2(x_1, y_1) & \varphi_2(x_2, y_2) & \cdots & \varphi_2(x_N, y_N) \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_N(x_1, y_1) & \varphi_N(x_2, y_2) & \cdots & \varphi_N(x_N, y_N) \end{bmatrix}$$

$$\times \begin{bmatrix} W_1^{(m)} \\ W_2^{(m)} \\ \vdots \\ W_N^{(m)} \end{bmatrix}, \quad (15)$$

or in the vector form,

$$\boldsymbol{g}_{x^m} = \boldsymbol{g} \boldsymbol{W}^m. \quad (16)$$

If the collocation matrix $\boldsymbol{g}$ is nonsingular, a unique solution of Eq. (16) can be obtained. As indicated in Ref. [48], the collocation matrix $\boldsymbol{g}$ is conditionally positive-definite when the MQ RBFs are used. Hence, the coefficient vector $\boldsymbol{W}^m$ can be computed by

$$\boldsymbol{W}^m = \boldsymbol{g}^{-1} \boldsymbol{g}_{x^m}. \quad (17)$$

In the RBFDQ method described above, the shape parameter c is a free parameter which should be specified by the user. This shape parameter significantly affects the accuracy of the derivative approximation. Ding *et al.* [49] studied the relationship between the shape parameter c and the accuracy of the RBFDQ method by numerical experiments. The normalization of the shape parameter was suggested and it worked well in removing the difficulty of a different number of supporting points for each reference point. The specific procedure in the 2D case is to transform the local support to a unit circle, which gives

$$\varphi = \sqrt{\left(\bar{x} - \frac{x_i}{D_i}\right)^2 + \left(\bar{y} - \frac{y_i}{D_i}\right)^2 + \bar{c}^2}, \quad i = 1, \ldots, N, \quad (18)$$

where $(\bar{x}, \bar{y}) = (x/D_i, y/D_i)$; $\bar{c} = c/D_i$; D_i denotes the diameter of the minimal circle enclosing all (N) supporting points in the support region for the point (x_i, y_i). Note that the transformation in Eq. (18) changes the formulation of the weighting coefficients in the RBFDQ method as well. Correspondingly, the derivatives are approximated by the final expression as follows,

$$\left. \frac{\partial^m \boldsymbol{U}(x, y)}{\partial x^m} \right|_{(x_i, y_i)} = \sum_{j=1}^{N} \frac{\overline{W}_j^{(m)}}{D_i} \boldsymbol{U}_{ij}, \quad (19)$$

where $\overline{W}_j^{(m)}$ are the weighting coefficients computed by the RBFDQ method after the normalization.

Up to this point, we have described how to approximate the mth-order derivative with respect to x at (x_i, y_i) by the RBFDQ method. Similarly, other derivatives in Eq. (6) can be computed by calculating the corresponding weighting coefficients. Note that $\boldsymbol{W}^m$ in Eq. (17) would form a $9 \times N$ dimensional weighting coefficient matrix $\boldsymbol{W}$ for the nine spatial derivatives in Eq. (6) and $\boldsymbol{W}$ is uniquely determined by the coordinates of the mesh points (centers of neighboring cells of the current cell). In practical implementation, $\boldsymbol{W}$ is calculated once and stored when the computational mesh is fixed to save computational effort.

C. Evaluation of numerical fluxes by lattice Boltzmann flux solver

This subsection is dedicated to the illustration of the accurate evaluation of numerical fluxes at the cell interface in the high-order RBFDQ-FV method. As discussed in the Introduction, the LBFS [37] is adopted to evaluate the inviscid and viscous fluxes simultaneously. In practical application, a local coordinate system is introduced at the cell interface which provides the present RBFDQ-FV method with flexibility on unstructured grids. The outward normal direction and tangential direction of the cell interface—which are, respectively, denoted by the subscripts "1" and "2"—are used in the local coordinate system. Some details of the LBFS in the local coordinate system are discussed below. For the development of the LBFS and its difference from the conventional NS solvers and the lattice Boltzmann method (LBM) see Refs. [37,50,51].

Following the work of Wang *et al.* [37], Eqs. (1) and (2) can be recovered through multiscale CE expansion analysis, which gives the expression of the fluxes as follows,

$$F_1 = \sum_{\alpha=0}^{N_d} (\boldsymbol{e}_\alpha)_1 f_\alpha^{eq},$$

$$F_2 = \sum_{\alpha=0}^{N_d} (\boldsymbol{e}_\alpha)_1 (\boldsymbol{e}_\alpha)_1 \left[f_\alpha^{eq} + \left(1 - \frac{1}{2\tau_\nu}\right) f_\alpha^{neq} \right], \quad (20)$$

$$F_3 = \sum_{\alpha=0}^{N_d} (\boldsymbol{e}_\alpha)_1 (\boldsymbol{e}_\alpha)_2 \left[f_\alpha^{eq} + \left(1 - \frac{1}{2\tau_\nu}\right) f_\alpha^{neq} \right],$$

where $\boldsymbol{e}_\alpha$ is the lattice velocity vector and τ_ν is the single relaxation parameter. f_α^{eq} and f_α^{neq}, respectively, denote the equilibrium and nonequilibrium density distribution functions along the α direction. $N_d + 1$ is the number of discrete particle velocities in the lattice velocity model. In the present study, the commonly used D2Q9 lattice velocity model ($N_d + 1 = 9$) is applied and for the corresponding expression of f_α^{eq} refer to Ref. [37]. τ_ν in Eq. (20) is determined from the kinematic viscosity ν via

$$\nu = (\tau_\nu - 1/2) c_s^2 \delta_t, \quad (21)$$

where δ_t is the streaming time step. c_s is the sound speed and $c_s = 1/\sqrt{3}$ if the D2Q9 model is used [37]. The pressure can be calculated from the equation of state $p = \rho c_s^2$.

The nonequilibrium distribution function f_α^{eq} is approximated by

$$f_\alpha^{neq}(\boldsymbol{r}, t) = -\tau_\nu \left[f_\alpha^{eq}(\boldsymbol{r}, t) - f_\alpha^{eq}(\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t, t - \delta_t) \right] + O(\delta_t^2), \tag{22}$$

where $\boldsymbol{r}$ is the physical location of the Gaussian quadrature point along the cell interface and t is the time. $f_\alpha^{eq}(\boldsymbol{r}, t)$ and $f_\alpha^{eq}(\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t, t - \delta_t)$ are the equilibrium density distribution functions at $\boldsymbol{r}$ and its surrounding nodes $\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t$, respectively.

The calculation of $f_\alpha^{eq}(\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t, t - \delta_t)$ requires the flow variables at the location $\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t$. In this work, they are interpolated by Eq. (5) from the functional values at cell centroids $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ with fourth-order accuracy, i.e.,

$$\boldsymbol{U}(\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t) = \begin{cases} \boldsymbol{U}(\boldsymbol{r}_i) + \nabla \boldsymbol{U}(\boldsymbol{r}_i)\Delta \boldsymbol{X}_i + \frac{1}{2}\Delta \boldsymbol{X}_i^T \boldsymbol{H}(\boldsymbol{X}_i)\Delta \boldsymbol{X}_i + \frac{1}{6}\Delta^2 \boldsymbol{X}_i^T \boldsymbol{G}(\boldsymbol{X}_i)\Delta \boldsymbol{X}_i + O(\Delta \boldsymbol{X}^4), & (\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t) \in \Omega_i \\ \boldsymbol{U}(\boldsymbol{r}_j) + \nabla \boldsymbol{U}(\boldsymbol{r}_j)\Delta \boldsymbol{X}_j + \frac{1}{2}\Delta \boldsymbol{X}_j^T \boldsymbol{H}(\boldsymbol{X}_j)\Delta \boldsymbol{X}_j + \frac{1}{6}\Delta^2 \boldsymbol{X}_j^T \boldsymbol{G}(\boldsymbol{X}_j)\Delta \boldsymbol{X}_j + O(\Delta \boldsymbol{X}^4), & (\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t) \in \Omega_j \end{cases}, \tag{23}$$

with

$$\Delta \boldsymbol{X}_k = [(\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t - \boldsymbol{r}_k)_x, (\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t - \boldsymbol{r}_k)_y]^T, \quad k = i \text{ or } j. \tag{24}$$

$\nabla \boldsymbol{U}, \boldsymbol{H}$, and $\boldsymbol{G}$ are given by

$$\nabla \boldsymbol{U} = \left(\frac{\partial \boldsymbol{U}}{\partial x}, \frac{\partial \boldsymbol{U}}{\partial y} \right),$$

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial^2 \boldsymbol{U}}{\partial x^2} & \frac{\partial^2 \boldsymbol{U}}{\partial x \partial y} \\ \frac{\partial^2 \boldsymbol{U}}{\partial y \partial x} & \frac{\partial^2 \boldsymbol{U}}{\partial y^2} \end{bmatrix}, \tag{25}$$

$$\boldsymbol{G} = \begin{bmatrix} \frac{\partial^3 \boldsymbol{U}}{\partial x^3} & 3\frac{\partial^3 \boldsymbol{U}}{\partial x^2 \partial y} \\ 3\frac{\partial^3 \boldsymbol{U}}{\partial y^2 \partial x} & \frac{\partial^3 \boldsymbol{U}}{\partial y^3} \end{bmatrix},$$

where the derivatives are approximated by the RBFDQ method as introduced in Sec. II B. Once the required flow variables are available, $f_\alpha^{eq}(\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t, t - \delta_t)$ in Eq. (22) can be obtained by the definition of the equilibrium distribution function. In addition, following the derivations in Ref. [37], the density and velocities at $\boldsymbol{r}$ can be computed by $f_\alpha^{eq}(\boldsymbol{r} - \boldsymbol{e}_\alpha \delta_t, t - \delta_t)$ and then we can calculate $f_\alpha^{eq}(\boldsymbol{r}, t)$. After getting $f_\alpha^{neq}(\boldsymbol{r}, t)$, the flux vector $\boldsymbol{F} = (F_1, F_2, F_3)^T$ in Eq. (20) can be obtained.

