Journal of Computational Physics 316 (2016) 700–720

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A multi-dimensional high-order DG-ALE method based on gas-kinetic theory with application to oscillating bodies

Xiaodong Ren [a,*], Kun Xu [a,b], Wei Shyy [b]

[a] Department of Mathematics, School of Science, Hong Kong University of Science and Technology, Hong Kong, China
[b] Department of Mechanical and Aerospace Engineering, School of Engineering, Hong Kong University of Science and Technology, Hong Kong, China

ARTICLE INFO

Article history:
Received 17 May 2015
Received in revised form 31 January 2016
Accepted 13 April 2016
Available online 19 April 2016

Keywords:
Gas-kinetic theory
Arbitrary Lagrangian–Eulerian (ALE) method
Discontinuous Galerkin (DG)
Geometric conservation law (GCL)

ABSTRACT

This paper presents a multi-dimensional high-order discontinuous Galerkin (DG) method in an arbitrary Lagrangian–Eulerian (ALE) formulation to simulate flows over variable domains with moving and deforming meshes. It is an extension of the gas-kinetic DG method proposed by the authors for static domains (X. Ren et al., 2015 [22]). A moving mesh gas kinetic DG method is proposed for both inviscid and viscous flow computations. A flux integration method across a translating and deforming cell interface has been constructed. Differently from the previous ALE-type gas kinetic method with piecewise constant mesh velocity at each cell interface within each time step, the mesh velocity variation inside a cell and the mesh moving and rotating at a cell interface have been accounted for in the finite element framework. As a result, the current scheme is applicable for any kind of mesh movement, such as translation, rotation, and deformation. The accuracy and robustness of the scheme have been improved significantly in the oscillating airfoil calculations. All computations are conducted in a physical domain rather than in a reference domain, and the basis functions move with the grid movement. Therefore, the numerical scheme can preserve the uniform flow automatically, and satisfy the geometric conservation law (GCL). The numerical accuracy can be maintained even for a largely moving and deforming mesh. Several test cases are presented to demonstrate the performance of the gas-kinetic DG-ALE method.

1. Introduction

Problems with variable domains are commonly present in many engineering applications, such as aeroelastic problems [1–4], thermoelastic, and insect flight problems [5]. Due to the complication of these problems, measurements are sometimes limited for capturing partial information of these complex flow phenomena. Fortunately, the numerical simulation gives us another choice to understand the flow problem. Generally, in order to resolve and capture these complex phenomena, such as separations and vortices, a high order numerical method is necessary. Moreover, the numerical method used should be able to handle variable domains while maintaining a high order of accuracy.

* Corresponding author.
E-mail addresses: maxdren@ust.hk (X. Ren), makxu@ust.hk (K. Xu), weishyy@ust.hk (W. Shyy).

http://dx.doi.org/10.1016/j.jcp.2016.04.028
0021-9991/

Most finite volume (FV) methods [6,7] are based on the piecewise constant representation of flow variables and resort to the reconstruction techniques to obtain high order accuracy in space. Generally, these methods are effective for structured meshes, but may face reconstruction problems on arbitrary grids, such as the multidimensional unstructured mesh cases, due to the adaptation of extended stencils. In order to avoid difficulties in the reconstruction, one possible way is to develop compact high order methods. The discontinuous Galerkin (DG) method [8,9] becomes one of these idealized choices. For the DG method, the higher accuracy is achieved by means of higher order polynomial approximation inside each cell. Only the information from adjacent cells with common cell interfaces is needed for the update of the degree of freedom within each cell. Therefore, the DG method can deliver higher order accurate solutions without solely relying on the reconstruction techniques and large stencils, but the smoothness of the solution seems required in order to faithfully use the DG method, especially for the Navier–Stokes solutions.

The arbitrary Lagrangian–Eulerian (ALE) method [10] is commonly used for simulations of flows involving moving boundaries. In the ALE description, the grids can be moved in any arbitrarily specified way independent of the flow motions. Due to its flexibility, large distortions of mesh can be handled in comparison with the Lagrangian framework. At the same time, the interfaces between different materials can be tracked much more easily in comparison with the purely Eulerian approach. Thus, combining a DG method for high-order accuracy with the ALE approach for its flexibility may provide us an effective way to simulate flows over deformable and movable domains.

In the past two decades, a number of DG methods in the ALE formulation have been proposed for both compressible [11–17] and incompressible flows [18–20]. An ALE matrix-free DG method was first presented by Lomtev et al. [11] for compressible viscous flows in two- and three-dimensional (2D and 3D) moving domains. A force-directed algorithm from graph theory is used to update the moving mesh while minimizing distortions. Klaij et al. [17] extended the ALE space-time DG method [13,14] from inviscid compressible flows to the compressible Navier–Stokes (NS) equations. Persson et al. [15] introduced a continuous mapping between a fixed reference configuration and the time-dependent domain. The NS equations are reformulated as conservation laws for the independent variables in the reference configuration space and solved thereafter. An additional equation for the time evolution of the transformation Jacobian is used to preserve the freestream solution. Nguyen [16] presented an ALE DG method in which the geometric conservation law (GCL) is always satisfied, with the preservation of freestream solution. However, for all these DG methods, the equations with higher order derivatives must be transformed into a first order system in order to avoid the so-called "non-conforming problem", and the inviscid and viscous fluxes are numerically treated differently. Besides, a separating temporal discretization, for instance, the total variation diminishing (TVD) Runge–Kutta method, is always needed to improve the stability and time accuracy.

Alternatively, the NS solutions can be recovered using a kinetic equation [21,22], in which a kinetic flux function including both inviscid and viscous terms can be obtained through a kinetic evolution model. In the gas kinetic scheme, the fluxes are constructed based on the integral solution of the gas kinetic Bhatnagar–Gross–Krook (BGK) model, which presents a multiscale evolution process from an initial non-equilibrium to an equilibrium state. Different from the Riemann problem, the flow evolution from a higher order initial reconstruction is explicitly followed. Based on the multi-dimensional gas kinetic formulation, a spatial and temporal dependent gas distribution function can be evaluated explicitly, and the flux is just computed at the cell barycenter or the interface center rather than on each Gaussian Quadrature points. Jin and Xu [23] proposed a 2nd order unified moving grid gas kinetic method in Eulerian space for viscous flow computations. A generalized coordinate transformation with arbitrary grid velocities is used to reformulate the gas kinetic BGK model in a moving frame reference and a unified direction splitting gas kinetic scheme is developed for viscous flow computations with moving meshes. Ni et al. [24] presented a 2nd order remapping-free ALE-type method, where the coupling of the fluid equations and GCL in [23] is removed. However, a piecewise constant grid velocity for each cell interface is used within each time step. It is a reasonable assumption for a 2nd order method, but not straightforward for a 3rd order method, where different grid velocity along a cell interface needs to be taken into account. For the traditional DG-ALE method [16], the Riemann solver is applied to each quadrature point, and the grid velocity distribution along the cell interface or inside the cell can be introduced implicitly. However, in the gas kinetic DG-ALE method, the higher-order gas-kinetic flux is a continuous function of space and time. How to couple this flux function with the DG formulation and a variation of mesh velocity needs investigated.

In this paper, a multi-dimensional 3rd order gas kinetic scheme with a grid velocity distribution is developed for flow simulations with moving or deforming meshes. The distribution of grid velocities inside a cell and along a cell interface can be varied within each time step and the GCL can be satisfied automatically. The paper is organized as follows. Section 2 is the construction of the numerical method. Section 3 shows some numerical examples and results. The last section draws the conclusion.

2. Multi-dimensional high-order ALE method based on gas kinetic theory

2.1. ALE formulation

Based on the 3D gas kinetic BGK model and the Chapman–Enskog expansion, the corresponding NS equations can be derived [21,22]:

$$\frac{\partial \boldsymbol{Q}(\boldsymbol{x},t)}{\partial t} + \nabla \bullet \boldsymbol{F} = \boldsymbol{0}, \quad \boldsymbol{x} \in \Omega^t, \tag{1}$$

where $\boldsymbol{x} = (x, y, z)$ is the space coordinates, t is the physical time, $\boldsymbol{Q}$ is the macroscopic conservative variables vector, $\boldsymbol{F}$ is the flux tensor, and Ω^t indicates the time-dependent flow domain.

In the ALE description, a referential domain Ω^0 is taken into account and the referential coordinates $\boldsymbol{X} = (X, Y, Z)$ are introduced to identify the gird nodes [10]. Here the reference domain is decomposed into a finite number of cells expressed as

$$\Omega^0 \doteq \bigcup_{e=1}^{m} \Omega_e^0 = \left\{\Omega_1^0, \Omega_2^0, \cdots, \Omega_m^0\right\} = \left\{\left\{\boldsymbol{X} | \boldsymbol{X} \in \Omega_1^0\right\}, \left\{\boldsymbol{X} | \boldsymbol{X} \in \Omega_2^0\right\}, \cdots, \left\{\boldsymbol{X} | \boldsymbol{X} \in \Omega_m^0\right\}\right\}, \quad m \in \mathbb{N}, \tag{2}$$

where m is the number of cells. A Taylor series expansion at the barycenter of the reference cell is used to represent the approximate solution [22] and the basis functions are denoted by $\{\upsilon_0^0(\boldsymbol{X}) = 1, \upsilon_1^0(\boldsymbol{X}), \cdots, \upsilon_{l-1}^0(\boldsymbol{X})\}$ with l basis functions.

The referential domain Ω^0 is mapped into a time-dependent flow domain Ω^t by using a time-dependent one-to-one mapping φ^t, which can be understood as the motion of grid nodes in the physical domain and can be represented by

$$\varphi^t(\boldsymbol{X}) = \boldsymbol{x} \in \Omega^t, \qquad \boldsymbol{X} = \phi^t(\boldsymbol{x}) \in \Omega^0, \tag{3}$$

in which ϕ^t is the inverse mapping of φ^t.

By using the mapping ϕ^t, one can obtain the time-dependent domain decomposition

$$\Omega^t \doteq \bigcup_{e=1}^{m} \Omega_e^t = \left\{\Omega_1^t, \Omega_2^t, \cdots, \Omega_m^t\right\} = \left\{\left\{\boldsymbol{x} | \phi^t(\boldsymbol{x}) \in \Omega_1^0\right\}, \left\{\boldsymbol{x} | \phi^t(\boldsymbol{x}) \in \Omega_1^0\right\}, \cdots, \left\{\boldsymbol{x} | \phi^t(\boldsymbol{x}) \in \Omega_m^0\right\}\right\}, \quad m \in \mathbb{N}, \tag{4}$$

and the corresponding basis functions are defined as

$$\upsilon_i^t(\boldsymbol{x}) = \upsilon_i^0\left(\phi^t(\boldsymbol{x})\right) = \upsilon_i^0(\boldsymbol{X}), \quad i = 0, 1, \cdots, l-1. \tag{5}$$

Therefore, the approximate polynomial solution in each cell can be written as

$$\boldsymbol{Q}(\boldsymbol{x}, t) = \bar{\boldsymbol{Q}}^t + \sum_{i=1}^{l-1} \hat{\boldsymbol{Q}}_i^t \left(\upsilon_i^t(\boldsymbol{x}) - \bar{\upsilon}_i^t\right) = \bar{\boldsymbol{Q}}^t + \sum_{i=1}^{l-1} \hat{\boldsymbol{Q}}_i^t \left(\upsilon_i^0(\boldsymbol{X}) - \frac{1}{|\Omega_e^t|} \int_{\Omega_e^t} \upsilon_i^0(\boldsymbol{X}) \mathrm{d}\boldsymbol{x}\right), \quad \boldsymbol{x} \in \Omega_e^t, \quad t \geq 0, \quad l \in \mathbb{N}, \tag{6}$$

in which $\bar{\boldsymbol{Q}}^t$ is the time-dependent cell averaged conservative variables, $|\Omega_e^t|$ is the time-dependent cell volume, and $\bar{\upsilon}_i^t$ is the time-dependent mean value of the i-th basis function within the cell Ω_e^t. Note that these basis functions are based on the time-independent reference cell, so the set of the basis is time-independent. However, due to the time-dependent cell shape, the cell mean value of the basis functions is time-dependent.

The fundamental ALE relation for the referential time derivation, Eulerian time derivative, and spatial gradient [10] is

$$\left.\frac{\partial Q}{\partial t}\right|_{\boldsymbol{X}} = \frac{\partial Q}{\partial t} + \boldsymbol{U}_g \bullet \nabla Q, \tag{7}$$

where Q is a scalar physical variable and $\boldsymbol{U}_g = (U_g, V_g, W_g)$ is the grid moving velocity. According to Eq. (7), the ALE formulation of Eq. (1) can be written as

$$\left.\frac{\partial \boldsymbol{Q}(\boldsymbol{x}, t)}{\partial t}\right|_{X} + \boldsymbol{Q} \nabla \bullet \boldsymbol{U}_g + \nabla \bullet (\boldsymbol{F} - \boldsymbol{Q}\boldsymbol{U}_g) = \boldsymbol{0}, \quad \boldsymbol{x} \in \Omega^t. \tag{8}$$

Using the Galerkin method, we have

$$\int_{\Omega_e^t} \left.\frac{\partial \boldsymbol{Q}(\boldsymbol{x}, t)}{\partial t}\right|_{X} \upsilon_j^t \mathrm{d}\boldsymbol{x} + \int_{\Omega_e^t} \boldsymbol{Q} \nabla \bullet \boldsymbol{U}_g \upsilon_j^t \mathrm{d}\boldsymbol{x} + \int_{\Omega_e^t} \nabla \bullet (\boldsymbol{F} - \boldsymbol{Q}\boldsymbol{U}_g) \upsilon_j^t \mathrm{d}\boldsymbol{x} = \boldsymbol{0}, \quad j = 0, 1, \cdots, l-1. \tag{9}$$

According to the definition of the basis function Eq. (5), the referential time derivation of the basis function equals to zero, i.e.,

$$\left.\frac{\partial \upsilon_j^t}{\partial t}\right|_{\boldsymbol{X}} = \frac{\mathrm{d}\upsilon_j^0(\boldsymbol{X})}{\mathrm{d}t} = 0, \tag{10}$$

which implies basis functions move with the grid motion. Therefore, the Reynolds transport theorem can be rewritten as

$$\int_{\Omega_e^t} \left.\frac{\partial(\boldsymbol{Q}\upsilon_j^t)}{\partial t}\right|_{\boldsymbol{X}} \mathrm{d}\boldsymbol{x} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\int_{\Omega_e^t} \boldsymbol{Q}\upsilon_j^t \mathrm{d}\boldsymbol{x}\right) - \int_{\Omega_e^t} \boldsymbol{Q}\upsilon_j^t \nabla \bullet \boldsymbol{U}_g \mathrm{d}\boldsymbol{x} = \int_{\Omega_e^t} \left.\frac{\partial \boldsymbol{Q}}{\partial t}\right|_{\boldsymbol{X}} \upsilon_j^t \mathrm{d}\boldsymbol{x}. \tag{11}$$

Substituting Eq. (11) into Eq. (9), via integration by parts, one has

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_{\Omega_e^t} \boldsymbol{Q}\,\upsilon_j^t \mathrm{d}\boldsymbol{x}\right) + \int_{\partial\Omega_e^t} \upsilon_j^t(\boldsymbol{F}-\boldsymbol{Q}\,\boldsymbol{U}_g)\bullet\boldsymbol{n}\mathrm{d}s - \int_{\Omega_e^t} \nabla\upsilon_j^t \bullet (\boldsymbol{F}-\boldsymbol{Q}\,\boldsymbol{U}_g)\mathrm{d}\boldsymbol{x} = \boldsymbol{0}, \tag{12}$$

where $\boldsymbol{n}$ is the outward normal unit vector of the corresponding cell interface $\partial\Omega_e^t$.

Finally, a DG method in the ALE formulation can be obtained by substituting the approximate solution (6) into the above expression,

$$\bar{\boldsymbol{Q}}^{\,n+1}\left|\Omega_e^{n+1}\right| = \bar{\boldsymbol{Q}}^{\,n}\left|\Omega_e^{n}\right| - \int_0^{\Delta t}\left(\int_{\partial\Omega_e^t} (\boldsymbol{F}-\boldsymbol{Q}\,\boldsymbol{U}_g)\bullet\boldsymbol{n}\mathrm{d}s\right)\mathrm{d}t, \tag{13}$$

$$\begin{aligned}\sum_{i=1}^{l-1} M_{ij}^{n+1}\,\hat{\boldsymbol{Q}}_i^{\,n+1} &= \sum_{i=1}^{l-1} M_{ij}^{n}\,\hat{\boldsymbol{Q}}_i^{\,n} + \bar{\boldsymbol{Q}}^{\,n}\bar{\upsilon}_j^n\left|\Omega_e^n\right| - \bar{\boldsymbol{Q}}^{\,n+1}\bar{\upsilon}_j^{n+1}\left|\Omega_e^{n+1}\right| \\ &+ \int_0^{\Delta t}\left(\int_{\Omega_e^t} \nabla\upsilon_j^t \bullet (\boldsymbol{F}-\boldsymbol{Q}\,\boldsymbol{U}_g)\mathrm{d}\boldsymbol{x}\right)\mathrm{d}t - \int_0^{\Delta t}\left(\int_{\partial\Omega_e^t} \upsilon_j^t(\boldsymbol{F}-\boldsymbol{Q}\,\boldsymbol{U}_g)\bullet\boldsymbol{n}\mathrm{d}s\right)\mathrm{d}t, \quad j=1,\cdots,l-1,\end{aligned} \tag{14}$$

in which $M_{ij}^n = \int_{\Omega_e^n}\upsilon_i^n\upsilon_j^n\mathrm{d}\boldsymbol{x} - \bar{\upsilon}_i^n\bar{\upsilon}_j^n|\Omega_e^n|$, $M_{ij}^{n+1} = \int_{\Omega_e^{n+1}}\upsilon_i^{n+1}\upsilon_j^{n+1}\mathrm{d}\boldsymbol{x} - \bar{\upsilon}_i^{n+1}\bar{\upsilon}_j^{n+1}|\Omega_e^{n+1}|$, and the superscripts "$n$" and "$n+1$" are the time step indexes.

One should note that the above expressions are integrated in time explicitly without using any time discretized method. The grid velocity $\boldsymbol{U}_g$ inside a cell and along a cell interface can be varied point by point. If we set the grid velocity to be zero, i.e., $\boldsymbol{U}_g = 0$, the method with static mesh in [22] can be exactly recovered. The critical problem is how to evaluate the flux terms in Eqs. (13) and (14) based on the gas kinetic scheme. In the previous gas kinetic scheme [23,24], a piecewise constant velocity is considered. However, for an arbitrary mesh motion, the grid velocity should not be constant inside the cell or along the cell interface. Based on our studying, a 2nd order method with a constant velocity assumption can give reasonable results. However, for higher order methods, the velocity variation inside a cell or along the cell interface must be considered. Thus, in the following, we will proposed a gas kinetic scheme with a mesh velocity distribution.

2.2. Gas kinetic flux scheme with a mesh velocity distribution

2.2.1. Gas kinetic flux inside a cell

The 3rd order NS gas distribution function within each control volume can be written as (see [22])

$$f(\boldsymbol{x},t,\boldsymbol{u},\xi) = f^{\mathrm{I}} + f^{\mathrm{V}}, \tag{15}$$

$$f^{\mathrm{I}} = g^{c0}\left(d_0^{c0} + d_1^{c0}\upsilon_1'' + d_2^{c0}\upsilon_2'' + d_3^{c0}\upsilon_3'' + b_{xy}^{c0}\upsilon_4'' + b_{xz}^{c0}\upsilon_5'' + b_{yz}^{c0}\upsilon_6'' + b_{xx}^{c0}\upsilon_7'' + b_{yy}^{c0}\upsilon_8'' + b_{zz}^{c0}\upsilon_9''\right), \tag{16}$$

$$f^{\mathrm{V}} = -\tau g^{c0}\left(D_0^{c0} + D_1^{c0}\upsilon_1'' + D_2^{c0}\upsilon_2'' + D_3^{c0}\upsilon_3''\right), \tag{17}$$

where $\boldsymbol{u} = (u,v,w)$ is the particle velocity vector, $\tau = \mu^{c0}/p^{c0}$ is the particle collision time, μ^{c0} is the dynamic viscosity, p^{c0} is the pressure, and $\upsilon_1'' = x - x^{c0}$, $\upsilon_2'' = y - y^{c0}$, $\upsilon_3'' = z - z^{c0}$, $\upsilon_4'' = \upsilon_1''\upsilon_2''$, $\upsilon_5'' = \upsilon_1''\upsilon_3''$, $\upsilon_6'' = \upsilon_2''\upsilon_3''$, $\upsilon_7'' = \frac{1}{2}\upsilon_1''\upsilon_1''$, $\upsilon_8'' = \frac{1}{2}\upsilon_2''\upsilon_2''$, $\upsilon_9'' = \frac{1}{2}\upsilon_3''\upsilon_3''$. The superscript "$c0$" indicates the initial barycenter of the cell for each time step. The equilibrium state g^{c0} is a Maxwellian distribution. All the coefficients, i.e., d_i^{c0} $(i=0,1,2,3)$, b_{xy}^{c0}, b_{xz}^{c0}, b_{yz}^{c0}, b_{xx}^{c0}, b_{yy}^{c0}, b_{zz}^{c0}, D_i^{c0} $(i=0,1,2,3)$, for their definitions and evaluations can be found in [22].

Due to the relations of the gas distribution function f and the macroscopic variables $\boldsymbol{F}$ and $\boldsymbol{Q}$, the volume integration in Eq. (14) can be expressed as

$$\begin{aligned}&\int_0^{\Delta t}\left(\int_{\Omega_e^t} \nabla\upsilon_j^t \bullet (\boldsymbol{F}-\boldsymbol{Q}\,\boldsymbol{U}_g)\mathrm{d}\boldsymbol{x}\right)\mathrm{d}t \\ &= \int_0^{\Delta t}\left(\int_{\Omega_e^t} \nabla\upsilon_j^t \bullet \left(\int \boldsymbol{u} f\boldsymbol{\psi}\mathrm{d}\boldsymbol{\Xi} - \boldsymbol{U}_g\int f\boldsymbol{\psi}\mathrm{d}\boldsymbol{\Xi}\right)\mathrm{d}\boldsymbol{x}\right)\mathrm{d}t \\ &= \int_0^{\Delta t}\left[\boldsymbol{F}_0\bullet\int_{\Omega_e^t}\nabla\upsilon_j^t\mathrm{d}\boldsymbol{x} - \boldsymbol{F}_{g,0}\int_{\Omega_e^t}\nabla\upsilon_j^t\bullet\boldsymbol{U}_g\mathrm{d}\boldsymbol{x} + \sum_{i=1}^{9}\left(\boldsymbol{F}_i\bullet\int_{\Omega_e^t}\upsilon_i''\nabla\upsilon_j^t\mathrm{d}\boldsymbol{x} - \boldsymbol{F}_{g,i}\int_{\Omega_e^t}\upsilon_i''\nabla\upsilon_j^t\bullet\boldsymbol{U}_g\mathrm{d}\boldsymbol{x}\right)\right]\mathrm{d}t,\end{aligned} \tag{18}$$

$$\begin{cases} \boldsymbol{F}_0 = \langle \boldsymbol{u} d_0^{c0} \rangle - \tau \langle \boldsymbol{u} D_0^{c0} \rangle, & \boldsymbol{F}_{g,0} = \langle d_0^{c0} \rangle - \tau \langle D_0^{c0} \rangle, \\ \boldsymbol{F}_1 = \langle \boldsymbol{u} d_1^{c0} \rangle - \tau \langle \boldsymbol{u} D_1^{c0} \rangle, & \boldsymbol{F}_{g,1} = \langle d_1^{c0} \rangle - \tau \langle D_1^{c0} \rangle, \\ \boldsymbol{F}_2 = \langle \boldsymbol{u} d_2^{c0} \rangle - \tau \langle \boldsymbol{u} D_2^{c0} \rangle, & \boldsymbol{F}_{g,2} = \langle d_2^{c0} \rangle - \tau \langle D_2^{c0} \rangle, \\ \boldsymbol{F}_3 = \langle \boldsymbol{u} d_3^{c0} \rangle - \tau \langle \boldsymbol{u} D_3^{c0} \rangle, & \boldsymbol{F}_{g,3} = \langle d_3^{c0} \rangle - \tau \langle D_3^{c0} \rangle, \\ \boldsymbol{F}_4 = \langle \boldsymbol{u} b_{xy}^{c0} \rangle, & \boldsymbol{F}_{g,4} = \langle b_{xy}^{c0} \rangle, \\ \boldsymbol{F}_5 = \langle \boldsymbol{u} b_{xz}^{c0} \rangle, & \boldsymbol{F}_{g,5} = \langle b_{xz}^{c0} \rangle, \\ \boldsymbol{F}_6 = \langle \boldsymbol{u} b_{yz}^{c0} \rangle, & \boldsymbol{F}_{g,6} = \langle b_{yz}^{c0} \rangle, \\ \boldsymbol{F}_7 = \langle \boldsymbol{u} b_{xx}^{c0} \rangle, & \boldsymbol{F}_{g,7} = \langle b_{xx}^{c0} \rangle, \\ \boldsymbol{F}_8 = \langle \boldsymbol{u} b_{yy}^{c0} \rangle, & \boldsymbol{F}_{g,8} = \langle b_{yy}^{c0} \rangle, \\ \boldsymbol{F}_9 = \langle \boldsymbol{u} b_{zz}^{c0} \rangle, & \boldsymbol{F}_{g,9} = \langle b_{zz}^{c0} \rangle, \end{cases} \tag{19}$$

in which the notation $\langle \cdots \rangle$ is defined as

$$\langle \cdots \rangle = \int g^{c0} (\cdots) \boldsymbol{\psi} \mathrm{d}\boldsymbol{\Xi}, \tag{20}$$

the vector of moments $\boldsymbol{\psi}$ is

$$\boldsymbol{\psi} = \left(1, u, v, w, \frac{1}{2}(u^2 + v^2 + w^2 + \xi^2)\right)^{\mathrm{T}}, \tag{21}$$

the internal variable $\xi^2 = \xi_1^2 + \xi_2^2 + \cdots + \xi_K^2$, the volume element in the phase space $\mathrm{d}\boldsymbol{\Xi} = \mathrm{d}u\mathrm{d}v\mathrm{d}w\mathrm{d}\xi_1\mathrm{d}\xi_2 \cdots \mathrm{d}\xi_K$, $K = (5 - 3\gamma)/(\gamma - 1)$ is the total degree of freedom in the internal variable, γ is the specific heat ratio.

It should be noted that the grid velocity $\boldsymbol{U}_g$ is a function of space and time, which cannot be moved outside of the space integration. In Section 2.4, we will show the method to define the grid velocity distribution. Besides, these space integrations, i.e., $\int_{\Omega_e^t} \nabla \upsilon_j^t \mathrm{d}\boldsymbol{x}$, $\int_{\Omega_e^t} \nabla \upsilon_j^t \bullet \boldsymbol{U}_g \mathrm{d}\boldsymbol{x}$, $\int_{\Omega_e^t} \upsilon_i'' \nabla \upsilon_j^t \mathrm{d}\boldsymbol{x}$, $\int_{\Omega_e^t} \upsilon_i'' \nabla \upsilon_j^t \bullet \boldsymbol{U}_g \mathrm{d}\boldsymbol{x}$, are time-dependent. Theoretically, they can be evaluated explicitly. However, it is quite complicated. Numerically, we can use the arithmetic mean values to represent them and the scheme can be rewritten as

$$\begin{aligned} \int_0^{\Delta t} \left(\int_{\Omega_e^t} \nabla \upsilon_j^t \bullet (\boldsymbol{F} - \boldsymbol{Q}\boldsymbol{U}_g) \mathrm{d}\boldsymbol{x} \right) \mathrm{d}t = & \int_0^{\Delta t} \boldsymbol{F}_0 \mathrm{d}t \bullet \overline{\nabla \upsilon_j} - \int_0^{\Delta t} \boldsymbol{F}_{g,0} \mathrm{d}t \overline{\nabla \upsilon_j \bullet \boldsymbol{U}_g} \\ & + \sum_{i=1}^{9} \left(\int_0^{\Delta t} \boldsymbol{F}_i \mathrm{d}t \bullet \overline{\upsilon_i'' \nabla \upsilon_j} - \int_0^{\Delta t} \boldsymbol{F}_{g,i} \mathrm{d}t \overline{\upsilon_i'' \nabla \upsilon_j \bullet \boldsymbol{U}_g} \right), \end{aligned} \tag{22}$$

with

$$\begin{cases} \overline{\nabla \upsilon_j} = \dfrac{1}{2} \left(\displaystyle\int_{\Omega_e^n} \nabla \upsilon_j^n \mathrm{d}\boldsymbol{x} + \int_{\Omega_e^{n+1}} \nabla \upsilon_j^{n+1} \mathrm{d}\boldsymbol{x} \right), \\ \overline{\nabla \upsilon_j \bullet \boldsymbol{U}_g} = \dfrac{1}{2} \left(\displaystyle\int_{\Omega_e^n} \nabla \upsilon_j^n \bullet \boldsymbol{U}_g \mathrm{d}\boldsymbol{x} + \int_{\Omega_e^{n+1}} \nabla \upsilon_j^{n+1} \bullet \boldsymbol{U}_g \mathrm{d}\boldsymbol{x} \right), \\ \overline{\upsilon_i'' \nabla \upsilon_j} = \dfrac{1}{2} \left(\displaystyle\int_{\Omega_e^n} (\upsilon_i'' \nabla \upsilon_j)^n \mathrm{d}\boldsymbol{x} + \int_{\Omega_e^{n+1}} (\upsilon_i'' \nabla \upsilon_j)^{n+1} \mathrm{d}\boldsymbol{x} \right), \\ \overline{\upsilon_i'' \nabla \upsilon_j \bullet \boldsymbol{U}_g} = \dfrac{1}{2} \left(\displaystyle\int_{\Omega_e^n} (\upsilon_i'' \nabla \upsilon_j)^n \bullet \boldsymbol{U}_g \mathrm{d}\boldsymbol{x} + \int_{\Omega_e^{n+1}} (\upsilon_i'' \nabla \upsilon_j)^{n+1} \bullet \boldsymbol{U}_g \mathrm{d}\boldsymbol{x} \right), \end{cases}$$

in which these integrals are evaluated using Gaussian Quadrature rules and the number of quadrature points depends on the order of approximation. A quadrature rule for the integral of any function $h(\boldsymbol{x})$ over a domain in the method is generally described as

$$\int_{\Omega_e} h(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \sum_i w_i h(\boldsymbol{x}_i), \tag{23}$$

where w_i denotes the weight and $\boldsymbol{x}_i$ is the coordinate of the i-th quadrature point.