Note that, as stated previously, the whole flux $\boldsymbol{F}$ is computed in the local coordinate system by the LBFS, while the flux $\boldsymbol{F}_n$ in Eq. (4) is defined in the global coordinate system. In the practical calculation, the following transformation should be adopted.

$$\boldsymbol{F}_n = (F_1, F_2 n_{1x} + F_3 n_{2x}, F_2 n_{1y} + F_3 n_{2y})^T, \tag{26}$$

where $\boldsymbol{n}_1 = (n_{1x}, n_{1y})$ is the unit vector in direction 1 of the local coordinate system. $\boldsymbol{n}_2 = (n_{2x}, n_{2y})$ is the unit vector in direction 2.

Up to this point, the LBFS and its implementation in the present high-order RBFDQ-FV method have been presented. Generally, the LBFS is applied to the incompressible viscous flows with second-order accuracy. One may naturally doubt that the second-order accuracy of the LBFS may spoil the accuracy of the present high-order method. To resolve this question, we should note that, in the LBFS, the second-order accuracy is with regards to δ_t locally, as shown in Eq. (22), rather than the mesh spacing h which is used to determine the accuracy of a numerical method [52]. If the physical streaming time step δ_t is much smaller than h, the overall accuracy of the present RBFDQ-FV method would not be affected significantly. This analysis will be further verified in

Sec. III A with the corresponding accuracy test. Furthermore, since the reconstruction in the LBFS is conducted physically and locally at every Gaussian quadrature point along each cell interface, different δ_t could be chosen for different interfaces, which provides great flexibility for application on unstructured grids. Meanwhile, due to the independence between the time-marching step Δt and the streaming time step δ_t (δ_t is only used in the solution reconstruction), the choice of small δ_t would not affect the genuine time evolution process in the present method. Basically, to avoid extrapolation, the discrete particles and points in the physical space constructed around the Gaussian quadrature point must be within the right and left cells of the interface. Given this limitation, in the 2D case, δ_t is locally determined by

$$\delta_t \leqslant \left(\frac{1}{2} - \frac{1}{2\sqrt{3}} \right) h_{\min}, \tag{27}$$

where $h_{\min}$ is the minimum edge length of the left and right cells of the interface for a quadrilateral mesh or the radius of the inscribed circle in the left and right cells of the interface for a triangular mesh.

D. Efficient implicit time-marching strategy

As discussed in Sec. II A, there exists a premultiplied matrix in the time-dependent term in Eq. (10), which reflects the implicit essence of the present high-order method. To solve Eq. (10) efficiently and entirely consider the implicit nature; the time-marching strategy coupled with the implicit LU-SGS [38,42–44] and ESDIRK [39] schemes is presented in this subsection.

For the steady case, due to the unimportant time accuracy, the widely used implicit LU-SGS is applied and then Eq. (10) is rewritten as

$$\boldsymbol{M}\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t} = -\boldsymbol{R}^{n+1}(\boldsymbol{U}), \tag{28}$$

where Δt is the time step determined by the CFL condition. The superscripts "n" and "$n + 1$" represent the current time level at time t and the new time level at time $t + \Delta t$, respectively. In the calculation, the residual $\boldsymbol{R}^{n+1}$ is linearized by applying a linearized implicit technique [42,53] about the current time level n as

$$\boldsymbol{R}^{n+1} \approx \boldsymbol{R}^n + \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} \right)^n \Delta \boldsymbol{U}^n, \tag{29}$$

where $\Delta \mathbf{U}^n = \mathbf{U}^{n+1} - \mathbf{U}^n$ and $\partial \mathbf{R}/\partial \mathbf{U}$ denotes the flux Jacobian. With Eq. (29), Eq. (28) is rewritten as

$$\left[\frac{\mathbf{M}}{\Delta t} + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^n\right]\Delta \mathbf{U}^n = -\mathbf{R}^n(\mathbf{U}). \tag{30}$$

Apparently, the term in the square bracket on the left-hand side of Eq. (30), i.e., the implicit operator, consists of two parts: the coefficient matrix $\mathbf{M}$ and the flux Jacobian $\partial \mathbf{R}/\partial \mathbf{U}$. In this way, the implicit nature of the present high-order FV method is combined into the implicit operator conveniently. In general, the implicit operator is a large, sparse, and non-symmetric system matrix with dimensions equal to the total number of cells. In the LU-SGS method, to efficiently solve Eq. (30), the implicit operator is factorized into the diagonal and off-diagonal terms as follows,

$$\left[\frac{\mathbf{M}}{\Delta t} + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^n\right]\Delta \mathbf{U}^n = (\mathbf{D} + \mathbf{L})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{P})\Delta \mathbf{U}^n, \tag{31}$$

where $\mathbf{L}$, $\mathbf{P}$, and $\mathbf{D}$, respectively, denote the matrices of strictly lower triangular, upper triangular, and diagonal terms. In practical implementation, the coefficient matrix $\mathbf{M}$ in Eq. (31) can be simplified by the diagonal matrix which shares the diagonal elements $M_{i,i}$ with $\mathbf{M}$. Moreover, due to the unconcerned temporal accuracy, $\partial \mathbf{R}/\partial \mathbf{U}$ is not evaluated analytically but approximated by a simplified formulation with the spectral radius to reduce the numerical complexity. Finally, the operators $\mathbf{L}$, $\mathbf{P}$, and $\mathbf{D}$ for the reference cell Ω_i are given as

$$\mathbf{L} = \sum_{j \in L(i)} \frac{1}{2}\left[\frac{\Delta \boldsymbol{F}_{c,j}}{\Delta \mathbf{U}_j} \cdot \boldsymbol{n}_{ij} - (\Lambda_{ij} + \hat{\Lambda}_{ij})\boldsymbol{I}\right]A_{ij},$$

$$\mathbf{P} = \sum_{j \in P(i)} \frac{1}{2}\left[\frac{\Delta \boldsymbol{F}_{c,j}}{\Delta \mathbf{U}_j} \cdot \boldsymbol{n}_{ij} - (\Lambda_{ij} + \hat{\Lambda}_{ij})\boldsymbol{I}\right]A_{ij}, \tag{32}$$

$$\mathbf{D} = \left[\frac{M_{i,i}}{\Delta t} + \frac{1}{2}\sum_{j \in N(i)}(\Lambda_{ij} + \hat{\Lambda}_{ij})A_{ij}\right]\boldsymbol{I},$$

where $L(i)$ and $P(i)$, respectively, denote the nearest neighbors of the control cell Ω_i which belong to the lower and upper matrices. $N(i)$ is the set of direct neighbor cells of the cell Ω_i. $\Delta \boldsymbol{F}_{c,j} = \boldsymbol{F}_c(\mathbf{U}_j + \Delta \mathbf{U}_j) - \boldsymbol{F}_c(\mathbf{U}_j)$ and $\boldsymbol{F}_c$ is the convective flux. Λ_{ij} is the maximal eigenvalue of the convective flux Jacobian and the subscript ij denotes the interface between the control cells Ω_i and Ω_j. The factor $\hat{\Lambda}_{ij}$ denotes the maximum viscous eigenvalue. Equation (32) can be solved by the following forward and backward sweep procedures,

$$\Delta \mathbf{U}_i^* = \mathbf{D}^{-1}\left\{-\mathbf{R}_i^n - \sum_{j \in L(i)} \frac{1}{2}[\Delta \boldsymbol{F}_{c,j}^* \cdot \boldsymbol{n}_{ij} \right.$$
$$\left. - (\Lambda_{ij} + \hat{\Lambda}_{ij})\boldsymbol{I}\Delta \mathbf{U}_j^*]A_{ij}\right\},$$

$$\Delta \mathbf{U}_i^n = \Delta \mathbf{U}_i^* - \mathbf{D}^{-1}\left\{\sum_{j \in P(i)} \frac{1}{2}[\Delta \boldsymbol{F}_{c,j}^n \cdot \boldsymbol{n}_{ij} \right.$$
$$\left. - (\Lambda_{ij} + \hat{\Lambda}_{ij})\boldsymbol{I}\Delta \mathbf{U}_j^n]A_{ij}\right\}, \tag{33}$$

TABLE I. Butcher tableau for the third-order four-stage ESDIRK scheme.

J	1	2	3	4
a_{1j}	0	0	0	0
a_{2j}	$\frac{1767732205903}{4055673282236}$	$\frac{1767732205903}{4055673282236}$	0	0
a_{3j}	$\frac{2746238789719}{10658868560708}$	$\frac{-640167445237}{6845629431997}$	$\frac{1767732205903}{4055673282236}$	0
a_{4j}	$\frac{1471266399579}{7840856788654}$	$\frac{-4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
b_j	a_{41}	a_{42}	a_{43}	a_{44}

with

$$\Delta \boldsymbol{F}_{c,j}^* = \boldsymbol{F}_c(\mathbf{U}_j^n + \Delta \mathbf{U}_j^*) - \boldsymbol{F}_c(\mathbf{U}_j^n),$$

$$\Delta \boldsymbol{F}_{c,j}^n = \boldsymbol{F}_c(\mathbf{U}_j^n + \Delta \mathbf{U}_j^n) - \boldsymbol{F}_c(\mathbf{U}_j^n), \tag{34}$$

where $\Delta \mathbf{U}^*$ is the intermediate solution in the forward sweep. The macroscopic flow variables can be updated by $\mathbf{U}^{n+1} = \Delta \mathbf{U}^n + \mathbf{U}^n$ after obtaining the solution correction $\Delta \mathbf{U}^n$.