2.2.2. Gas kinetic flux at a cell interface

The cell-interface-flux evaluation with static mesh has been introduced in [22]. Due to the general discontinuous initial condition, the moments of a Maxwellian distribution function with bounded integration limits are also needed besides the ones with unbounded integration limits. If the cell interface is moving or deforming, the bounded integration limits are different from the stationary coordinate system and the method in [22] can't be applied straightforwardly. In order to maintain the integration limits, which take account the particle transport across a moving cell interface, a flux evaluation scheme in a moving coordinate system [24] has been proposed. This previous scheme is based on the assumption that the mesh velocity is piecewise constant. It is a reasonable assumption for a 2nd order method, but not straightforward for a 3rd order method, which would lead to numerical oscillation and instability. To consider the mesh velocity distribution along the interface, we can write the mesh velocity for each point on the cell interface as

$$\boldsymbol{U}_g = \boldsymbol{U}_g^{fc} + \boldsymbol{U}_g^r, \tag{24}$$

in which $\boldsymbol{U}_g^{fc} = (U_g^{fc}, V_g^{fc}, W_g^{fc})$ is the grid velocity at the interface barycenter. The interface flux $\boldsymbol{F} - \boldsymbol{Q}\boldsymbol{U}_g$ in (12) is rewritten as

$$\boldsymbol{F} - \boldsymbol{Q}\boldsymbol{U}_g = \bar{\boldsymbol{F}} - \boldsymbol{Q}\boldsymbol{U}_g^r. \tag{25}$$

In the right hand side, the first term is the mean flux with a piecewise constant mesh velocity $\boldsymbol{U}_g^{fc}$. For the second term, it is a correction on each interface point, which means the difference between the real flux crossing each interface point and the mean flux $\bar{\boldsymbol{F}}$.

Firstly, we can evaluate the flux standing on a moving coordinate system with a moving velocity $\boldsymbol{U}_g^{fc}$. In this moving coordinate system, the initial transformation from $\boldsymbol{Q}$ to $\boldsymbol{Q}^r$ needs to take the moving velocity into account as

$$\boldsymbol{Q} = \left(\rho, \rho U, \rho V, \rho W, \frac{1}{2}\rho\left[(U)^2 + (V)^2 + (W)^2\right] + \rho e\right)^{\mathrm{T}}, \tag{26}$$

$$\boldsymbol{Q}^r = \left(\rho, \rho U^r, \rho V^r, \rho W^r, \frac{1}{2}\rho\left[\left(U^r\right)^2 + \left(V^r\right)^2 + \left(W^r\right)^2\right] + \rho e\right)^{\mathrm{T}}, \tag{27}$$

in which $\rho$and ρe are the density and thermal energy which have the same values in both coordinate systems, $\boldsymbol{U}^r = \boldsymbol{U} - \boldsymbol{U}_g^{fc}$ is the relative velocity in the moving coordinate system. Based on the initial condition $\boldsymbol{Q}^r$, the gas distribution function f in the moving reference of frame can be written as (see [22])

$$f\left(\boldsymbol{x}, t, \boldsymbol{u}^r, \xi\right) = f^S + f^D, \tag{28}$$

$$\begin{aligned} f^S \;=\; & g^{fc0}(d_0^{fc0} + d_1^{fc0}\upsilon_1''' + d_2^{fc0}\upsilon_2''' + d_3^{fc0}\upsilon_3''' + b_{xy}^{fc0}\upsilon_4''' + b_{xz}^{fc0}\upsilon_5''' + b_{yz}^{fc0}\upsilon_6''' + b_{xx}^{fc0}\upsilon_7''' + b_{yy}^{fc0}\upsilon_8''' + b_{zz}^{fc0}\upsilon_9''') \\ & - \tau g^{fc0}(D_0^{fc0} + D_1^{fc0}\upsilon_1''' + D_2^{fc0}\upsilon_2''' + D_3^{fc0}\upsilon_3'''), \end{aligned} \tag{29}$$

$$\begin{aligned} f^D \;=\; & -g^{fc0}e^{-t/\tau_n}\left[\begin{array}{l} 1 - (t+\tau)B_0^{fc0} + (\frac{1}{2}t+\tau)t(uB_4^{fc0} + vB_5^{fc0} + wB_6^{fc0}) - \tau A_t^{fc0} + \tau t B_2^{fc0} \\ \quad + (a_x^{fc0} - tB_4^{fc0} - \tau D_1^{fc0})\upsilon_1''' + (a_y^{fc0} - tB_5^{fc0} - \tau D_2^{fc0})\upsilon_2''' \\ \quad + (a_z^{fc0} - tB_6^{fc0} - \tau D_3^{fc0})\upsilon_3''' \\ \quad + (b_{xy}^{fc0}\upsilon_4''' + b_{xz}^{fc0}\upsilon_5''' + b_{yz}^{fc0}\upsilon_6''' + b_{xx}^{fc0}\upsilon_7''' + b_{yy}^{fc0}\upsilon_8''' + b_{zz}^{fc0}\upsilon_9''') \end{array}\right] \\ & + g^{*fc0}e^{-t/\tau_n}\left[\begin{array}{l} 1 - (t+\tau)B_0^{*fc0} + (\frac{1}{2}t+\tau)t(uB_4^{*fc0} + vB_5^{*fc0} + wB_6^{*fc0}) - \tau A_t^{*fc0} + \tau t B_2^{*fc0} \\ \quad + (a_x^{*fc0} - tB_4^{*fc0} - \tau D_1^{*fc0})\upsilon_1''' + (a_y^{*fc0} - tB_5^{*fc0} - \tau D_2^{*fc0})\upsilon_2''' \\ \quad + (a_z^{*fc0} - tB_6^{*fc0} - \tau D_3^{*fc0})\upsilon_3''' \\ \quad + (b_{xy}^{*fc0}\upsilon_4''' + b_{xz}^{*fc0}\upsilon_5''' + b_{yz}^{*fc0}\upsilon_6''' + b_{xx}^{*fc0}\upsilon_7''' + b_{yy}^{*fc0}\upsilon_8''' + b_{zz}^{*fc0}\upsilon_9''') \end{array}\right], \end{aligned} \tag{30}$$

in which

$$\begin{aligned} & g^{*fc0} = g^{Lfc0}H\left[u_n^r\right] + g^{Rfc0}\left(1 - H\left[u_n^r\right]\right), \\ & g^{*fc0}(\cdots)^{*fc0} = g^{Lfc0}(\cdots)^{Lfc0}H\left[u_n^r\right] + g^{Rfc0}(\cdots)^{Rfc0}\left(1 - H\left[u_n^r\right]\right), \end{aligned} \tag{31}$$

$u_n^r = \boldsymbol{u}^r \bullet \boldsymbol{n}$ is the relative particle velocity in the normal direction of the interface, $H[u_n^r]$ is the Heaviside function defined as

$$H\left[u_n^r\right] = \begin{cases} 0, & u_n^r \le 0, \\ 1, & u_n^r > 0. \end{cases} \tag{32}$$

$\tau = \mu^{fc0}/p^{fc0}$ is the physical collision time, μ^{fc0} is the dynamic viscosity, and p^{fc0} is the pressure. τ_n is the numerical collision time [22], which controls the contributions of the non-equilibrium state and the equilibrium one in the discontinuous part f^D of the gas distribution function. $\upsilon_1''' = x - x^{fc0}$, $\upsilon_2''' = y - y^{fc0}$, $\upsilon_3''' = z - z^{fc0}$, $\upsilon_4''' = \upsilon_1'''\upsilon_2'''$, $\upsilon_5''' = \upsilon_1'''\upsilon_3'''$, $\upsilon_6''' = \upsilon_2'''\upsilon_3'''$, $\upsilon_7''' = \frac{1}{2}\upsilon_1'''\upsilon_1'''$, $\upsilon_8''' = \frac{1}{2}\upsilon_2'''\upsilon_2'''$, $\upsilon_9''' = \frac{1}{2}\upsilon_3'''\upsilon_3'''$. The superscript "$fc0$" indicates the initial barycenter of the cell interface for each time step, "R" indicates the right side of the cell interface, and "L" is the left side. The Maxwellian distribution g^{Rfc0}, g^{Lfc0}, g^{fc0}, and other parameters, i.e., $d_i^{fc0}(i=0,1,2,3)$, b_{xy}^{fc0}, b_{xz}^{fc0}, b_{yz}^{fc0}, b_{xx}^{fc0}, b_{yy}^{fc0}, b_{zz}^{fc0}, b_{xy}^{*fc0}, b_{xz}^{*fc0}, b_{yz}^{*fc0}, b_{xx}^{*fc0}, b_{yy}^{*fc0}, b_{zz}^{*fc0}, $D_i^{fc0}(i=0,1,2,3)$, $B_i^{fc0}(i=0,2,4,5,6)$, $B_i^{*fc0}(i=0,2,4,5,6)$, can be evaluated by using the method proposed in [22] which will not be repeated here.

In this paper, a simplified discontinuous part for providing numerical dissipation is used for the flux evaluation, leading to a more efficient scheme, as

$$f^D = -g^{fc0}e^{-t/\tau_n} + g^{*fc0}e^{-t/\tau_n}. \tag{33}$$

Actually, just the discontinuous jump of the pointwise values is considered in the simplified scheme and all spatial and temporal variation terms are removed from the complete formulation Eq. (30), which saves computational time greatly, but keeps the same accuracy as the original one in all test cases in this paper. The flux standing on the moving coordinate system can be written as

$$\tilde{\boldsymbol{F}}^r - \boldsymbol{Q}^r\boldsymbol{U}_g^r = \left(\tilde{\boldsymbol{F}}_1^r, \tilde{\boldsymbol{F}}_2^r, \tilde{\boldsymbol{F}}_3^r, \tilde{\boldsymbol{F}}_4^r, \tilde{\boldsymbol{F}}_5^r\right)^{\mathrm{T}} = \int \boldsymbol{u}^r f \boldsymbol{\psi}^r \mathrm{d}\Xi^r - \boldsymbol{U}_g^r \int f\boldsymbol{\psi}^r \mathrm{d}\Xi^r, \tag{34}$$

where $\boldsymbol{\psi}^r = (1, u^r, v^r, w^r, \frac{1}{2}((u^r)^2 + (v^r)^2 + (w^r)^2 + \xi^2))^{\mathrm{T}}$, $\mathrm{d}\Xi^r = \mathrm{d}u^r\mathrm{d}v^r\mathrm{d}w^r\mathrm{d}\xi_1\cdots\mathrm{d}\xi_K$.

Then, the fluxes across the moving cell interface in the stationary inertia reference of frame for all computational cells, which will be used in Eq. (13) and (14), are

$$\begin{aligned}\boldsymbol{F} - \boldsymbol{Q}\,\boldsymbol{U}_g = \bar{\boldsymbol{F}} - \boldsymbol{Q}\,\boldsymbol{U}_g^r &= \int \boldsymbol{u}^r f\boldsymbol{\psi}\,\mathrm{d}\Xi^r - \boldsymbol{U}_g^r \int f\boldsymbol{\psi}\,\mathrm{d}\Xi^r \\ &= \begin{pmatrix} \tilde{\boldsymbol{F}}_1^r \\ \tilde{\boldsymbol{F}}_2^r + U_g^{fc}\tilde{\boldsymbol{F}}_1^r \\ \tilde{\boldsymbol{F}}_3^r + V_g^{fc}\tilde{\boldsymbol{F}}_1^r \\ \tilde{\boldsymbol{F}}_4^r + W_g^{fc}\tilde{\boldsymbol{F}}_1^r \\ \tilde{\boldsymbol{F}}_5^r + U_g^{fc}\tilde{\boldsymbol{F}}_2^r + V_g^{fc}\tilde{\boldsymbol{F}}_3^r + W_g^{fc}\tilde{\boldsymbol{F}}_4^r + \frac{1}{2}(\boldsymbol{U}_g^{fc}\bullet\boldsymbol{U}_g^{fc})\tilde{\boldsymbol{F}}_1^r \end{pmatrix}.\end{aligned} \tag{35}$$

Again, the time-dependent basis function integrations are defined as

$$\begin{cases} \overline{\upsilon_j\boldsymbol{n}} = \dfrac{1}{2}\left(\displaystyle\int_{\partial\Omega_e^n} \upsilon_j^n\boldsymbol{n}^n\mathrm{d}s + \int_{\partial\Omega_e^{n+1}} \upsilon_j^{n+1}\boldsymbol{n}^{n+1}\mathrm{d}s\right), \\ \overline{\upsilon_j\boldsymbol{U}_g\bullet\boldsymbol{n}} = \dfrac{1}{2}\left(\displaystyle\int_{\partial\Omega_e^n} \upsilon_j^n\boldsymbol{U}_g\bullet\boldsymbol{n}^n\mathrm{d}s + \int_{\partial\Omega_e^{n+1}} \upsilon_j^{n+1}\boldsymbol{U}_g\bullet\boldsymbol{n}^{n+1}\mathrm{d}s\right), \\ \overline{\upsilon_i'''\upsilon_j\boldsymbol{n}} = \dfrac{1}{2}\left(\displaystyle\int_{\partial\Omega_e^n} (\upsilon_i'''\upsilon_j)^n\boldsymbol{n}^n\mathrm{d}s + \int_{\partial\Omega_e^{n+1}} (\upsilon_i'''\upsilon_j)^{n+1}\boldsymbol{n}^{n+1}\mathrm{d}s\right), \\ \overline{\upsilon_i'''\upsilon_j\boldsymbol{U}_g\bullet\boldsymbol{n}} = \dfrac{1}{2}\left(\displaystyle\int_{\partial\Omega_e^n} (\upsilon_i'''\upsilon_j)^n\boldsymbol{U}_g\bullet\boldsymbol{n}^n\mathrm{d}s + \int_{\partial\Omega_e^{n+1}} (\upsilon_i'''\upsilon_j)^{n+1}\boldsymbol{U}_g\bullet\boldsymbol{n}^{n+1}\mathrm{d}s\right), \end{cases} \tag{36}$$

in which these integrals are evaluated using Gaussian Quadrature rules and the number of quadrature points depends on the order of approximation and the quadrature rule used.

2.3. One-to-one mapping

The initial domain is taken as the referential domain in this paper and the referential coordinates are equal to the initial space coordinates, i.e., $\boldsymbol{X} = \boldsymbol{x}^0 = (x^0, y^0, z^0)$.

In the following, we will introduce the method to acquire the one-to-one mapping φ^t and its inverse mapping ϕ^t. For the sake of brevity, here we just take the hexahedral element as an example. The time-dependent one-to-one mappings φ^t and ϕ^t are based on a transformation from an arbitrary hexahedral cell to a regular hexahedron as shown in Fig. 1.

Considering the trilinear mapping $\mathcal{A}$ with the properties

$$\mathcal{A}(\boldsymbol{\chi}_i) = (a_1 + b_1\chi_i)(a_2 + b_2\eta_i)(a_3 + b_3\zeta_i) = \boldsymbol{x}_i, \quad i = 1, 2, \cdots, 8, \tag{37}$$

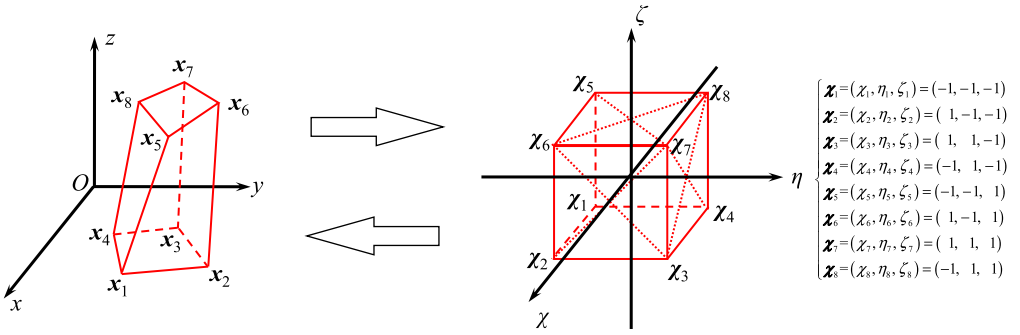

Fig. 1. An arbitrary hexahedral cell to a regular hexahedron.

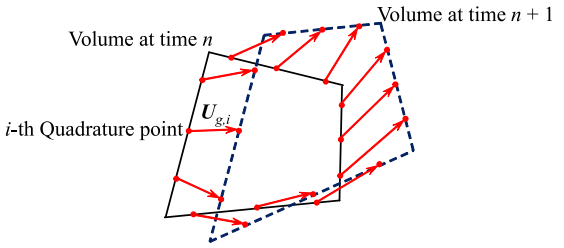

Fig. 2. A sketch of the mesh velocity distribution on the cell interfaces.

in which a_1, a_2, a_3, b_1, b_2, b_3 are unknown coefficients, we have

$$\boldsymbol{x} = \mathcal{A}(\boldsymbol{\chi}) = \sum_{i=1}^{8} N_i \boldsymbol{x}_i, \tag{38}$$

with the shape functions $N_1 = \frac{1}{8}(1-\chi)(1-\eta)(1-\zeta)$, $N_2 = \frac{1}{8}(1+\chi)(1-\eta)(1-\zeta)$, $N_3 = \frac{1}{8}(1+\chi)(1+\eta)(1-\zeta)$, $N_4 = \frac{1}{8}(1-\chi)(1+\eta)(1-\zeta)$, $N_5 = \frac{1}{8}(1-\chi)(1-\eta)(1+\zeta)$, $N_6 = \frac{1}{8}(1+\chi)(1-\eta)(1+\zeta)$, $N_7 = \frac{1}{8}(1+\chi)(1+\eta)(1+\zeta)$, $N_8 = \frac{1}{8}(1-\chi)(1+\eta)(1+\zeta)$. Meanwhile, the inverse mapping $\mathcal{B} = \mathcal{A}^{-1}$ can be obtained easily. Then, the one-to-one mappings can be defined as

$$\boldsymbol{x}^t = \varphi^t(\boldsymbol{X}) = \varphi^t(\boldsymbol{x}^0) = \mathcal{A}^t\big(\mathcal{B}^0(\boldsymbol{x}^0)\big), \tag{39}$$

$$\boldsymbol{X} = \boldsymbol{x}^0 = \phi^t(\boldsymbol{x}^t) = \mathcal{A}^0\big(\mathcal{B}^t(\boldsymbol{x}^t)\big), \tag{40}$$

in which $\boldsymbol{x}^t = \mathcal{A}^t(\boldsymbol{\chi})$, $\mathcal{B}^t(\boldsymbol{x}^t) = \boldsymbol{\chi}$, $\mathcal{B}^0(\boldsymbol{x}^0) = \boldsymbol{\chi}$, $\boldsymbol{x}^0 = \mathcal{A}^0(\boldsymbol{\chi})$, $\boldsymbol{x}^t \in \Omega_e^t$, $\boldsymbol{X} = \boldsymbol{x}^0 \in \Omega_e^0$, $\forall e \in \{1, 2, \cdots, m\}$.

2.4. Grid velocity distribution

In the practice, the positions of grid nodes at the moving boundary Γ^t are user prescribed or obtained from the structure response in applications of fluid–structure interaction at each time step. With the definition of the boundary nodes displacements as $\Delta\boldsymbol{x} = \boldsymbol{x}^{\Gamma^n} - \boldsymbol{x}^{\Gamma^0}$, various methods [25–28] can be used to compute the new positions of the inner nodes at each time step. In this paper, we adopt the method based on radial basis functions [28] to set the mesh nodes motion.

After the mesh nodes positions are determined, the one-to-one mapping mentioned above can be used to determine the quadrature points at time n and $n+1$ which have the same referential quadrature point as

$$\boldsymbol{x}_i^n = \mathcal{A}^n\big(\mathcal{B}^0(\boldsymbol{x}_i^0)\big), \tag{41}$$

$$\boldsymbol{x}_i^{n+1} = \mathcal{A}^{n+1}\big(\mathcal{B}^0(\boldsymbol{x}_i^0)\big). \tag{42}$$

The grid velocity for i-th quadrature point is defined as

$$\boldsymbol{U}_{g,i} = \frac{\boldsymbol{x}_i^{n+1} - \boldsymbol{x}_i^n}{\Delta t} = \frac{\mathcal{A}^{n+1}(\mathcal{B}^0(\boldsymbol{x}_i^0)) - \mathcal{A}^n(\mathcal{B}^0(\boldsymbol{x}_i^0))}{\Delta t}. \tag{43}$$

Based on the above grid velocity definition on each quadrature point, the grid velocity distribution is obtained. Fig. 2 shows a mesh velocity distribution on the cell interfaces.

2.5. Geometric conservation law (GCL)

Geometric conservation law (GCL) was originally discussed by Thomas and Lombard [29]. Generally, the preservation of a constant solution is often referred to as the GCL, which means that a uniform flow on a grid that moves arbitrarily in space as a function of time must be preserved for any numerical discretization scheme [15,16,30]. Here, we show the method proposed in this paper can preserve the uniform flow. Giving a constant solution $\boldsymbol{Q}(x,t)=\boldsymbol{Q}_0$, the flux inside a cell is rewritten as

$$\int_{\Omega_e^t}\nabla\upsilon_j^t\bullet(\boldsymbol{F}-\boldsymbol{Q}\boldsymbol{U}_g)\mathrm{d}\boldsymbol{x}=\boldsymbol{F}(\boldsymbol{Q}_0)\bullet\int_{\Omega_e^t}\nabla\upsilon_j^t\mathrm{d}\boldsymbol{x}-\boldsymbol{Q}_0\int_{\Omega_e^t}\nabla\upsilon_j^t\bullet\boldsymbol{U}_g\mathrm{d}\boldsymbol{x}, \tag{44}$$

in which $\boldsymbol{F}(\boldsymbol{Q}_0)=\int\boldsymbol{u}g(\boldsymbol{Q}_0)\boldsymbol{\psi}\mathrm{d}\Xi$, $\boldsymbol{Q}_0=\int g(\boldsymbol{Q}_0)\boldsymbol{\psi}\mathrm{d}\Xi$. The flux at a cell interface is

$$\int_{\partial\Omega_e^t}\upsilon_j^t(\boldsymbol{F}-\boldsymbol{Q}\boldsymbol{U}_g)\bullet\boldsymbol{n}\mathrm{d}s=\boldsymbol{F}(\boldsymbol{Q}_0)\bullet\int_{\partial\Omega_e^t}\upsilon_j^t\boldsymbol{n}\mathrm{d}s-\boldsymbol{Q}_0\int_{\partial\Omega_e^t}\upsilon_j^t\boldsymbol{U}_g\bullet\boldsymbol{n}\mathrm{d}s. \tag{45}$$

Then, Eq. (12) has the following expression

$$\boldsymbol{Q}_0\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_e^t}\upsilon_j^t\mathrm{d}\boldsymbol{x}=\boldsymbol{F}(\boldsymbol{Q}_0)\bullet\left(\int_{\Omega_e^t}\nabla\upsilon_j^t\mathrm{d}\boldsymbol{x}-\int_{\partial\Omega_e^t}\upsilon_j^t\boldsymbol{n}\mathrm{d}s\right)-\boldsymbol{Q}_0\left(\int_{\Omega_e^t}\nabla\upsilon_j^t\bullet\boldsymbol{U}_g\mathrm{d}\boldsymbol{x}-\int_{\partial\Omega_e^t}\upsilon_j^t\boldsymbol{U}_g\bullet\boldsymbol{n}\mathrm{d}s\right). \tag{46}$$

The first term on the right hand side is equal to zero according to the divergence theorem and Eq. (46) is simplified as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_e^t}\upsilon_j^t\mathrm{d}\boldsymbol{x}+\int_{\Omega_e^t}\nabla\upsilon_j^t\bullet\boldsymbol{U}_g\mathrm{d}\boldsymbol{x}-\int_{\partial\Omega_e^t}\upsilon_j^t\boldsymbol{U}_g\bullet\boldsymbol{n}\mathrm{d}s=0. \tag{47}$$

The third term in the above equation can be further expanded as

$$\int_{\partial\Omega_e^t}\upsilon_j^t\boldsymbol{U}_g\bullet\boldsymbol{n}\mathrm{d}s=\int_{\Omega_e^t}\nabla\bullet\left(\upsilon_j^t\boldsymbol{U}_g\right)\mathrm{d}\boldsymbol{x}=\int_{\Omega_e^t}\nabla\upsilon_j^t\bullet\boldsymbol{U}_g\mathrm{d}\boldsymbol{x}+\int_{\Omega_e^t}\upsilon_j^t\nabla\bullet\boldsymbol{U}_g\mathrm{d}\boldsymbol{x}. \tag{48}$$

Substituting Eq. (48) into (47), we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_e^t}\upsilon_j^t\mathrm{d}\boldsymbol{x}-\int_{\Omega_e^t}\upsilon_j^t\nabla\bullet\boldsymbol{U}_g\mathrm{d}\boldsymbol{x}=0. \tag{49}$$

It is clearly seen that Eq. (49) is always satisfied due to the Reynolds transport theorem Eq. (11), which is based on the basis function definition in this paper. Thus, GCL is satisfied.

Numerically, we consider a uniform inviscid compressible flow in the domain $[0,10]\times[-5,5]\times[0,1]$ to validate our method. The initial condition is

$$\begin{cases}\rho(x,y,z,0)=1.0,\\ U(x,y,z,0)=0.3,\\ V(x,y,z,0)=0.7,\\ W(x,y,z,0)=0.0,\\ p(x,y,z,0)=1.0.\end{cases} \tag{50}$$

A perfect gas with specific heat ratio $\gamma=1.4$ is considered. Periodic boundary conditions are applied in both x and y directions, and a symmetric boundary condition is applied in the z direction. Three typical types of mesh motions are used for checking the GCL: a rigid translation Eq. (51), a rigid rotation Eq. (52) and a sinusoidal deformation Eq. (53), as following

$$x(t)=x^0+\frac{1}{2}\sin\left(\frac{2\pi t}{t^0}\right),\qquad y(t)=y^0+\frac{1}{2}\sin\left(\frac{2\pi t}{t^0}\right),\qquad z(t)=z^0, \tag{51}$$

$$\begin{aligned}x(t)&=\left(x^0-5\right)\cos\left(\frac{t}{5t^0}\right)-y^0\sin\left(\frac{t}{5t^0}\right)+5,\\ y(t)&=\left(x^0-5\right)\sin\left(\frac{t}{5t^0}\right)+y^0\cos\left(\frac{t}{5t^0}\right),\quad z(t)=z^0,\end{aligned} \tag{52}$$

$$\begin{aligned}x(t)&=x^0+\frac{1}{2}\sin\left(\frac{\pi x^0}{5}\right)\sin\left(\frac{\pi y^0}{5}\right)\sin\left(\frac{2\pi t}{t^0}\right),\\ y(t)&=y^0+\frac{1}{2}\sin\left(\frac{\pi x^0}{5}\right)\sin\left(\frac{\pi y^0}{5}\right)\sin\left(\frac{2\pi t}{t^0}\right),\qquad z(t)=z^0,\end{aligned} \tag{53}$$

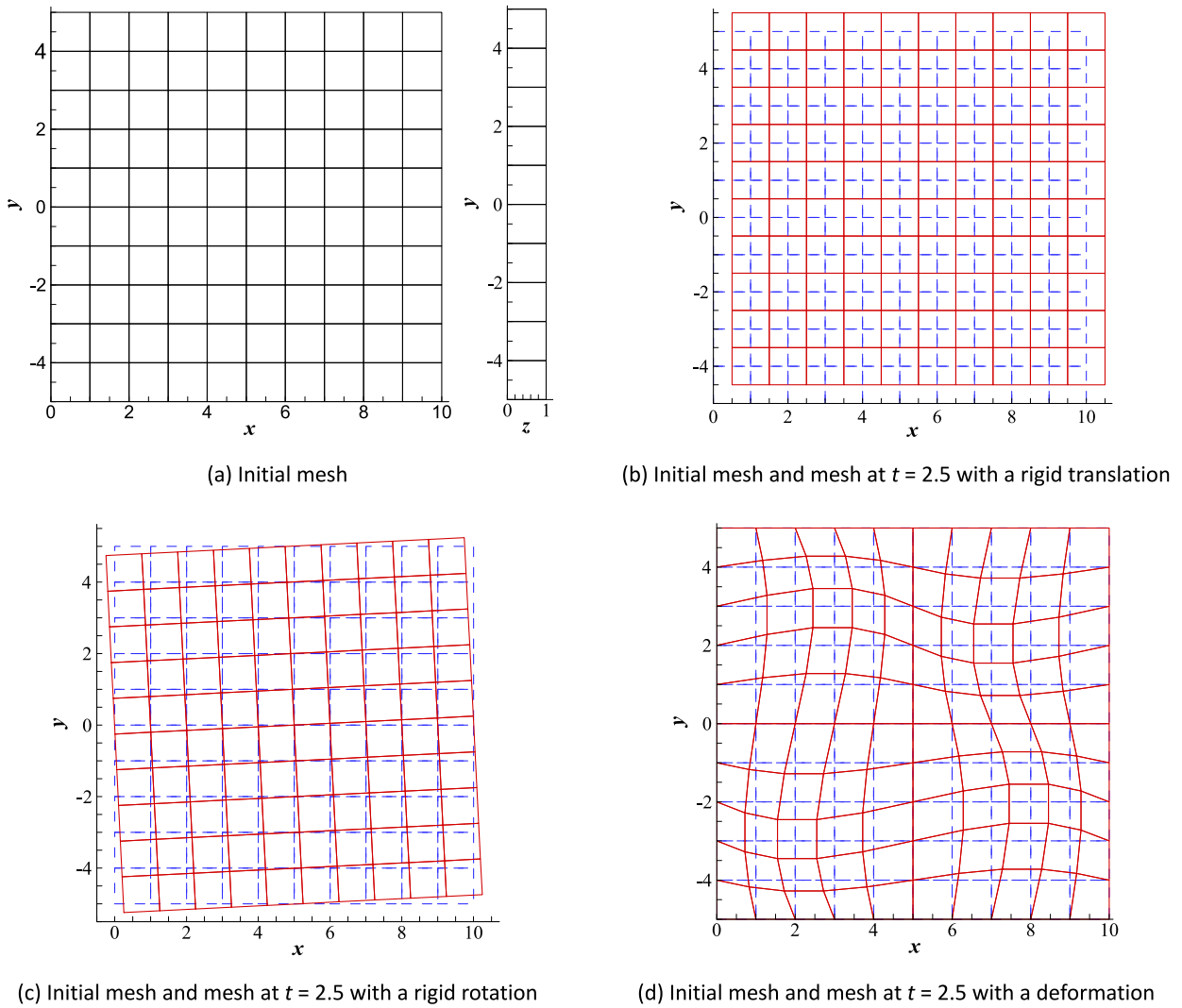

Fig. 3. The sample gird with 10 cells in x and y direction, and 1 cell in z direction.

in which $t^0 = 10$, (x^0, y^0, z^0) are the initial grid coordinates at $t = 0$, and (x, y, z) are the new coordinates which are time dependent. A sample uniform grid is used here for the calculation, see Fig. 3. The grid in the x–y plane is refined by quartering all cells for the convergence study, while one layer of grid in the z direction is kept. The spatial errors of the density ρ, the velocity U, and the pressure p at $t = t^0$ are shown in Table 1. The results with static mesh are also listed there for reference. It is shown that all errors are significantly smaller than the previous results [16,30], which implies the satisfaction of the geometric conservation law.