When an unsteady flow problem is considered, the temporal accuracy also plays a vital role. In this work, the third-order four-stage ESDIRK method with dual time-stepping (DTS) approach is incorporated to efficiently obtain the time-accurate solution. After applying the ESDIRK method to Eq. (10), we have

$$\mathbf{U}^k = \mathbf{U}^n - \Delta t \sum_{j=1}^{k} a_{kj}\mathbf{M}^{-1}\mathbf{R}(\mathbf{U}^j), \quad k = 1, 2, \ldots, \alpha, \tag{35}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \Delta t \sum_{j=1}^{\alpha} b_j\mathbf{M}^{-1}\mathbf{R}(\mathbf{U}^j), \tag{36}$$

where α denotes the number of stages; a_{kj} and b_j are the weights (cf. Table I). The vectors $\mathbf{U}^k$, $\mathbf{U}^n$, and $\mathbf{U}^{n+1}$ are, respectively, the solutions at stage k, the previous time level n and the final time level $n + 1$. Note that since the last stage in Eq. (35) gives the solution at the new time level, i.e., $\mathbf{U}^{n+1} = \mathbf{U}^\alpha$, it is unnecessary to compute Eq. (36). To advance $\mathbf{U}^k$ from the time level n to $n + 1$, the sequential set of $\alpha - 1$ nonlinear algebraic equations defined in Eq. (35) would be solved. For simplicity, the DTS approach is applied by adding a term $\partial \mathbf{U}^k/\partial \tau^*$ to Eq. (10), which yields

$$\mathbf{M}\frac{d\mathbf{U}}{d\tau^*} + \mathbf{M}\frac{d\mathbf{U}}{dt} = -\mathbf{R}(\mathbf{U}). \tag{37}$$

Then, at each stage k of the ESDIRK method in Eq. (35), we can get

$$\mathbf{M}\frac{\mathbf{U}^{k,s+1} - \mathbf{U}^{k,s}}{\Delta \tau^*}$$
$$= -\mathbf{M}\frac{\mathbf{U}^{k,s+1} - \mathbf{U}^n}{\Delta t} - a_{kk}\mathbf{R}(\mathbf{U}^{k,s+1}) - \sum_{j=1}^{k-1} a_{kj}\mathbf{R}(\mathbf{U}^j)$$
$$= -\mathbf{R}^*(\mathbf{U}^{k,s+1}), \quad k = 1, \ldots, \alpha, \tag{38}$$

where $\Delta \tau^*$ denotes the pseudotime step, s is the step of the pseudotime, and $\mathbf{R}^*(\mathbf{U}^{k,s+1})$ represents the unsteady residual.

TABLE II. Accuracy test for isotropic vortex problem on regular triangular grids.

Scheme	Grid size	L_∞ errors	Order	L_1 errors	Order
Fourth-order k-exact FV	1/2	3.54×10^{-2}		1.31×10^{-3}	
	1/4	3.16×10^{-3}	3.482	1.11×10^{-4}	3.557
	1/6	6.11×10^{-4}	4.055	1.97×10^{-5}	4.264
	1/8	1.78×10^{-4}	4.286	5.63×10^{-6}	4.352
Fourth-order RBFDQ FV	1/2	2.26×10^{-2}		1.08×10^{-3}	
	1/4	9.62×10^{-4}	4.555	4.26×10^{-5}	4.662
	1/6	1.52×10^{-4}	4.548	6.56×10^{-6}	4.615
	1/8	5.61×10^{-5}	3.470	2.29×10^{-6}	3.659

It is worth noting that the subscript "i" has been omitted for simplicity in Eqs. (37) and (38). In fact, Eq. (38) can be treated as a steady-state problem and the solution at each stage of the ESDIRK method can be obtained efficiently.

Furthermore, the unsteady residual $\mathbf{R}^*(\mathbf{U}^{k,s+1})$ given in Eq. (38) is linearized with respect to the known pseudotime level s at stage k as

$$\mathbf{R}^*(\mathbf{U}^{k,s+1}) \approx \mathbf{R}^*(\mathbf{U}^{k,s}) + \left(\frac{\partial \mathbf{R}^*}{\partial \mathbf{U}}\right)^{k,s} \Delta\mathbf{U}^{k,s}$$

$$= \mathbf{R}^*(\mathbf{U}^{k,s}) + \left[\frac{\mathbf{M}}{\Delta t} + a_{kk}\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^{k,s}\right]\Delta\mathbf{U}^{k,s},$$
(39)

where $\Delta\mathbf{U}^{k,s} = \mathbf{U}^{k,s+1} - \mathbf{U}^{k,s}$. Then the implicit integration in the pseudotime with the ESDIRK method is obtained in an unfactorized form as follows,

$$\left[\frac{\mathbf{M}}{\Delta\tau^*} + \frac{\mathbf{M}}{\Delta t} + a_{kk}\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^{k,s}\right]\Delta\mathbf{U}^{k,s}$$
$$= -\mathbf{R}^*(\mathbf{U}^{k,s}), \quad k = 1, \dots, \alpha,$$
(40)

where

$$\mathbf{R}^*(\mathbf{U}^{k,s}) = \mathbf{M}\frac{\mathbf{U}^{k,s}}{\Delta t} + a_{kk}\mathbf{R}(\mathbf{U}^{k,s})$$
$$+ \left[-\mathbf{M}\frac{\mathbf{U}^n}{\Delta t} + \sum_{j=1}^{k-1} a_{kj}\mathbf{R}(\mathbf{U}^j)\right].$$
(41)

Note that the matrix $\mathbf{M}$ in $\mathbf{R}^*(\mathbf{U}^{k,s})$ cannot be simplified to preserve the accuracy while $\mathbf{M}$ in the implicit operator on the left-hand side of Eq. (40) can be simplified to combine with the LU-SGS method. As a result, the operators $\mathbf{L}$, $\mathbf{P}$, and $\mathbf{D}$ at the kth stage of the ESDIRK method are cast as

$$\mathbf{L} = \sum_{j\in L(i)} \frac{1}{2}\left[\frac{\Delta \boldsymbol{F}_{c,j}}{\Delta \mathbf{U}_j} \cdot \boldsymbol{n}_{ij} - (\Lambda_{ij} + \hat{\Lambda}_{ij})\boldsymbol{I}\right]A_{ij},$$

$$\mathbf{P} = \sum_{j\in P(i)} \frac{1}{2}\left[\frac{\Delta \boldsymbol{F}_{c,j}}{\Delta \mathbf{U}_j} \cdot \boldsymbol{n}_{ij} - (\Lambda_{ij} + \hat{\Lambda}_{ij})\boldsymbol{I}\right]A_{ij}, \quad (42)$$

$$\mathbf{D} = \left[\frac{M_{i,i}}{\Delta\tau^*} + \frac{M_{i,i}}{\Delta t} + \frac{a_{kk}}{2}\sum_{j\in N(i)}(\Lambda_{ij} + \hat{\Lambda}_{ij})A_{ij}\right]\boldsymbol{I}.$$

The corresponding two-step solution procedure is given as

$$\Delta\mathbf{U}_i^{(1)} = \mathbf{D}^{-1}\left\{-\mathbf{R}^*(\mathbf{U}_i^s) - \sum_{j\in L(i)} \frac{1}{2}[\Delta\boldsymbol{F}_{c,j}^{(1)} \cdot \boldsymbol{n}_{ij}\right.$$

$$\left. - (\Lambda_{ij} + \hat{\Lambda}_{ij})\boldsymbol{I}\Delta\mathbf{U}_j^{(1)}]A_{ij}\right\},$$

$$\Delta\mathbf{U}_i^s = \Delta\mathbf{U}_i^{(1)} - \mathbf{D}^{-1}\left\{\sum_{j\in P(i)} \frac{1}{2}[\Delta\boldsymbol{F}_{c,j}^s \cdot \boldsymbol{n}_{ij}\right.$$

$$\left. - (\Lambda_{ij} + \hat{\Lambda}_{ij})\boldsymbol{I}\Delta\mathbf{U}_j^s]A_{ij}\right\}, \quad (43)$$
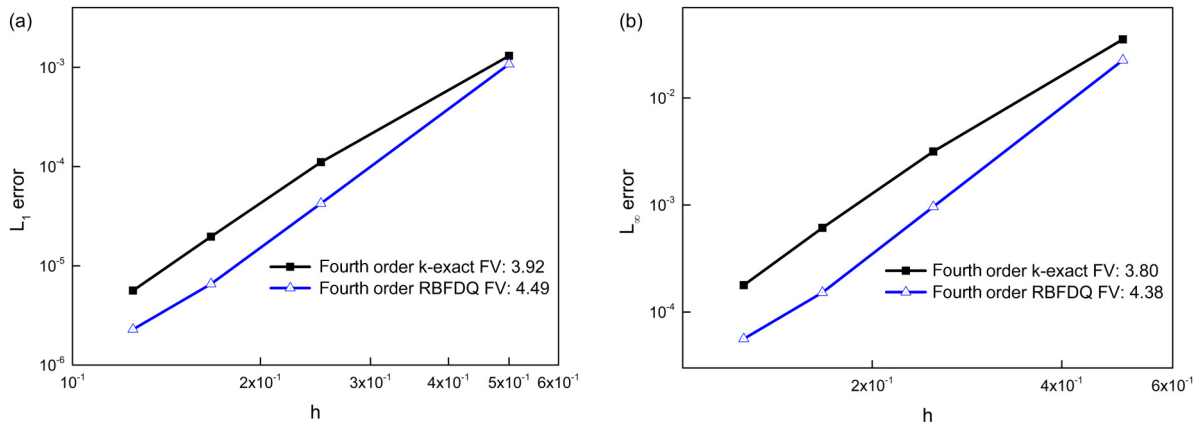

FIG. 1. Accuracy comparison between the high-order RBFDQ-FV and k-exact FV methods for the isentropic vortex problem on regular triangular grids: (a) L_1 norm, (b) L_∞ norm.