Table 1
Errors of the numerical solutions versus free flow condition.

Meshes	h	Density		U velocity		Pressure	
		L^1 error	L^∞ error	L^1 error	L^∞ error	L^1 error	L^∞ error
Static	1.000	8.46e−16	2.69e−15	1.08e−15	9.76e−15	8.46e−15	1.49e−14
	0.500	7.79e−16	3.70e−15	5.38e−14	9.44e−13	2.45e−14	3.08e−13
	0.250	1.95e−12	8.75e−12	1.31e−12	6.11e−12	9.65e−13	4.55e−12
	0.125	2.14e−12	9.06e−12	1.39e−12	5.58e−12	8.64e−13	4.02e−12
A rigid translation	1.000	5.17e−13	1.73e−12	3.93e−13	1.09e−12	2.91e−13	9.15e−13
	0.500	7.24e−13	2.33e−12	5.23e−13	1.67e−12	3.17e−13	1.11e−12
	0.250	8.79e−13	2.93e−12	6.04e−13	1.98e−12	3.91e−13	1.84e−12
	0.125	1.35e−12	4.93e−12	9.54e−13	3.44e−12	5.83e−13	3.48e−12

(*continued on next page*)

Table 1 (*continued*)

Meshes	h	Density		$\boldsymbol{U}$ velocity		Pressure	
		L^1 error	L^∞ error	L^1 error	L^∞ error	L^1 error	L^∞ error
A rigid rotation	1.000	4.54e−13	1.69e−12	3.45e−13	1.08e−12	2.49e−13	1.02e−12
	0.500	8.48e−13	2.18e−12	5.95e−13	1.65e−12	3.35e−13	1.39e−12
	0.250	1.07e−12	2.91e−12	7.33e−13	1.98e−12	3.53e−13	1.56e−12
	0.125	1.61e−12	3.98e−12	1.13e−12	3.59e−12	4.99e−13	2.62e−12
A sinusoidal deformation	1.000	1.38e−12	3.95e−12	1.01e−12	2.68e−12	3.84e−13	1.57e−12
	0.500	1.34e−12	4.24e−12	8.57e−13	2.65e−12	5.97e−13	2.22e−12
	0.250	1.15e−12	3.93e−12	7.93e−13	2.41e−12	4.89e−13	2.33e−12
	0.125	1.18e−12	4.55e−12	8.03e−13	3.10e−12	5.54e−13	3.10e−12

3. Numerical examples

3.1. Accuracy test – an isentropic vortex in freestream

A compressible isentropic vortex propagating in freestream is used for the accuracy test in this paper. Similar test cases have also been used by several authors [31–34]. It can be described as a perturbation added to a mean flow. The mean flow velocity $\boldsymbol{U}_\infty = (U_\infty, V_\infty, W_\infty)$, pressure p_∞, and density ρ_∞ take the freestream condition. Initially, an isentropic vortex with no perturbation in entropy is added to the mean flow field. The initial conditions are given by

$$\begin{cases} U(x,y,z,0) = U_\infty + \delta U(x,y,z,0) = U_\infty - \dfrac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} (y - y_c), \\ V(x,y,z,0) = V_\infty + \delta V(x,y,z,0) = V_\infty + \dfrac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} (x - x_c), \\ W(x,y,z,0) = W_\infty + \delta W(x,y,z,0) = 0.0, \\ T(x,y,z,0) = \dfrac{p_\infty}{\rho_\infty} - \delta T(x,y,z,0) = \dfrac{p_\infty}{\rho_\infty} - \dfrac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2}, \\ S(x,y,z,0) = \dfrac{p_\infty}{\rho_\infty^\gamma} + \delta S(x,y,z,0) = \dfrac{p_\infty}{\rho_\infty^\gamma}, \end{cases} \tag{54}$$

where (x_c, y_c, z) are the vortex center coordinates, $r^2 = (x - x_c)^2 + (y - y_c)^2$, the specific heat ratio $\gamma = 1.4$, and the vortex strength $\varepsilon = 5.0$. The first case is a stationary vortex with the mean flow velocity to be zero, and the second case is a diagonally convecting vortex with the mean flow velocity $\boldsymbol{U}_\infty = (1.0, 1.0, 0.0)$. For both cases, $\rho_\infty = 1.0$ and $p_\infty = 1.0$.

The computational domain Ω is taken as $[0, 10] \times [-5, 5] \times [0, 1]$, and the vortex is centered at $(x_c, y_c, z) = (5.0, 0.0, z)$ initially. Periodic boundary conditions are applied in both x and y directions, and the symmetry boundary condition is applied in the z direction. As a results of isentropy, the exact solution corresponds to a pure advection of the vortex with the mean flow velocity. The grids are the same as ones used for GCL test. Again, the three types of mesh motions, i.e., Eq. (51)–(53), are considered here. The spatial errors and the orders of accuracy are computed at $t = 10$ (one time period), which are shown in Table 2 and Table 3. The results show that the mesh moving or deforming leads to additional errors, however, the method is indeed a 3rd order accuracy, regardless the mesh movement.

Table 2
Errors and orders – stationary vortex.

Meshes	h	Density		$\boldsymbol{U}$ velocity		Pressure	
		L^1		L^1		L^1	
		errors	orders	errors	orders	errors	orders
Static	1.000	2.03e−1	–	5.70e−1	–	1.90e−1	–
	0.500	2.67e−2	*2.92*	5.95e−2	*3.26*	2.72e−2	*2.80*
	0.250	1.85e−3	*3.85*	4.97e−3	*3.58*	1.73e−3	*3.98*
	0.125	9.42e−5	*4.30*	2.36e−4	*4.40*	9.10e−5	*4.24*
A rigid translation	1.000	9.48e−1	–	1.18e+0	–	9.56e−1	–
	0.500	8.56e−2	*3.47*	1.08e−1	*3.45*	1.04e−1	*3.20*
	0.250	6.11e−3	*3.81*	1.10e−2	*3.29*	7.10e−3	*3.87*
	0.125	3.34e−4	*4.19*	5.97e−4	*4.21*	4.00e−4	*4.15*
A rigid rotation	1.000	2.58e−1	–	5.55e−1	–	2.13e−1	–
	0.500	2.87e−2	*3.17*	6.44e−2	*3.11*	2.93e−2	*2.86*
	0.250	1.91e−3	*3.91*	6.21e−3	*3.37*	1.70e−3	*4.11*
	0.125	9.51e−5	*4.33*	3.81e−4	*4.03*	8.11e−5	*4.39*

Table 2 (*continued*)

Meshes	h	Density		$\boldsymbol{U}$ velocity		Pressure	
		L^1		L^1		L^1	
		errors	orders	errors	orders	errors	orders
A sinusoidal deformation	1.000	4.95e−1	–	8.87e−1	–	4.23e−1	–
	0.500	6.89e−2	*2.84*	1.26e−1	*2.82*	5.98e−2	*2.82*
	0.250	6.55e−3	*3.40*	9.94e−3	*3.66*	3.93e−3	*3.93*
	0.125	3.29e−4	*4.31*	6.10e−4	*4.03*	2.03e−4	*4.28*

Table 3
Errors and orders – diagonally convecting vortex.

Meshes	h	Density		$\boldsymbol{U}$ velocity		Pressure	
		L^1		L^1		L^1	
		errors	orders	errors	orders	errors	orders
Static	1.000	2.90e−1	–	6.56e−1	–	4.02e−1	–
	0.500	2.75e−2	*3.40*	4.50e−2	*3.87*	3.08e−2	*3.71*
	0.250	1.80e−3	*3.93*	2.50e−3	*4.17*	1.94e−3	*3.99*
	0.125	1.09e−4	*4.05*	1.42e−4	*4.14*	1.34e−4	*3.85*
A rigid translation	1.000	3.82e−1	–	6.17e−1	–	4.73e−1	–
	0.500	3.75e−2	*3.35*	5.01e−2	*3.62*	4.32e−2	*3.45*
	0.250	2.82e−3	*3.73*	4.10e−3	*3.61*	3.57e−3	*3.60*
	0.125	2.01e−4	*3.81*	3.10e−4	*3.73*	2.48e−4	*3.85*
A sinusoidal deformation	1.000	4.88e−1	–	7.85e−1	–	7.03e−1	–
	0.500	6.20e−2	*2.97*	8.43e−2	*3.22*	7.94e−2	*3.15*
	0.250	3.83e−3	*4.02*	5.95e−3	*3.83*	5.03e−3	*3.98*
	0.125	2.59e−4	*3.89*	4.04e−4	*3.88*	3.33e−4	*3.92*

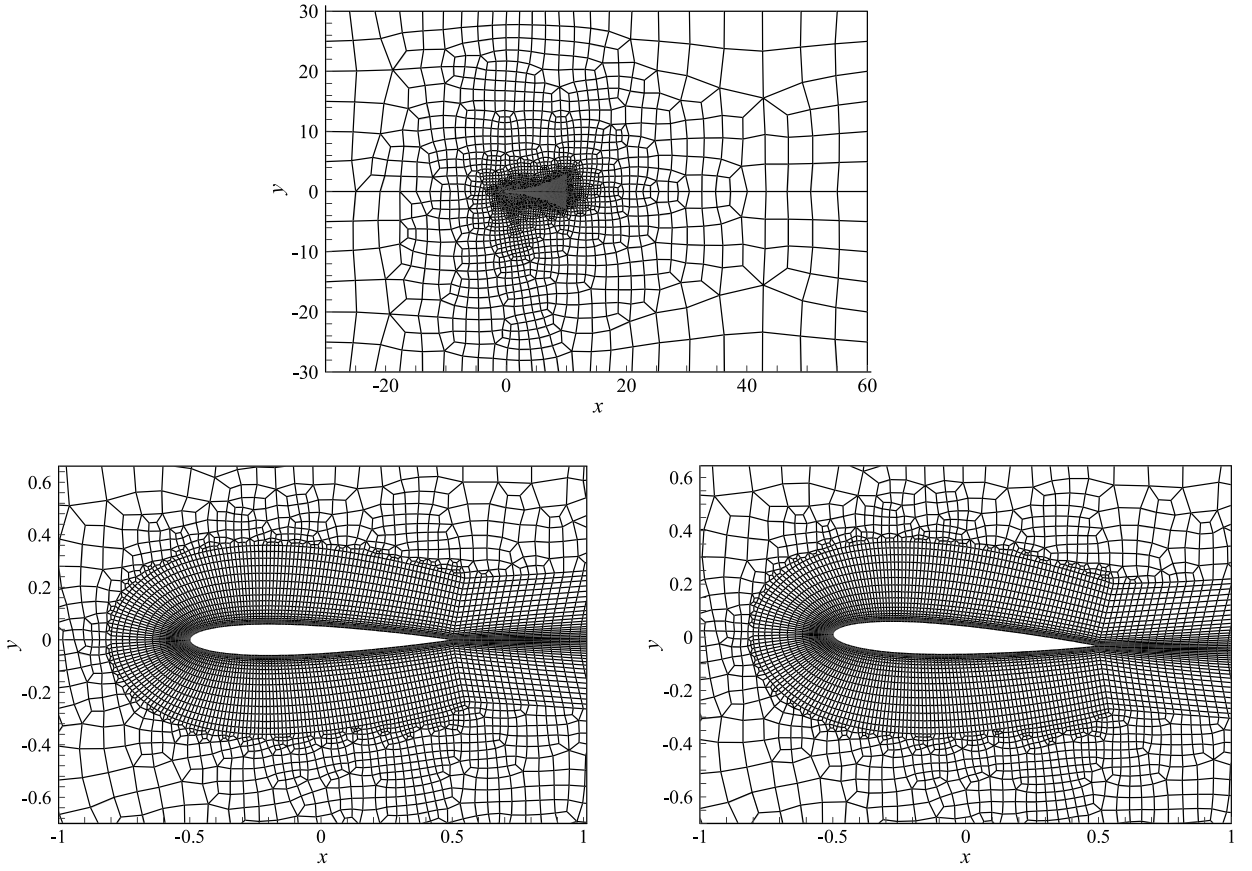

Fig. 4. Initial mesh (*top*) and zoomed-in meshes near the airfoil at two different angles of attack: $\alpha = \alpha_m$ (*bottom-left*) and $\alpha = \alpha_m + \alpha_0$ (*bottom-right*).

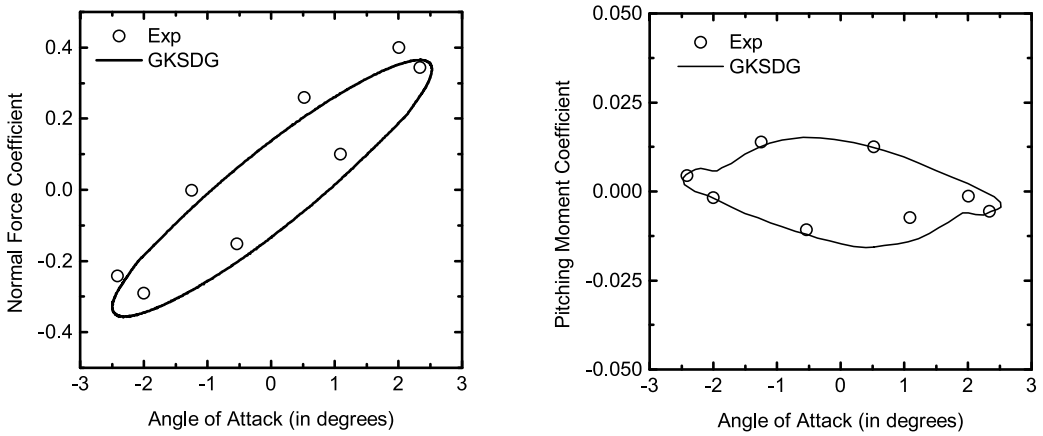

Fig. 5. Normal force coefficients, pitching moment coefficients versus angles of attack.