TABLE III. Accuracy test for decaying vortex flow on regular triangular grids.

Scheme	Grid size	L_1 errors	Order	L_2 errors	Order
	2/10	2.22×10^{-2}		2.51×10^{-2}	
	2/20	2.30×10^{-3}	3.272	2.57×10^{-3}	3.287
Fourth-order k-exact FV	2/30	4.64×10^{-4}	3.948	5.19×10^{-4}	3.948
	2/40	1.45×10^{-4}	4.050	1.62×10^{-4}	4.050
	2/80	8.28×10^{-6}	4.129	9.26×10^{-6}	4.127
	2/10	7.96×10^{-3}		9.01×10^{-3}	
	2/20	3.31×10^{-4}	4.589	3.94×10^{-4}	4.514
Fourth-order RBFDQ FV	2/30	6.48×10^{-5}	4.020	8.00×10^{-5}	3.935
	2/40	2.06×10^{-5}	3.981	2.59×10^{-5}	3.926
	2/80	1.85×10^{-6}	3.475	2.28×10^{-6}	3.501
	2/30	6.78×10^{-3}		6.09×10^{-3}	
	2/40	3.95×10^{-3}	1.881	3.55×10^{-3}	1.876
Second-order FV	2/80	1.02×10^{-3}	1.951	9.20×10^{-4}	1.949
	2/100	6.57×10^{-4}	1.982	5.91×10^{-4}	1.981
	2/160	2.58×10^{-4}	1.991	2.32×10^{-4}	1.990
	2/20	1.12×10^{-2}		1.25×10^{-2}	
Second-order RBFDQ FV	2/30	4.71×10^{-3}	2.127	5.25×10^{-3}	2.134
	2/40	2.72×10^{-3}	1.911	3.02×10^{-3}	1.917
	2/80	7.12×10^{-4}	1.933	8.24×10^{-4}	1.875

with

$$\Delta \boldsymbol{F}_{c,j}^{(1)} = \boldsymbol{F}_c\big(\mathbf{U}_j^s + \Delta\mathbf{U}_j^{(1)}\big) - \boldsymbol{F}_c\big(\mathbf{U}_j^s\big),$$

$$\Delta \boldsymbol{F}_{c,j}^{s} = \boldsymbol{F}_c\big(\mathbf{U}_j^s + \Delta\mathbf{U}_j^s\big) - \boldsymbol{F}_c\big(\mathbf{U}_j^s\big), \quad (44)$$

where $\Delta\mathbf{U}^{(1)}$ is the intermediate solution correction. Note that the subscript "k" in Eqs. (42)–(44) has been omitted for simplicity. Once $\mathbf{U}^s$ at the stage k meets the convergence tolerance of the pseudosteady-state solution, $\mathbf{U}^s$ will be set as the initial value for the computation at the next stage $k + 1$.

III. NUMERICAL RESULTS AND DISCUSSIONS

In this section, a series of benchmark cases is tested on unstructured grids to validate the accuracy, efficiency, and stability of the present implicit high-order RBFDQ-FV method with LBFS. In the unsteady simulation, the CFL number for the local pseudotime step is 8. There is no limitation for the number of iterations in every physical time step and the

convergence criterion for the pseudosteady state per physical time step is a drop of five orders of magnitude of the unsteady residual R^* from the initial value. Although the optimal shape parameter c may differ in different problems, in all simulations, we experimentally propose to set the shape parameter c to $300h_{\min}$ before the normalization. All simulations are conducted on a PC with 2.30 GHz CPU.

A. Accuracy tests

1. Accuracy test of high-order RBFDQ-FV method

To validate the accuracy of the present high-order RBFDQ-FV method without LBFS, the isentropic vortex transport problem [9,54] is tested. This benchmark case is commonly used to assess the accuracy of our numerical method. It involves convection of an isentropic vortex in an inviscid flow, where the free-stream conditions are $(\rho, u, v, p) = (1, 1, 1, 1)$. In the flow field, the following perturbations are
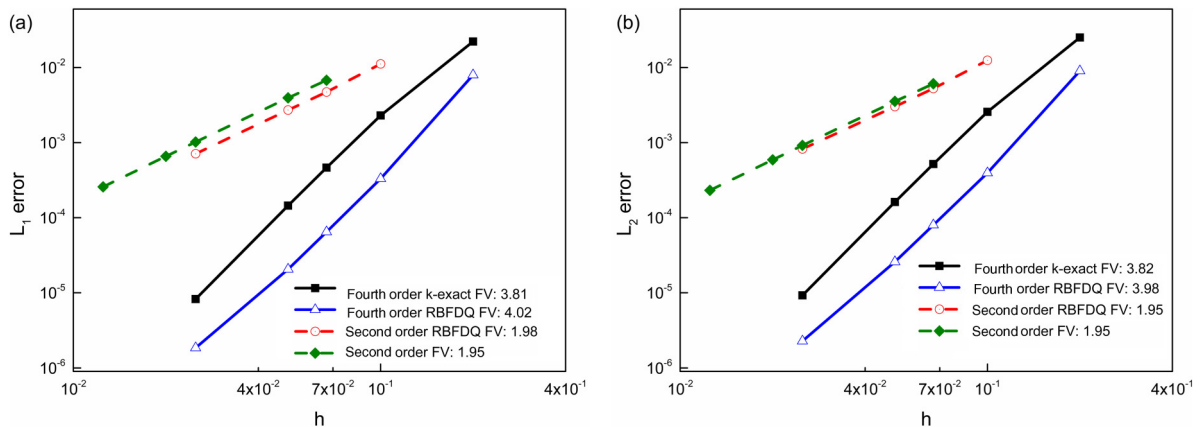

FIG. 2. Convergence studies for decaying vortex flow at $t = L/U$: (a) L_1 norm, (b) L_2 norm. The numbers in the figures denote the slopes of the linearly fitted lines.
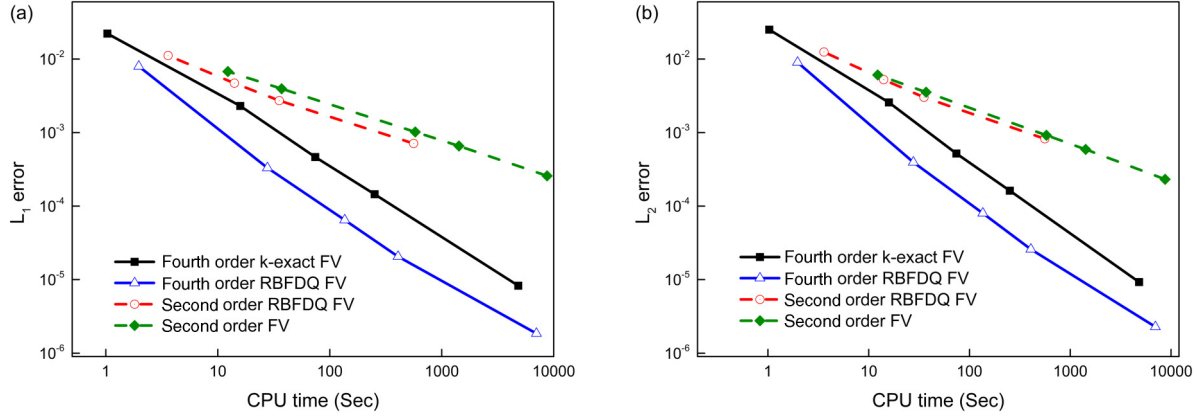
FIG. 3. Efficiency comparison between different FV methods for 2D decaying vortex flow at $t = L/U$: (a) L_1 norm, (b) L_2 norm.

added to the free stream without any entropy gradient,

$$(\delta u, \delta v) = \frac{\varepsilon}{2\pi} e^{0.5(1-r^2)} (-\bar{y}, \bar{x}), \tag{45}$$

where $(\bar{x}, \bar{y}) = (x-5, y-5)$, $r^2 = \bar{x}^2 + \bar{y}^2$ and the vortex strength $\varepsilon = 5$. The temperature T and entropy S are given as

$$T = 1 - \frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2}, \quad S = 1, \tag{46}$$

where γ is the specific heat ratio. The density ρ and pressure p are computed from T and S by

$$T = \frac{p}{\rho}, \quad S = \frac{p}{\rho^\gamma}. \tag{47}$$

The computational domain is $[0, 10] \times [0, 10]$ and the periodic boundary condition is applied to all boundaries. The regular triangular grids are used and the grid spacing ranges from 1/2 to 1/8. Since this case is dedicated to assess the accuracy of the RBFDQ-FV method without LBFS by simulating an inviscid flow problem, the inviscid fluxes are calculated by the widely used Roe scheme [55] and results at $t = 2.0$ are extracted. For comparison purposes, the accuracy tests of the k-exact FV method are performed as well.

Table II lists the errors of density in terms of L_1 and L_∞ norms as well as the convergence rates and Fig. 1 compares the accuracy. Clearly, the fourth-order accuracy of both the high-order RBFDQ-FV method and the k-exact FV method can be verified. In addition, the slopes of linearly fitted lines in Fig. 1 reveal the higher accuracy of the RBFDQ-FV method than the k-exact FV method. From Table II, the fourth-order RBFDQ-FV method gives lower relative errors than the fourth-order k-exact FV method. These observations evidently show that the present high-order RBFDQ-FV method outperforms the k-exact FV method in terms of accuracy.

2. Accuracy test of high-order RBFDQ-FV method with LBFS

As discussed in Sec. II C, the present high-order RBFDQ-FV method uses the LBFS to evaluate the inviscid and viscous fluxes simultaneously for incompressible flows. The first test has validated the fourth-order accuracy of the present RBFDQ-FV method itself. This accuracy test would further check whether the LBFS would spoil the high-order accuracy of the present method. In this test, the decaying vortex flow problem which has the following analytical solution is
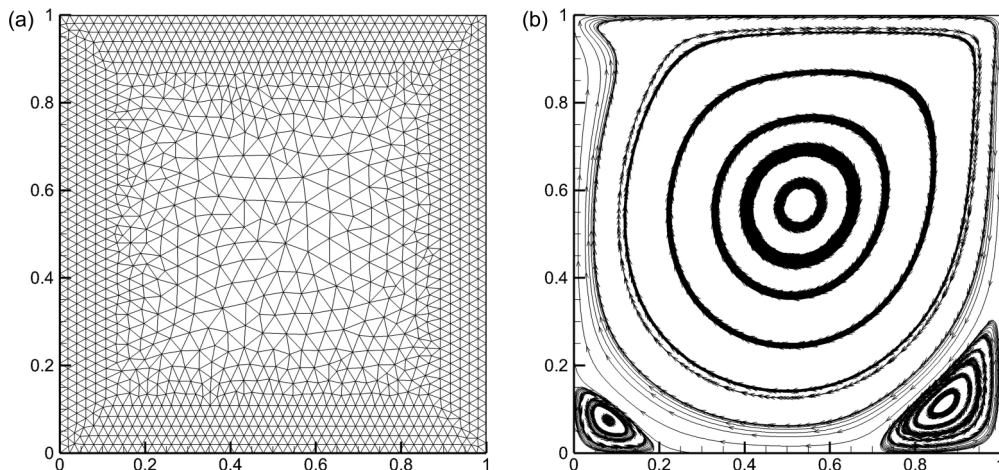

FIG. 4. Lid-driven cavity flow: (a) mesh and (b) streamlines obtained by the high-order RBFDQ-FV method at Re $= 1000$.
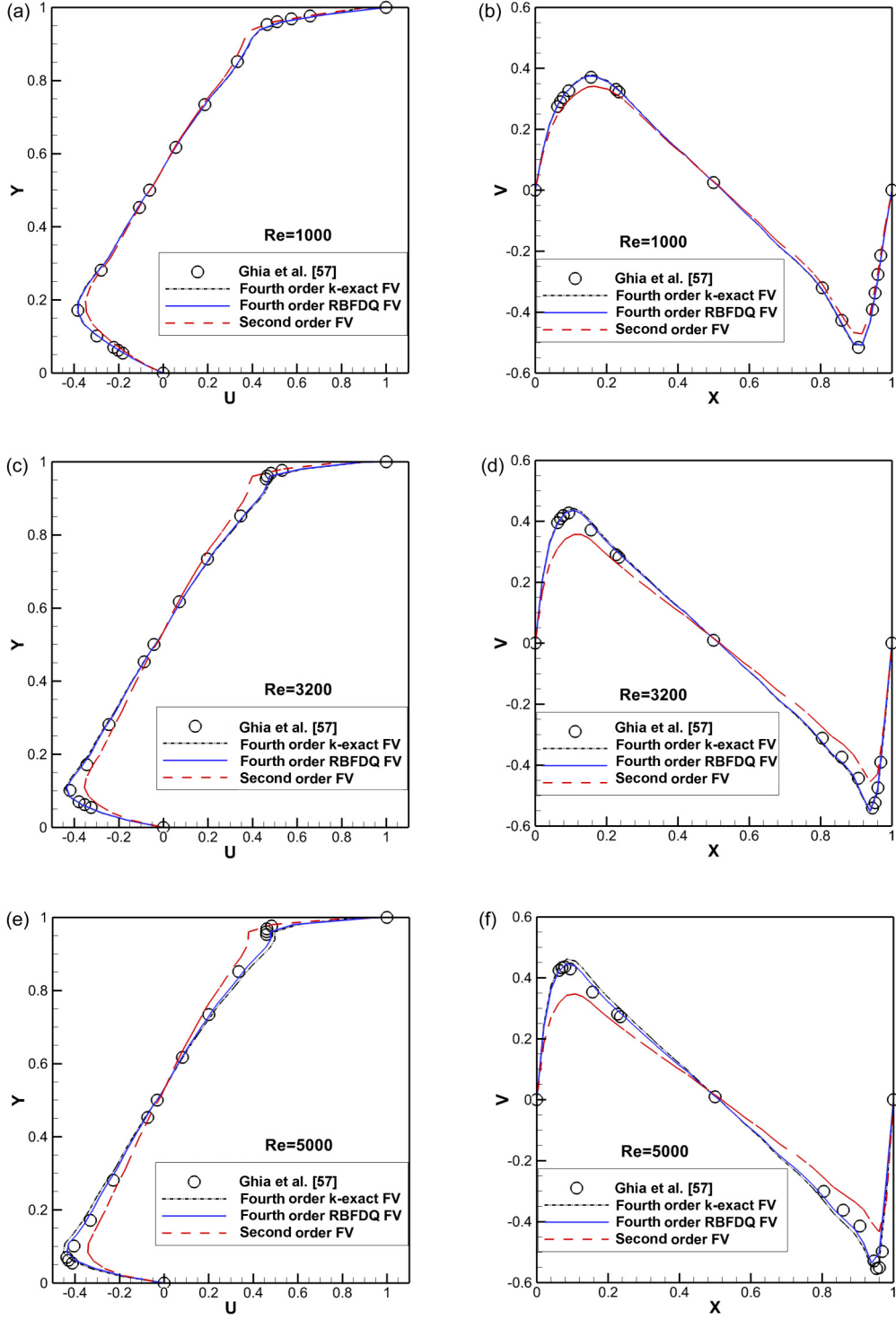
FIG. 5. Comparison of u-velocity and v-velocity profiles along vertical and horizontal central lines between different FV methods with LBFS: (a) u-velocity and (b) v-velocity profiles at Re = 1000; (c) u-velocity and (d) v-velocity profiles at Re = 3200; (e) u-velocity and (f) v-velocity profiles at Re = 5000.

considered,

$$u(x, y, t) = -U \cos(\pi x/L) \sin(\pi y/L) e^{-2\pi^2 Ut/(\text{Re}L)},$$

$$v(x, y, t) = U \sin(\pi x/L) \cos(\pi y/L) e^{-2\pi^2 Ut/(\text{Re}L)},$$

$$\rho(x, y, t) = \rho_0 - \frac{\rho_0 U^2}{4c_s^2} [\cos(2\pi x/L) + \cos(2\pi y/L)]$$

$$\times e^{-4\pi^2 Ut/(\text{Re}L)}, \tag{48}$$

where the Reynolds number is Re $= UL/\nu = 10$ and the initial density ρ_0 is taken as 1. The streaming distance δ_x equivalent to δ_t is selected as $0.2h_{\min}$ and the relaxation parameter τ_ν is set as 0.8. The computational domain is $[-L, L] \times [-L, L]$. The computations are run until $t = L/U$ and the relative errors of the computed velocity component u are measured using the L_1 norm and L_2 norm, which are

TABLE IV. Comparison of memory cost and computational time for lid-driven cavity flow.

Cells of mesh	Re	Fourth-order RBFDQ-FV		Second-order FV	
		Memory (Mb)	CPU time (s)	Memory (Mb)	CPU time (s)
2734	1000	8.5	277.34	3.5	66.13
	3200	8.5	834.41	3.5	198.45
	5000	8.5	1278.39	3.6	307.58
11032	1000			9.9	459.41
	3200			9.9	1459.81
	5000			9.9	2296.64

defined as

$$L_1(u) = \frac{1}{N_{\text{cell}}} \sum_{i=1}^{N_{\text{cell}}} \left(\left| \frac{u_i - u_i^e}{U} \right| \right),$$

$$L_2(u) = \left[\frac{1}{N_{\text{cell}}} \sum_{i=1}^{N_{\text{cell}}} \left(\frac{u_i - u_i^e}{U} \right)^2 \right]^{\frac{1}{2}}, \qquad (49)$$

where u_i and u_i^e denote the numerical result and the exact solution, respectively. N_{cell} is the number of cells in the flow domain.

In the convergence study, the regular triangular grids with the grid spacing of $h = 2/10$ to $2/160$ are used. Table III tabulates the L_1 and L_2 norms of the relative errors of the velocity component u and the rates of convergence. In Fig. 2, the accuracy of the high- and second-order FV methods is compared. Clearly, all the FV methods, i.e., the k-exact FV and RBFDQ-FV methods, achieve the theoretical accuracy with LBFS, which is consistent with the discussion of accuracy in Sec. II C. Additionally, the solutions obtained by both high-order methods are much more accurate than those computed by the second-order methods. Note that the relative errors given by the RBFDQ-FV method are smaller than those of the k-exact FV method, which confirms that the present RBFDQ-FV method is more accurate than the k-exact FV method with the same order of accuracy. Furthermore, Fig. 3 compares the efficiency of the four FV methods. It is observed that the fourth-order RBFDQ-FV method with LBFS requires

less CPU time than the high-order k-exact FV and the second-order FV methods to achieve the same accuracy. This verifies the higher efficiency of the high-order RBFDQ-FV method compared to the high-order k-exact FV and second-order FV methods.