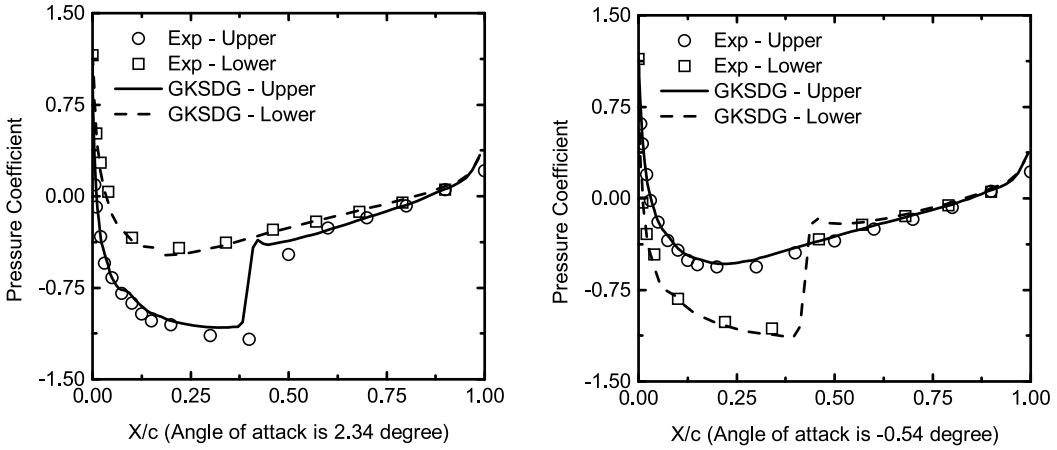

Fig. 6. Pressure coefficient distributions at different angles of attack.

3.2. Flow over an oscillatory pitching NACA0012 airfoil

Pitching around a given pivot point is an elementary motion for a 2D airfoil. In order to show the capability of the method to simulate this kind of problems, a NACA0012 airfoil is used here and the pitching axis is at the quarter chord from the leading edge. The inlet freestream condition has a Mach number $Ma = 0.755$, Reynolds number $Re = 5.5 \times 10^6$, and the specific heat ratio $\gamma = 1.4$. The time-dependent angle of attack is prescribed by $\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t)$ with the mean incidence $\alpha_m = 0.016$ degree, the pitching amplitude $\alpha_0 = 2.51$ degree, and the reduced frequency $\omega c/2U_\infty = 0.0814$. c is the chord length and U_∞ is the freestream velocity, which is parallel to the x axis. The airfoil is placed in x–y plane and the pivot point coordinates are $(x_p, y_p) = (-0.25, 0.0)$. The computational domain is $[-30, 60] \times [-30, 30] \times [0, 1]$ with a symmetry boundary condition applied in the z direction. Fig. 4 shows the initial mesh and two instant zoomed-in meshes near the airfoil at different angles of attack. The cells near the airfoil are moving and deforming with the airfoil while the cells away from the airfoil are kept static. For the periodic motion, the flow field computed at $\alpha = \alpha_m$ is used as an initial condition.

The time-dependent normal force coefficients C_N and pitching moment coefficients C_M, defined in Eq. (55)–(56), are shown in Fig. 5. The numerical solutions are in good agreement with the experimental data reported by Landon [35]. Fig. 6 presents the detailed instantaneous pressure coefficient C_p (Eq. (57)) distributions around the surfaces at different angles of attack. They are also in good agreement with the experimental data. Here we treat the flow problem as an inviscid one which may cause the small difference between the numerical results and the experimental data for

$$C_N = \int_0^1 (C_p^{Lower} - C_p^{Upper}) \mathrm{d}l, \tag{55}$$

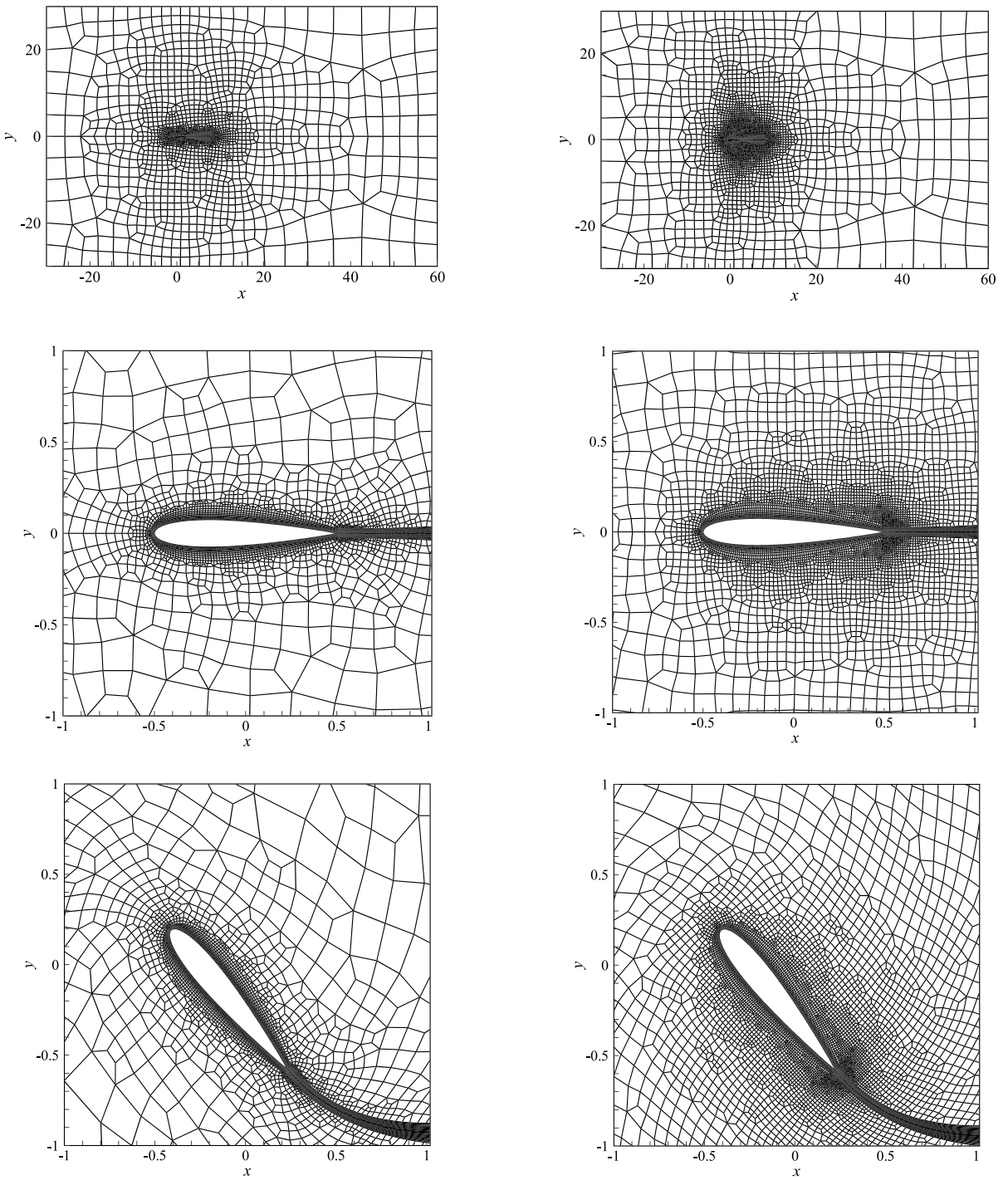

Fig. 7. Meshes for simulation around the pitching airfoil NACA0015. Initial Coarse Mesh (*top-left, middle-left*). Coarse Mesh at 50 degree angle of attack (*bottom-left*). Initial Fine Mesh (*top-right, middle-right*). Fine Mesh at 50 degree angle of attack (*bottom-right*).

$$C_M = \int_0^1 \left(C_p^{Lower} - C_p^{Upper}\right)(0.25 - l)\mathrm{d}l, \tag{56}$$

$$C_p = \frac{(p - p_\infty)}{\frac{1}{2}\rho_\infty U_\infty^2}, \tag{57}$$

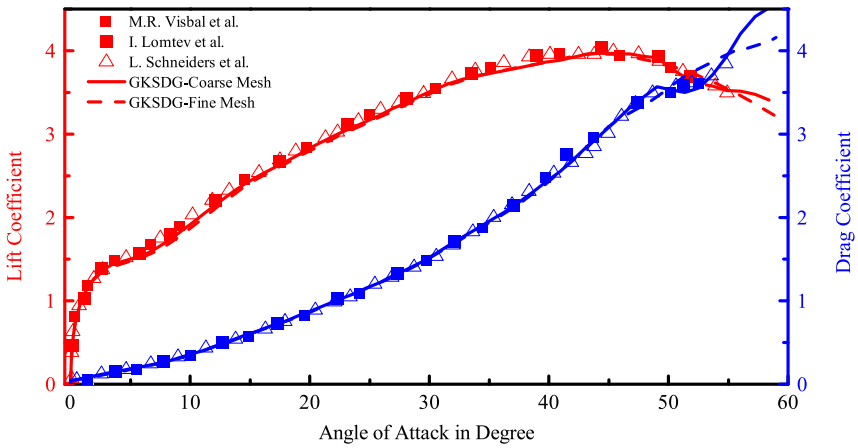

Fig. 8. Lift coefficients and drag coefficients versus angles of attack.

in which l is the distance from the leading edge, $(\rho_\infty, U_\infty, p_\infty)$ are the inlet freestream density, velocity and pressure, and the superscripts "*Lower*" and "*Upper*" denote lower and upper surfaces respectively.

3.3. Laminar flow around a rapidly pitching NACA0015 airfoil

Next, we test proposed method by simulating the laminar flow around a rapidly pitching airfoil. This test case has been used for validation by several authors [11,36,37]. NACA0015 airfoil is pitching upwards about a fixed axis. The pitching rate is $\omega(t) = \omega_0(1 - \exp(-4.6t/t_0))$ rad/s, in which $t_0 = c/U_\infty$ is a referential time when the pitching rate will reach 99% of its final rate, $\omega_0 = 0.6U_\infty/c$, U_∞ is the freestream velocity, and $c = 1$ is the chord length. The fixed axis is located at the quarter chord from the leading edge. The Reynolds number is $Re = 10{,}000$ based on the chord length and freestream parameters, and the inlet Mach number is $Ma = 0.2$ with the specific heat ratio $\gamma = 1.4$. The flow field computed at zero degree angle of attack is used as initial condition.

The computational domain used here is the same as that in Section 3.2. The pivot point coordinates are located at $(x_p, y_p) = (-0.25, 0.0)$. The initial mesh and mesh at 50 degree angle of attack in the x–y plane are shown in Fig. 7. One layer grid is used in the z direction. Here we define two meshes: Coarse Mesh: the number of nodes on the airfoil is 160, the minimum wall-normal spacing on the airfoil is 0.001 (around the leading edge and trailing edge), and the maximum spacing is about 0.003 (around the mid-chord); Fine Mesh: the number of nodes on the airfoil is 200, the minimum spacing on the airfoil is 0.0005 (around the leading edge and trailing edge), and the maximum spacing is about 0.0015 (around the mid-chord). The referential results in [36,37] were obtained using a mesh with the minimum spacing 0.0005. In our calculation, the cells near the airfoil are moving and deforming with the airfoil while cells away from it are kept static.

From the comparison, as shown in Fig. 8, the result with coarse mesh is in excellent agreement with the reference data, and the result with fine mesh has a little different with the reference, especially for the drag coefficient at large angles of attack. Since the fine mesh has better resolution of small scale flow structures at large angles of attack, the results with fine mesh should be more accuracy. Fig. 9 shows the instantaneous normalized vorticities $\omega_z c/U_\infty$ at different angles of attack and we can find that the fine mesh gives more detailed flow structures at large angles of attack.

Furthermore, the results with a piecewise constant grid velocity [24] and a variable mesh velocity along the cell interface proposed in this paper are compared with each other, shown in Fig. 10. It indicates that the mesh velocity treatment affects the results significantly and a variable mesh velocity along the cell interface is necessary for a high order method.

3.4. Flow over a high-frequency plunging SD7003 airfoil at low-Reynolds number

Flows over SD7003 airfoil with different motions have been experimentally studied in a water tunnel by McGowan et al. [38,39] and numerically studied by Visbal et al. [40]. In this section, the ALE method based on gas kinetic theory is used to simulate flow at Reynolds number $Re = 10{,}000$. For this low Reynolds number, transition does not occur over the airfoil at moderate angles of attack prior to stall [40].

The plunging motion is defined as $h(t) = h_0 \sin(2kU_\infty t/c)$ with non-dimensional amplitude $h_0 = 0.05$ and reduced frequency $k = 3.93$. $c = 1$ is the chord length and the freestream velocity U_∞ is parallel to the x axis. The airfoil is placed in the x–y plane with an initial angle of attack $\alpha_0 = 4$ degree. McGowan et al. [39] pointed out the Mach number must be low enough for high-reduced-frequency in order to reduce the compressibility effect. Here we take two Mach numbers $Ma = 0.2$ and 0.05 with the specific heat ratio $\gamma = 5/3$. The computational domain is $[-20, 40] \times [-20, 20] \times [0, 1]$. Fig. 11 presents the initial mesh and two instant zoomed-in meshes near the airfoil. The number of nodes on the airfoil is 200 and the wall-normal spacing is 0.001. During the calculations, the mesh elements near the airfoil are moving and deforming

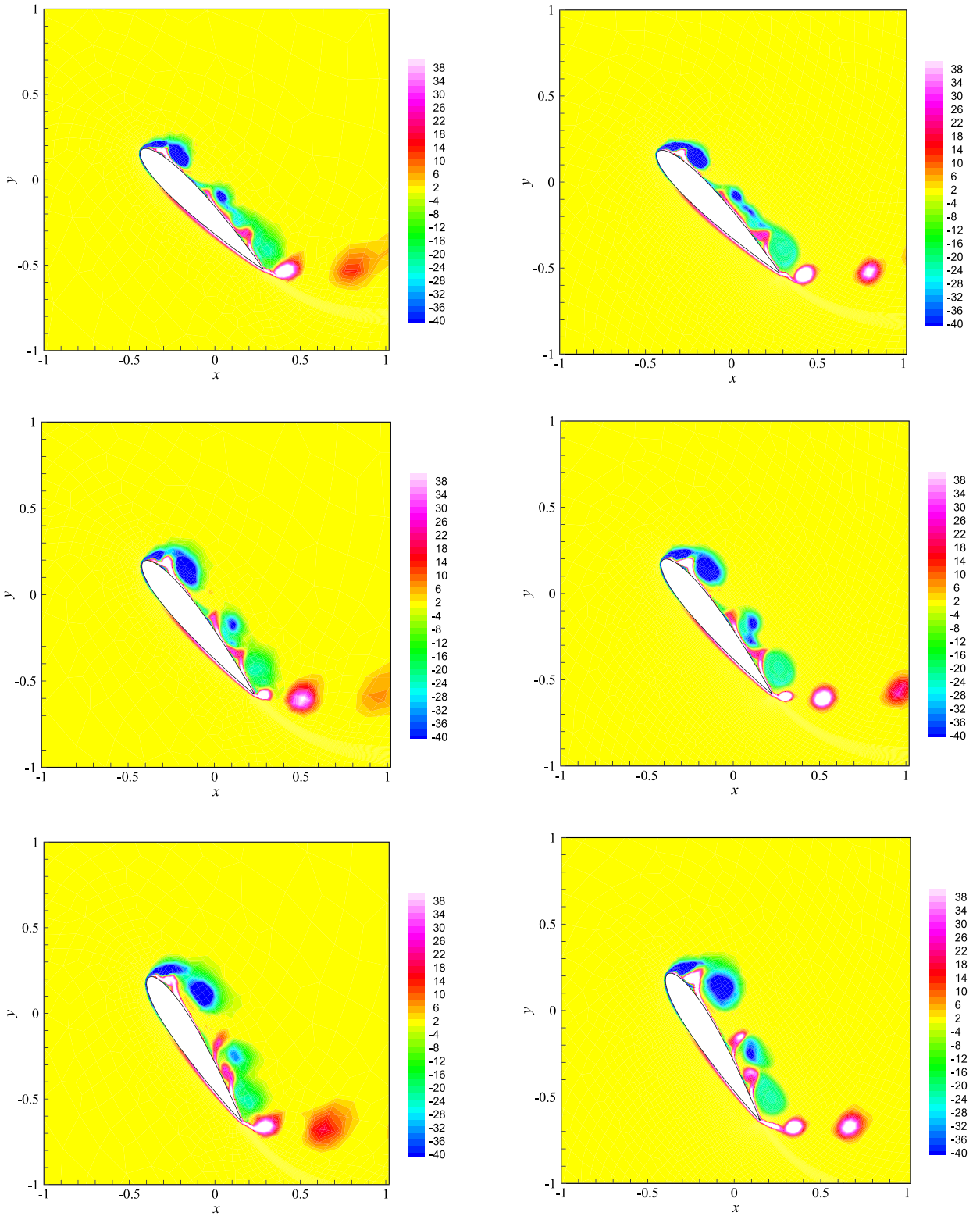

Fig. 9. The instantaneous normalized vorticities $\omega_z c/U_\infty$ at different angles of attack. 45 degree (*top*). 50 degree (*middle*). 57 degree (*bottom*). Coarse Mesh (*left*). Fine Mesh (*right*).

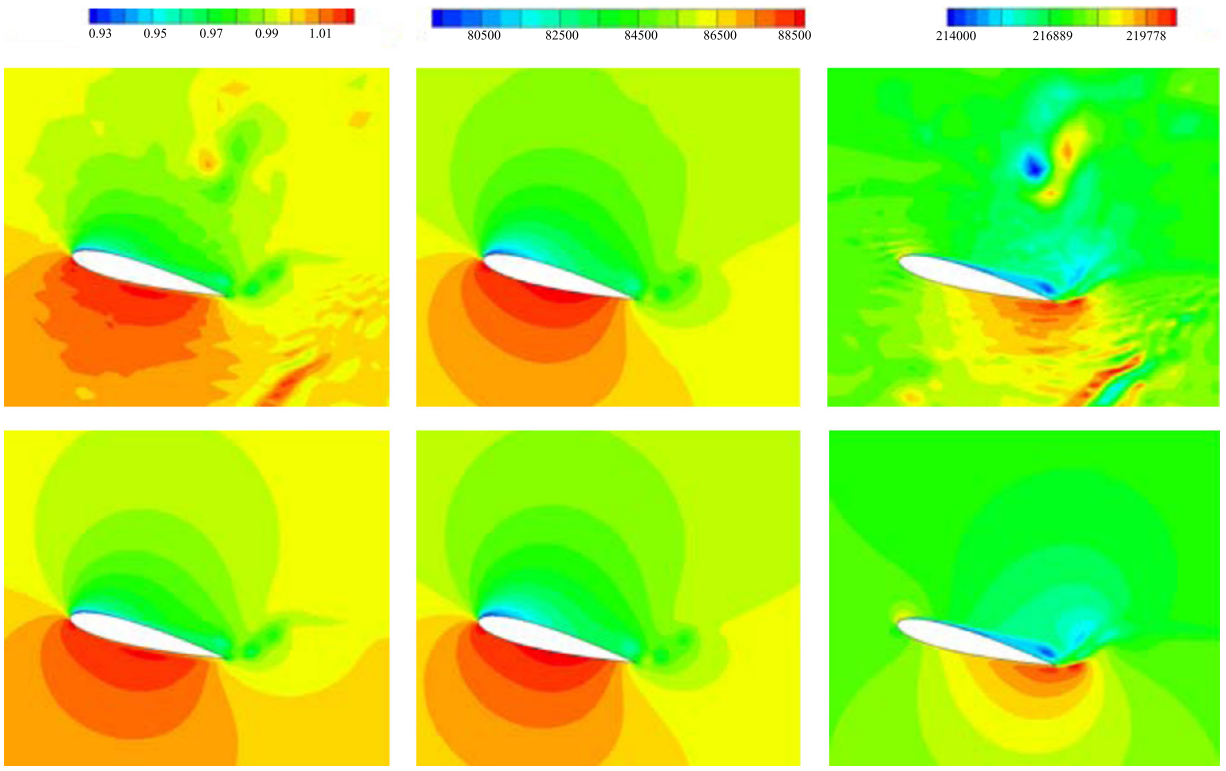

Fig. 10. Comparison of the results with a piecewise constant grid velocity (*up*) and a variable grid velocity along the cell interface (*low*). Density contours (*left*). Pressure (*middle*). Total energy (*right*).

with the airfoil motion while the cells far away are kept static. Visbal et al. [40] used a sixth-order accurate algorithm with a fine mesh (wall-normal resolution 0.00005 and streamwise resolution 0.005). The flow field computed at $h = 0.0$ is used as initial condition.

Fig. 12 shows the time-dependent lift coefficients C_L and drag coefficients C_D. C_L at $Ma = 0.05$ has a better agreement with the experimental data in [39] while the lift with $Ma = 0.2$ appears a phase lag. Besides, the vorticities are also compared with the experiment data [38,39] in Fig. 13, and they are in good agreement with each other. This indicates the proposed method has the capacity to handle this kind of problems and gives reasonable results.

3.5. Hypersonic flow over two relatively moving cylinders

In this section, a hypersonic flow over two relatively moving cylinders is simulated to validate that the method for hypersonic flows. Two cylinders are moving forward with a Mach number 8.03 based the sound speed of upstream flow condition, while the rear one is periodically oscillating around the front cylinder center, as shown in Fig. 14. The cylinder has a radius 1 m and the distance between the two cylinder centers is 4 m. The far flow filed condition is $T_\infty = 124.94$ K with the specific heat ratio $\gamma = 1.4$ and the specific gas constant $R = 287.096$ J/(kg K). The Reynolds number is $Re = 1.835 \times 10^5$ with the cylinder moving velocity, cylinder radius, and the far field flow parameters. The cylinder has a constant temperature $T_w = 294.44$ K. The trajectory of the center of the front cylinder is

$$\begin{cases} x_1(t) = x_{10} - 1799.5t, \\ y_1(t) = y_{10}, \end{cases} \tag{58}$$

in which (x_{10}, y_{10}) is the initial coordinate for the front cylinder center and $(x_1(t), y_1(t))$ is its new coordinate for each time step. The center of the rear cylinder has the following motion

$$\begin{cases} x_2(t) = (x_{20} - x_{10})\cos(4000t) - (y_{20} - y_{10})\sin(4000t) + x_{10} - 1799.5t, \\ y_2(t) = (x_{20} - x_{10})\sin(4000t) - (y_{20} - y_{10})\cos(4000t) + y_{10}, \end{cases} \tag{59}$$

where (x_{20}, y_{20}) is the initial coordinate for the rear cylinder center and $(x_2(t), y_2(t))$ is its new coordinates for each time step. Fig. 15 shows the comparison between the numerical results and the experimental data [41] and Fig. 16 shows the Mach number contours at different time. They are in good agreement and it indicates that the method proposed in this paper has the capacity to simulate the hypersonic flow with multiple moving objects.

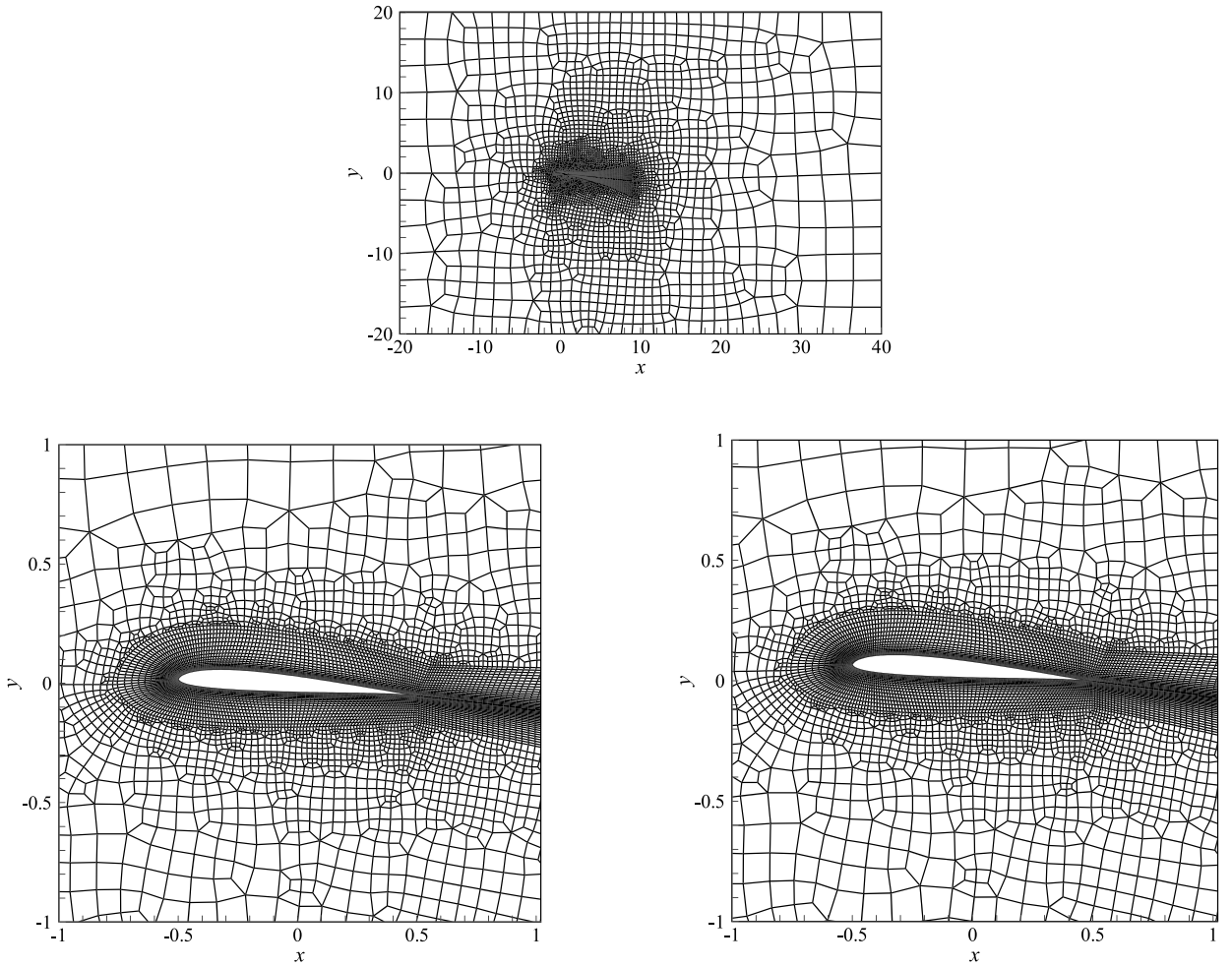

Fig. 11. Initial mesh (*top*) and two instant zoomed-in meshes near the airfoil: $h = 0.0$ (*bottom-left*) and $h = h_0$ (*bottom-right*).

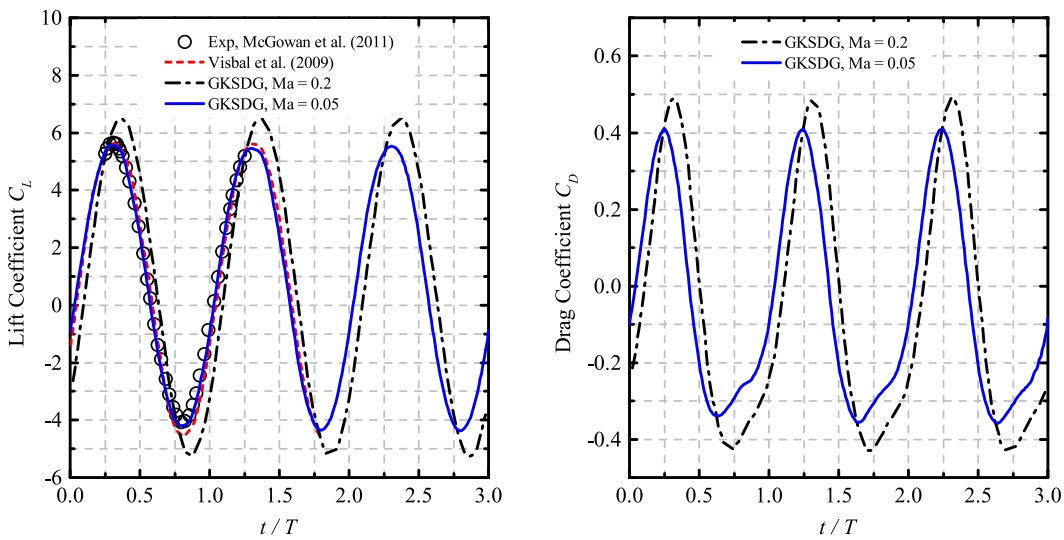

Fig. 12. Lift coefficients (*left*) and drag coefficients (*right*) versus normalized time t/T with T is the time period.