B. Lid-driven cavity flow

The accuracy, memory cost and computational efficiency of the high-order RBFDQ-FV method with LBFS are comprehensively evaluated by the lid-driven flow in a square cavity [56,57]. The conventional second-order FV and the high-order k-exact FV methods with LBFS are also applied for comparison purposes. In this problem, the viscous flow in a 2D square cavity is driven by the moving lid with a constant velocity U to form a circulation flow while the other three solid walls remain static with the nonslip boundary condition. Here, the Reynolds number is defined as Re $= UL/\nu$, where L denotes the length of the square cavity. The velocity of the lid is taken as $U = 0.1$. The computational domain of $[0, 1] \times [0, 1]$ discretized by the unstructured triangular cells is presented in Fig. 4, where the nonuniform grid spacings are, respectively, $h = 1/20$ for the cells in the middle and $h = 1/45$ for the cells near the walls, having 2734 cells in total. The streamlines at Re $= 1000$ obtained by the high-order RBFDQ-FV method with LBFS are depicted in Fig. 4 as well. Figure 5 presents the u-velocity and v-velocity profiles along the vertical and horizontal central lines obtained by the conventional second-order
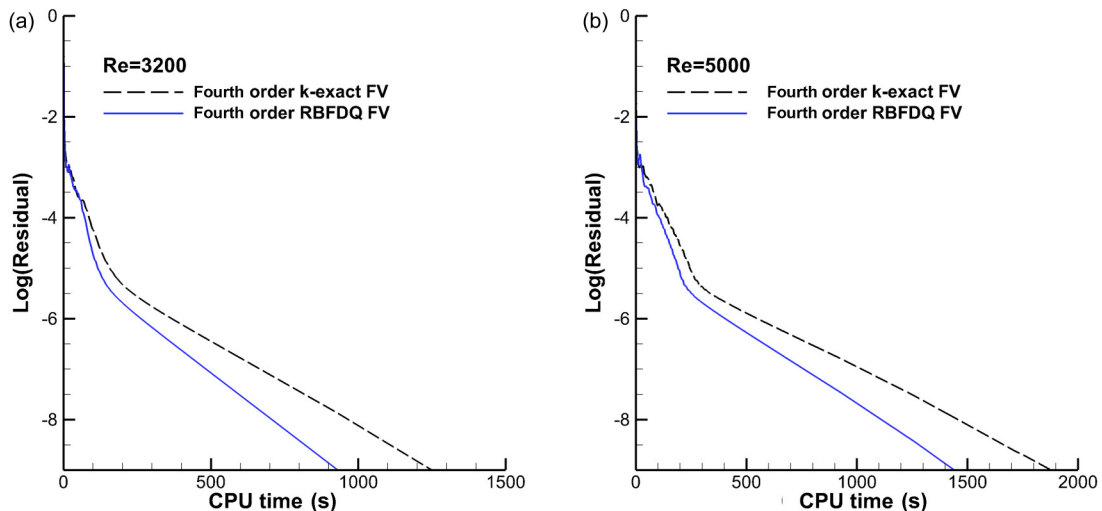

FIG. 6. Comparison of efficiency between the high-order RBFDQ-FV and k-exact FV methods at (a) Re $= 3200$ and (b) Re $= 5000$.
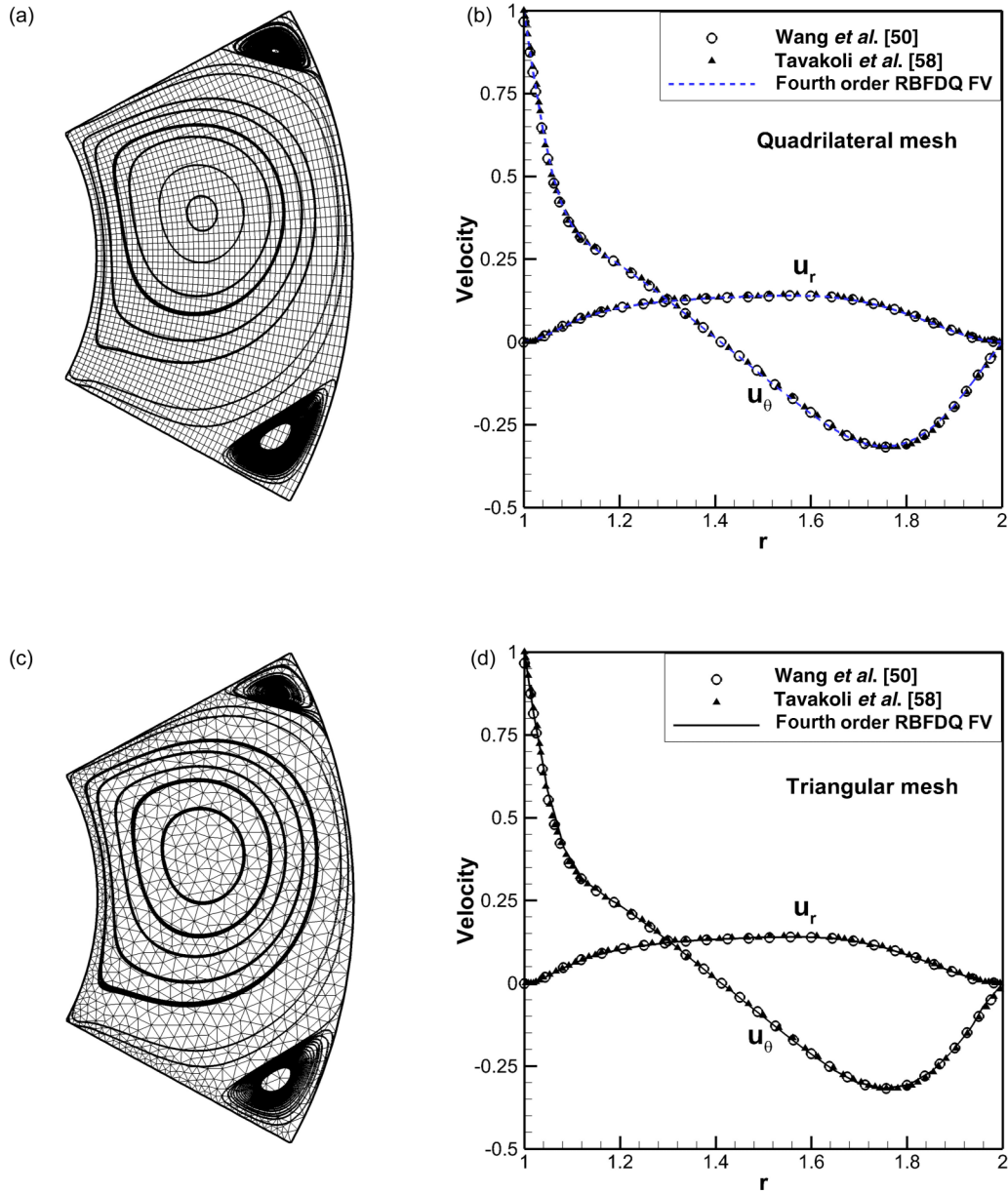
FIG. 7. Streamlines and velocity distributions along radial and azimuthal directions obtained by the high-order RBFDQ-FV method with LBFS: (a) streamlines and (b) velocity distributions on the unstructured quadrilateral mesh; (c) streamlines and (d) velocity distributions on the unstructured triangular mesh.

FV and the high-order RBFDQ-FV and k-exact FV methods with LBFS at three Reynolds numbers of Re = 1000, 3200, and 5000. The benchmark data of Ghia *et al.* [57] are also included. From the results, it is clear that, on the coarse mesh of 2734 cells, the high-order RBFDQ-FV and k-exact FV methods can give more accurate results than the second-order one. Moreover, results of the high-order RBFDQ-FV method achieve slightly better agreement with the benchmark data than those of the k-exact FV method, which demonstrates the higher accuracy of the present high-order method.

To compare the computational efficiency and memory cost of the present high-order method and the conventional second-order method, the mesh is refined for the second-order method to obtain comparable results. 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 11 032. As can be seen from the comparison of the computational cost in Table IV, when a similar accuracy is achieved on different mesh resolutions, the high-order RBFDQ-FV method is almost 1.8 times faster than the second-order method with less memory cost. Such outcomes provide strong evidence that the high-order RBFDQ-FV method performs better than the conventional second-order method in terms of numerical accuracy and computational efficiency.

Lastly, we compare the convergence and efficiency of the high-order RBFDQ-FV and k-exact FV methods for this steady case. Note that the LU-SGS scheme is also adopted as the time-marching strategy for the high-order k-exact FV method. As shown in Fig. 5, the results obtained by the high-order RBFDQ-FV method show slightly better agreement
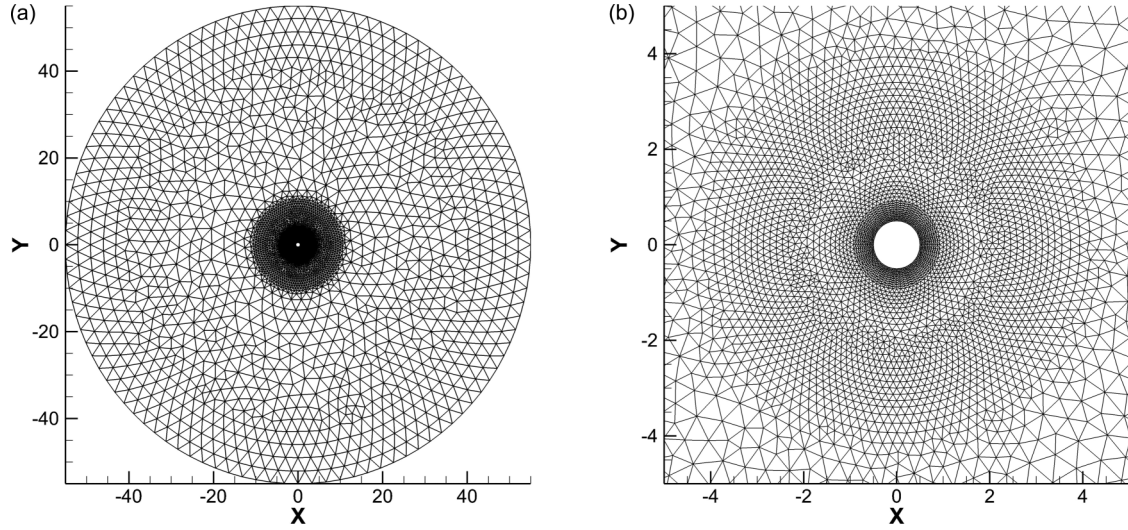
FIG. 8. Mesh used for viscous flow past a circular cylinder at Re = 100. Sixty-five grid points on the cylinder and 11 626 cells in total. (a) Computational domain and (b) zoomed mesh near cylinder.

with the benchmark data than the k-exact FV method on the mesh of 2734 cells at Re = 3200 and 5000. The corresponding convergence history and efficiency are compared in Fig. 6. It can be seen that the high-order RBFDQ-FV method converges faster than the k-exact FV method. Especially for the case of Re = 5000, the high-order RBFDQ-FV method uses only 74.5% of the CPU time spent by the high-order k-exact FV method. These observations clearly prove the better efficiency of the high-order RBFDQ-FV method than the k-exact FV method with the same order of accuracy.