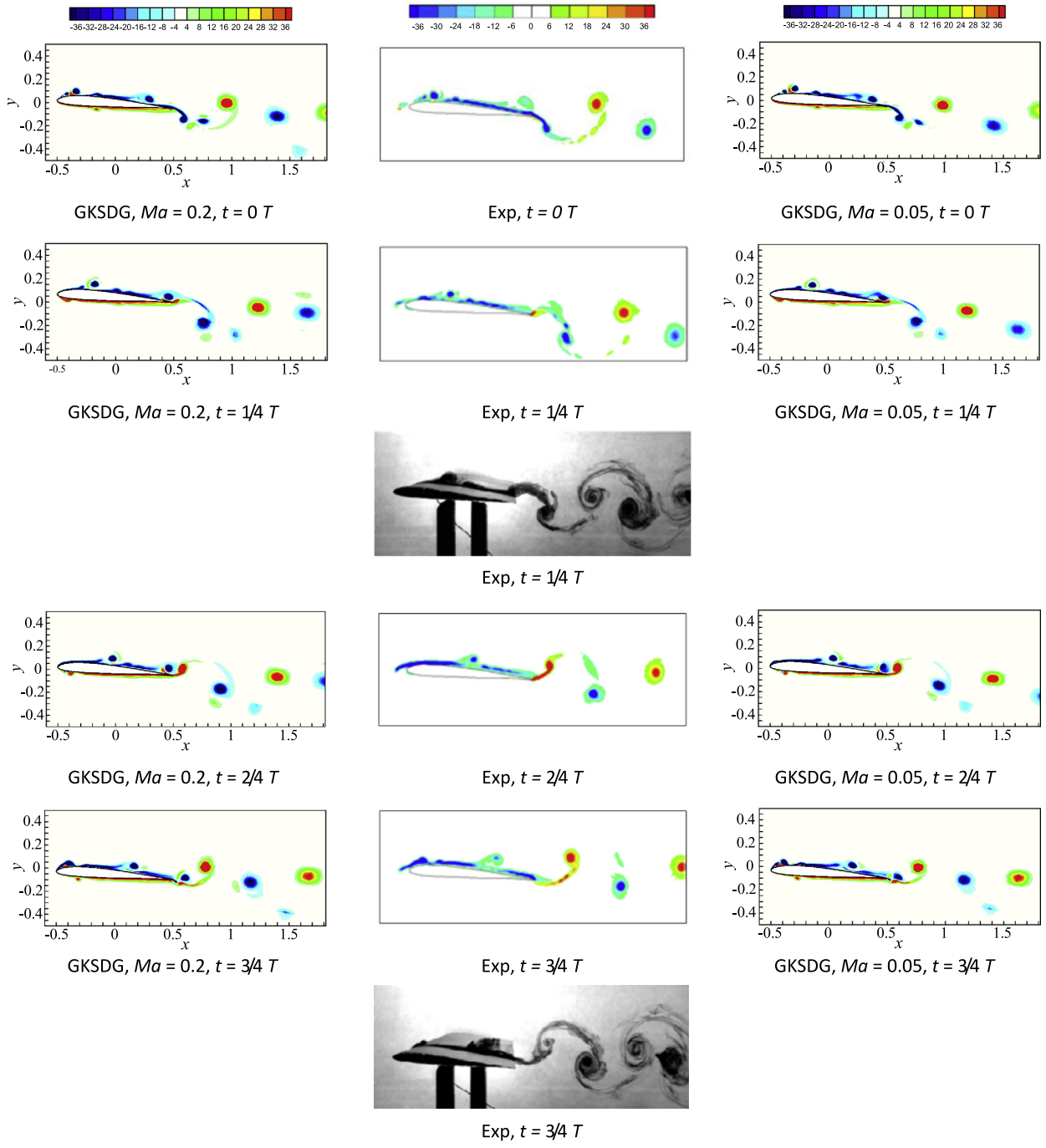

Fig. 13. Comparison of the vorticities $\omega_z c/U_\infty$. T is the time period.

4. Conclusions

A multi-dimensional high order DG-ALE method based on gas kinetic theory is introduced in this paper. It extends the third-order gas kinetic method from a static domain to the flow simulation over variable domain. A multi-dimensional gas kinetic model with an arbitrary mesh velocity distribution is proposed for the inviscid and viscous flow computations. Due to the spatial and temporal dependence of the gas distribution functions, the flux can be integrated within a cell and along a cell interface. At the same time, the flux can be integrated in time analytically without using any Runge–Kutta time-stepping method. The grid velocities inside a cell and along a cell interface can be arbitrarily chosen, such as interface translation and rotation, and the geometric conservation law can be satisfied accurately. The satisfaction of GCL has been proved

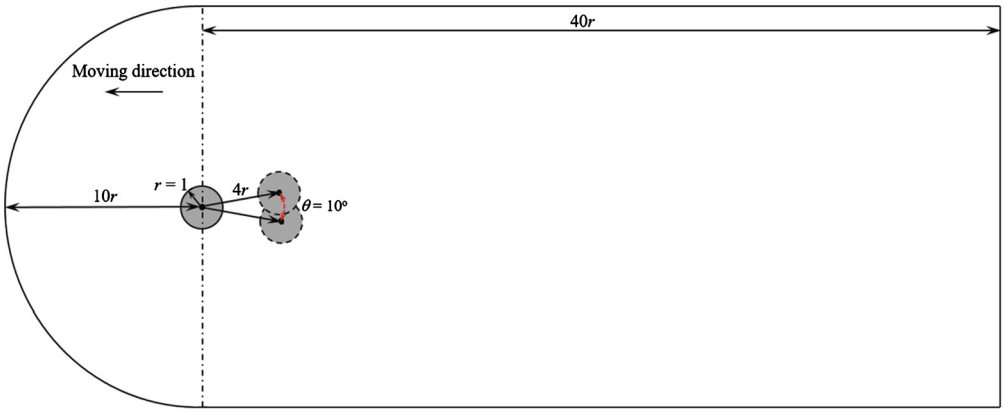

Fig. 14. Computational domain.

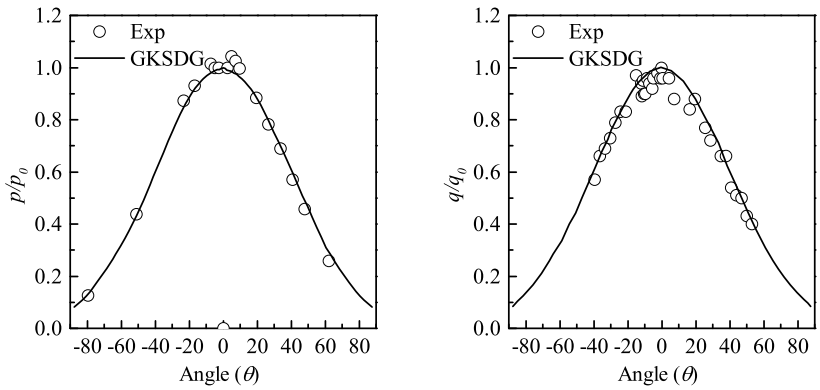

Fig. 15. Comparison of the numerical results with the experimental data. Pressure (*left*) and heat flux (*middle*) along the front cylindrical surface.

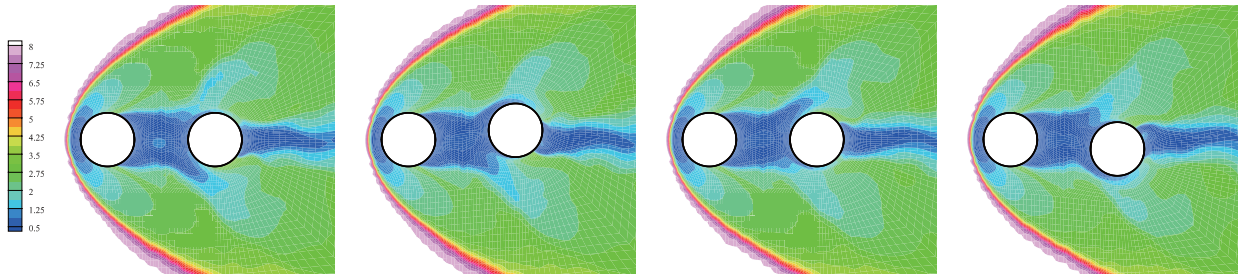

Fig. 16. Mach number contours at different time around the two moving cylinders.

mathematically and numerically. The numerical accuracy of the scheme is still maintained for a moving and deforming mesh. Several problems, i.e., flow over an oscillatory pitching NACA0012 airfoil, around a rapidly pitching NACA0015 airfoil, over a high-frequency plunging SD7003 airfoil, and two moving and oscillating cylinders, are computed to validate the numerical method. The current ALE-DG gas-kinetic scheme can be faithfully used for the study of oscillating bodies at modest flow speed.

Acknowledgements

The research was supported by Hong Kong Research Grants Council, University Grants Committee (620813, 16211014, 16207715) and HKUST (PROVOST13SC01, IRS15SC29, SBI14SC11).

References

[1] R. Kamakoti, W. Shyy, Fluid–structure interaction for aeroelastic applications, Prog. Aerosp. Sci. 40 (2004) 535–558.

[2] D.M. Schuster, D.D. Liu, L.J. Huttsell, Computational aeroelasticity: success, progress, challenge, J. Aircr. 40 (2003) 843–856.
[3] R.E. Bartels, A. Sayma, Computational aeroelastic modelling of airframes and turbomachinery: progress and challenges, Philos. Trans. R. Soc., Math. Phys. Eng. Sci. 365 (2007) 2469–2499.
[4] J.G. Marshall, M. Imregun, A review of aeroelasticity methods with emphasis on turbomachinery applications, J. Fluids Struct. 10 (1996) 237–267.
[5] W. Shyy, H. Aono, S.K. Chimakurthi, P. Trizila, C.-K. Kang, C.E. Cesnik, H. Liu, Recent progress in flapping wing aerodynamics and aeroelasticity, Prog. Aerosp. Sci. 46 (2010) 284–327.
[6] X.-s. Li, C.-w. Gu, Mechanism of Roe-type schemes for all-speed flows and its application, Comput. Fluids 86 (2013) 56–70.
[7] X.-s. Li, Uniform algorithm for all-speed shock-capturing schemes, Int. J. Comput. Fluid Dyn. 28 (2014) 329–338.
[8] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V, J. Comput. Phys. 141 (1998) 199–224.
[9] B. Cockburn, C.W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[10] J. Donea, A. Huerta, J.P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian–Eulerian methods, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds.), Encyclopedia of Computational Mechanics, John Wiley & Sons, Ltd., 2004.
[11] I. Lomtev, R.M. Kirby, G.E. Karniadakis, A discontinuous Galerkin ALE method for compressible viscous flows in moving domains, J. Comput. Phys. 155 (1999) 128–159.
[12] M. Prasad, J. Milovich, A. Shestakov, D. Kershaw, M. Shaw, 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous Galerkin method, in: B. Cockburn, G. Karniadakis, C.-W. Shu (Eds.), Discontinuous Galerkin Methods, Springer, Berlin, Heidelberg, 2000, pp. 397–405.
[13] J.J.W. Van der Vegt, H. Van der Ven, Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. General formulation, J. Comput. Phys. 182 (2002) 546–585.
[14] H. Van der Ven, J.J.W. Van der Vegt, Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows: II. Efficient flux quadrature, Comput. Methods Appl. Mech. Eng. 191 (2002) 4747–4780.
[15] P.O. Persson, J. Bonet, J. Peraire, Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains, Comput. Methods Appl. Mech. Eng. 198 (2009) 1585–1595.
[16] V.-T. Nguyen, An arbitrary Lagrangian–Eulerian discontinuous Galerkin method for simulations of flows over variable geometries, J. Fluids Struct. 26 (2010) 312–329.
[17] C.M. Klaij, J.J.W. van der Vegt, H. van der Ven, Space–time discontinuous Galerkin method for the compressible Navier–Stokes equations, J. Comput. Phys. 217 (2006) 589–611.
[18] J.J.W. van der Vegt, Y. Xu, Space–time discontinuous Galerkin method for nonlinear water waves, J. Comput. Phys. 224 (2007) 17–39.
[19] E. Ferrer, R.H.J. Willden, A high order Discontinuous Galerkin – Fourier incompressible 3D Navier–Stokes solver with rotating sliding meshes, J. Comput. Phys. 231 (2012) 7037–7056.
[20] S. Rhebergen, B. Cockburn, A space–time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains, J. Comput. Phys. 231 (2012) 4185–4204.
[21] K. Xu, A Gas-Kinetic BGK scheme for the Navier–Stokes equations and its connection with artificial dissipation and Godunov method, J. Comput. Phys. 171 (2001) 289–335.
[22] X. Ren, K. Xu, W. Shyy, C. Gu, A multi-dimensional high-order discontinuous Galerkin method based on gas kinetic theory for viscous flow computations, J. Comput. Phys. 292 (2015) 176–193.
[23] C. Jin, K. Xu, A unified moving grid gas-kinetic method in Eulerian space for viscous flow computation, J. Comput. Phys. 222 (2007) 155–175.
[24] G. Ni, S. Jiang, K. Xu, Remapping-free ALE-type kinetic method for flow computations, J. Comput. Phys. 228 (2009) 3154–3171.
[25] M.A. Potsdam, G.P. Guruswamy, A parallel multiblock mesh movement scheme for complex aeroelastic applications, AIAA Paper 716, 2001.
[26] X.-W. Gao, P.-C. Chen, L. Tang, Deforming mesh for computational aeroelasticity using a nonlinear elastic boundary element method, AIAA J. 40 (2002) 1512–1517.
[27] C.O.E. Burg, A robust unstructured grid movement strategy using three-dimensional torsional springs, AIAA Paper, No. 2004-2529, 2004.
[28] T.C.S. Rendall, C.B. Allen, Unified fluid–structure interpolation and mesh motion using radial basis functions, Int. J. Numer. Methods Eng. 74 (2008) 1519–1559.
[29] P. Thomas, C. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA J. 17 (1979) 1030–1037.
[30] W. Shyy, H. Udaykumar, M.M. Rao, R.W. Smith, Computational Fluid Dynamics with Moving Boundaries, Courier Corporation, 2012.
[31] G. Erlebacher, M.Y. Hussaini, C.-W. Shu, Interaction of a shock with a longitudinal vortex, J. Fluid Mech. 337 (1997) 129–153.
[32] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: A. Quarteroni (Ed.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer-Verlag, Berlin/New York, 1998, pp. 325–432.
[33] H.C. Yee, N.D. Sandham, M.J. Djomehri, Low-dissipative high-order shock-capturing methods using characteristic-based filters, J. Comput. Phys. 150 (1999) 199–238.
[34] J. Luo, K. Xu, A high-order multidimensional gas-kinetic scheme for hydrodynamic equations, Sci. China, Technol. Sci. 56 (2013) 2370–2384.
[35] R. Landon, NACA 0012 oscillatory and transient pitching, AGARD Report 702, AGARD, 1982.
[36] M.R. Visbal, J.S. Shang, Investigation of the flow structure around a rapidly pitching airfoil, AIAA J. 27 (1989) 1044–1051.
[37] L. Schneiders, D. Hartmann, M. Meinke, W. Schröder, An accurate moving boundary formulation in cut-cell methods, J. Comput. Phys. 235 (2013) 786–809.
[38] G.Z. McGowan, A. Gopalarathnam, M.V. Ol, J.R. Edwards, F. Daniel, Computation vs. experiment for high-frequency low-Reynolds number airfoil pitch and plunge, in: 46th AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 2008.
[39] G.Z. McGowan, K. Granlund, M.V. Ol, A. Gopalarathnam, J.R. Edwards, Investigations of lift-based pitch-plunge equivalence for airfoils at low Reynolds numbers, AIAA J. 49 (2011) 1511–1524.
[40] M.R. Visbal, R.E. Gordnier, M.C. Galbraith, High-fidelity simulations of moving and flexible airfoils at low Reynolds numbers, Exp. Fluids 46 (2009) 903–922.
[41] A.R. Wieting, Experimental study of shock wave interface heating on a cylindrical leading edge, 1987.