C. Polar cavity flow

To further assess the performance of the present high-order method for simulating the incompressible flow problem with a curved boundary, the lid-driven flow in a polar cavity is tested. In this problem, a sector with the angle of 1 rad ($\theta = 1$) is

bounded by two straight walls and two curved walls at the radii of R_i and R_o [50]. The dimensionless Reynolds number defined as Re $= U_\theta (R_o - R_i)/\nu$ is used to characterize the flow pattern. U_θ is the azimuthal velocity on the inner curved wall with the radius of R_i. This simulation sets $R_i = 1.0$, $R_o = 2.0$, and $U_\theta = 0.1$. For comparison purposes, two grids are used for this problem with curved boundaries. One is the quadrilateral mesh which has 51×51 grid points and the other is the unstructured triangular mesh having 1790 triangle cells and 35 points on each side of the flow domain. Note that these two sets of grids are both coarser than the mesh used in Ref. [50].

In this case, results of the high-order RBFDQ-FV method with LBFS are presented. The obtained streamlines and
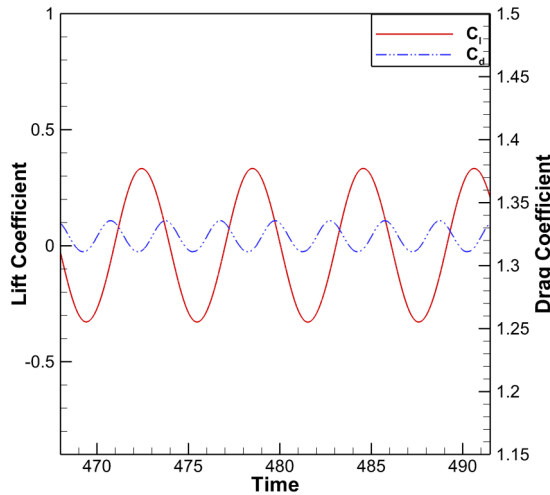

FIG. 9. Evolution of the lift and drag coefficients computed by the high-order RBFDQ-FV method for flow past a cylinder at Re = 100.
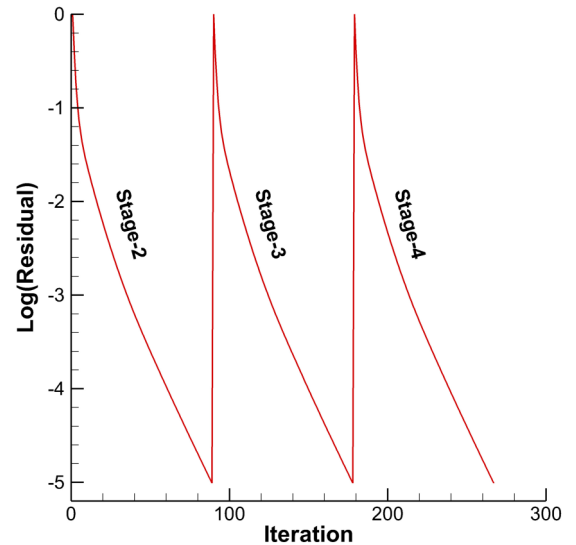

FIG. 10. Convergence history for the inner iteration at three implicit stages in the ESDIRK method used by the high-order RBFDQ-FV method for flow past a cylinder at Re = 100.

TABLE V. 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.* [37]	± 0.33	1.334 ± 0.02	0.164
	Wang and Zhou [61]	± 0.315	1.30 ± 0.006	0.159
	Pellerin *et al.* [51]	± 0.325	1.325	0.164
	Second-order FV method	± 0.327	1.338 ± 0.011	0.158
	Fourth-order RBFDQ-FV method	± 0.33	1.323 ± 0.012	0.164

velocity distributions along the radial and azimuthal directions at Re = 350 on the unstructured quadrilateral and triangular grids are depicted in Fig. 7. Clearly, they agree well with the data from Wang *et al.* [50] and Tavakoli *et al.* [58], even on the coarser mesh. This evidently demonstrates the high reliability of the present RBFDQ-FV method with LBFS for solving the problem with a curved boundary on unstructured grids.

D. Viscous flow past a stationary circular cylinder

To further examine the capability of the present high-order method for unsteady flow problems, the viscous flow past a stationary circular cylinder is simulated. The physical config-uration of this case is that the incoming viscous fluid with a constant free-stream velocity U_0 flows over a fixed circu-lar cylinder. To meet the incompressible limit, $U_0 = 0.1$ is adopted. The Reynolds number affecting the flow pattern is defined as Re $= U_0 L/\nu$, where L is the diameter of the circular cylinder. In this test, Re = 100. The pressure coefficient C_p, lift coefficient C_l, drag coefficient C_d, and Strouhal number St are used to quantify the numerical results and they are defined as follows,

$$C_p = \frac{p_w - p_0}{\rho_0 U_0^2/2}, \quad C_l = \frac{F_l}{\rho_0 U_0^2/2},$$

$$C_d = \frac{F_d}{\rho_0 U_0^2/2}, \quad \text{St} = \frac{f_q L}{U_0}, \quad (50)$$

where p_0 and p_w denote the pressure of the free stream and on the cylinder surface, respectively. F_l and F_d are, respectively, the lift force and the drag force. f_q represents the vortex shed-ding frequency. The nonslip boundary condition is applied on the cylinder surface and the far-field free-stream condition is enforced on the outer boundary. An unstructured mesh with 65 points on the cylinder surface and 11 626 cells in total is used. The far-field boundary is set at 55L away from the center of the cylinder. Figure 8 shows the flow domain and the unstructured mesh. The calculations are performed until $t = 500$ with a fixed physical time-step size of $\Delta t = 0.1$. The CFL number for the local pseudotime step is 8. Based on this setting for the implicit temporal scheme, the convergence for the pseudosteady state can be achieved within 100 iterations.

Results obtained by the high-order RBFDQ-FV and the conventional second-order FV methods with LBFS are ex-tracted and discussed. The evolution of the lift and drag coefficients of the high-order RBFDQ-FV method is shown in Fig. 9. Figure 10 shows one example for the inner iteration in the pseudotime at one physical time step. The numbers of

iterations for three implicit stages are all 89. Table V tabu-lates the computed dynamic parameters and the corresponding reference data [37,51,59]. As can be seen, on the same un-structured grids, the results of the RBFDQ-FV method are all within the range of the reference data and they have better agreement with the reference data than the ones of the second-order method. This validates the capability and flexibility of the present high-order method for simulating the unsteady incompressible flow problem on unstructured grids.

IV. CONCLUSIONS

This paper presents an efficient high-order finite volume method on unstructured grids for incompressible flows. In this method, a Taylor series expansion within each control cell is directly applied as the solution approximation function. The unknown derivatives are approximated by the mesh-free RBFDQ scheme. In this way, the high-order approximation of the flow variables is constructed straightforwardly and simply. The LBFS is applied to evaluate the inviscid and viscous fluxes at the cell interface simultaneously in a physical way, which avoids additional treatments for the viscous fluxes. This makes the RBFDQ-FV method competitive in simulating viscous flow problems. Different from other high-order FV methods, in the present method, the time-dependent term in-volves a coefficient matrix resultant from the volume integral of the high-order approximation function. This reflects the im-plicit nature of the present method and the coefficient matrix is coupled into the implicit LU-SGS and ESDIRK schemes as the efficient time-marching strategy.

To validate the accuracy and efficiency of the present method and its capability on unstructured grids, several nu-merical tests for incompressible flows are conducted. The obtained numerical results validate the fourth-order accuracy of the present high-order RBFDQ-FV method on unstruc-tured grids. It is shown that the present high-order method enjoys higher accuracy and better efficiency than the conven-tional second-order method. As compared with the k-exact FV method, the accuracy of the present high-order method is slightly higher and its computational efficiency is better. For instance, for lid-driven cavity flow at Re = 5000, the high-order RBFDQ-FV method takes only about 74.5% of the computational time of the k-exact method.

[1] L. Pan and K. Xu, A third-order compact gas-kinetic scheme on unstructured meshes for compressible Navier-Stokes solutions, J. Comput. Phys. **318**, 327 (2016).

[2] A. Boularas, S. Clain, and F. Baudoin, A sixth-order finite volume method for diffusion problem with curved boundaries, Appl. Math. Modell. **42**, 401 (2017).

[3] F. Bassi, A. Crivellini, S. Rebay, and M. Savini, Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k-ω turbulence model equations, Comput. Fluids **34**, 507 (2005).

[4] Q. Wang, Y. X. Ren, J. Pan, and W. Li, Compact high order finite volume method on unstructured grids III: Variational reconstruction, J. Comput. Phys. **337**, 1 (2017).

[5] F. Meng, J. W. Banks, W. D. Henshaw, and D. W. Schwendeman, Fourth-order accurate fractional-step IMEX schemes for the incompressible Navier-Stokes equations on moving overlapping grids, Comput. Methods Appl. Mech. Eng. **366**, 113040 (2020).

[6] O. Friedrich, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, J. Comput. Phys. **144**, 194 (1998).

[7] Y. Liu and Y. T. Zhang, A robust reconstruction for unstructured WENO schemes, J. Sci. Comput. **54**, 603 (2013).

[8] A. J. Christlieb, Y. Liu, Q. Tang, and Z. Xu, High order parametrized maximum-principle-preserving and positivity-preserving WENO schemes on unstructured meshes, J. Comput. Phys. **281**, 334 (2015).

[9] C. Hu and C. W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. **150**, 97 (1999).

[10] T. J. Barth and P. O. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, in *Proceedings of 28th Aerospace Sciences Meeting* (AIAA, Reston, VA, 1990), AIAA Paper 90-0013.

[11] T. J. Barth, Recent developments in high-order k-exact reconstruction on unstructured meshes, in *Proceedings of 31st Aerospace Sciences Meeting* (AIAA, Reston, VA, 1993), AIAA Paper 93-0668.

[12] C. Ollivier-Gooch and M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, J. Comput. Phys. **181**, 729 (2002).

[13] L. Cueto-Felgueroso, I. Colominas, X. Nogueira, F. Navarrina, and 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**, 4712 (2007).

[14] Y. Liu, W. Zhang, Y. Jiang, and Z. Ye, A high-order finite volume method on unstructured grids using RBF reconstruction, Comput. Math. Appl. **72**, 1096 (2016).

[15] G. Chesshire and W. D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comput. Phys. **90**, 1 (1990).

[16] W. D. Henshaw, A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids, J. Comput. Phys. **113**, 13 (1994).

[17] W. H. Reed and T. R. Hill, Triangular Mesh Methods for the Neutron Transport Equation, Technical Report No. LA-UR-73-479 (Los Alamos Scientific Laboratory, Los Alamos, NM, 1973).

[18] B. Cockburn and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, Math. Comput. **52**, 411 (1989).

[19] B. Cockburn and C. W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems, J. Comput. Phys. **141**, 199 (1998).

[20] J. Cheng, H. Yue, S. Yu, and T. Liu, A direct discontinuous Galerkin method with interface correction for the compressible Navier-Stokes equations on unstructured grids, Adv. Appl. Math. Mech. **10**, 1 (2018).

[21] F. Bassi and S. Rebay, A higher-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, J. Comput. Phys. **131**, 267 (1997).

[22] Z. J. Wang and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: Extension to two dimensional systems, J. Comput. Phys. **194**, 716 (2004).

[23] Z. J. Wang and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids VI: Extension to viscous flow, J. Comput. Phys. **215**, 41 (2006).

[24] Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: Basic formulation, J. Comput. Phys. **178**, 210 (2002).

[25] Z. J. Wang, Y. Liu, G. May, and A. Jameson, Spectral difference method for unstructured grids II: Extension to the Euler equations, J. Sci. Comput. **32**, 45 (2007).

[26] Z. J. Wang and Y. Sun, High-order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids, Commun. Comput. Phys. **2**, 301 (2007).

[27] Z. J. Wang and 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**, 8161 (2009).

[28] H. Gao and Z. J. Wang, A conservative correction procedure via reconstruction formulation with the chain-rule divergence evaluation, J. Comput. Phys. **232**, 7 (2013).

[29] C. Shu, H. Ding, and K. S. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. **192**, 941 (2003).

[30] C. Shu, H. Ding, H. Q. Chen, and T. G. Wang, An upwind local RBF-DQ method for simulation of inviscid compressible flows, Comput. Methods Appl. Mech. Eng. **194**, 2001 (2005).

[31] H. Ding, C. Shu, K. S. Yeo, and D. Xu, Numerical computation of three-dimensional incompressible viscous flows in the primitive variable form by local multiquadric differential quadrature method, Comput. Methods Appl. Mech. Eng. **195**, 516 (2006).

[32] W. X. Wu, C. Shu, and C. M. Wang, Vibration analysis of arbitrarily shaped membranes using local radial basis function-based differential quadrature method, J. Sound Vib. **306**, 252 (2007).

[33] Y. Sun and Z. J. Wang, Formulations and analysis of the spectral volume method for the diffusion equation, Commun. Numer. Meth. Eng. **20**, 927 (2004).

[34] J. J. Douglas and T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods, in *Computing Methods in Applied Sciences*, Lecture Notes in Physics Vol. 58, edited by R. Glowinski and J. L. Lions (Springer, Berlin, 1976).

[35] B. Cockburn and C. W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal. **35**, 2440 (1998).

[36] F. Bassi and S. Rebay, GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations, in *Discontinuous Galerkin Methods. Theory, Computation, and Applications*, *Lecture Notes in Computational Science and Engineering Vol. 11*, edited by M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, and T. Schlick (Springer-Verlag, New York, 2000), p. 197.

[37] C. Shu, Y. Wang, C. J. Teo, and J. Wu, Development of lattice Boltzmann flux solver for simulation of incompressible flows, Adv. Appl. Math. Mech. **6**, 436 (2014).

[38] W. Qin, J. Cai, H. Lu, P. K. Jimack, and M. A. Walkley, A study of preconditioned Jacobian-free Newton-Krylov discontinuous Galerkin method for compressible flows on 3D hexahedral grids, Adv. Appl. Math. Mech. **10**, 261 (2018).

[39] H. Bijl, M. H. Carpenter, V. N. Vatsa, and C. A. Kennedy, Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: Laminar flow, J. Comput. Phys. **179**, 313 (2002).

[40] Q. Tang, L. Chacon, T. V. Kolev, J. N. Shadid, and X. Z. Tang, An adaptive scalable fully implicit algorithm based on stabilized finite element for reduced visco-resistive MHD, arXiv:2106.00260.

[41] E. Franck, M. Hölzl, A. Lessig, and E. Sonnendrücker, Energy conservation and numerical stability for the reduced MHD models of the non-linear JOREK code, ESAIM: Math. Modell. Numer. Anal. **49**, 1331 (2015).

[42] H. Luo, J. D. Baum, and R. Löhner, A fast, matrix-free implicit method for compressible flows on unstructured grids, J. Comput. Phys. **146**, 664 (1998).

[43] M. Cheng, G. Wang, and H. H. Mian, Reordering of hybrid unstructured grids for an implicit Navier-Stokes solver based on openMP parallelization, Comput. Fluids **110**, 245 (2015).

[44] R. F. Chen and Z. J. Wang, An improved LU-SGS scheme with faster convergence for unstructured grids of arbitrary topology, in *Proceedings of 37th Aerospace Sciences Meeting* (AIAA, Reston, VA, 1999), AIAA Paper 99-0935.

[45] C. Shu, *Differential Quadrature and Its Application in Engineering* (Springer-Verlag London, 2000).

[46] C. Shu and B. E. Richards, Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations, Int. J. Numer. Methods Fluids **15**, 791 (1992).

[47] C. Shu and Y. T. Chew, Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems, Commun. Numer. Meth. Eng. **13**, 643 (1997).

[48] C. A. Micchelli, Interpolation of scattered data: Distance matrices and conditionally positive definite functions, Constr. Approx. **2**, 11 (1986).

[49] H. Ding, C. Shu, and D. B. Tang, Error estimates of local multiquadric-based differential quadrature (LMQDQ) method through numerical experiments, Int. J. Numer. Meth. Eng. **63**, 1513 (2005).

[50] Y. Wang, C. Shu, and 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**, 344 (2014).

[51] N. Pellerin, S. Leclaire, and M. Reggio, Solving incompressible fluid flows on unstructured meshes with the lattice Boltzmann flux solver, Eng. Appl. Comput. Fluid Mech. **11**, 310 (2017).

[52] Z. Chen, C. Shu, and D. Tan, Highly accurate simplified lattice Boltzmann method, Phys. Fluids **30**, 103605 (2018).

[53] J. Blazek, *Computational Fluid Dynamics: Principles and Applications* (Butterworth-Heinemann, Oxford, 2015).

[54] J. Li and Z. Du, A two-stage fourth order time-accurate discretization for Lax–Wendroff type flow solvers I. Hyperbolic conservation laws, SIAM J. Sci. Comput. **38**, A3046 (2016).

[55] P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. **43**, 357 (1981).

[56] S. Liao, Y. Zhang, and D. Chen, Runge-Kutta finite element method based on the characteristic for the incompressible Navier-Stokes equations, Adv. Appl. Math. Mech. **11**, 1415 (2019).

[57] U. Ghia, K. N. Ghia, and C. T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys. **48**, 387 (1982).

[58] E. Tavakoli, B. Lessani, and R. Hosseini, High-order pole-treatment in cylindrical coordinates for incompressible flow simulations with finite-difference collocated schemes, J. Comput. Phys. **296**, 1 (2015).

[59] M. Braza, P. Chassaing, and 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**, 79 (1986).

[60] C. Liu, X. Zheng, and C. H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, J. Comput. Phys. **139**, 35 (1998).

[61] J. Wang and C. H. Zhou, A novel immersed boundary method implemented by imposing reconstructed velocity on virtual boundary, Adv. Appl. Math. Mech. **13**, 83 (2021).