ELSEVIER

Journal of Fluids and Structures 26 (2010) 312–329

JOURNAL OF
FLUIDS AND
STRUCTURES

www.elsevier.com/locate/jfs

An arbitrary Lagrangian–Eulerian discontinuous Galerkin method for simulations of flows over variable geometries

Vinh-Tan Nguyen*

Institute of High Performance Computing, 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632, Singapore

Received 11 September 2008; accepted 13 November 2009
Available online 13 January 2010

Abstract

A numerical method for simulations of flow over variable geometries including deformable domains or moving boundaries is presented in the context of high-order discontinuous Galerkin (DG) methods in the framework of an arbitrary Lagrangian Eulerian (ALE) description to take into account the deformable domains where boundaries are moving with prescribed motions or deformed under external forces. The ALE DG approach is shown to satisfy the geometric conservation law while preserving high-order accuracy. For variable geometries, we develop a simple and effective mesh smoothing technique based on element size functions to handle deformable grids due to moving boundaries. The current approach is applied to simulations of moving domain problems including laminar flows over oscillating cylinders and a flapping foil to show the ability of handling variable geometries with large deformation and high accuracy. The simulation results are compared with earlier experimental and numerical studies.

Keywords: Arbitrary Lagrangian–Eulerian; Discontinuous Galerkin; Deformable bodies; Fluid–structure interaction; Compressible Navier–Stokes flows

1. Introduction

Flows over variable-geometry domains are largely involved in many engineering applications, including aeroelasticity, fluid–structure interactions, flapping and insect flight. Together with experimental studies, numerical simulations have played an important role in investigating and understanding these complex and multi-disciplinary phenomena. Numerically, one requires the simulation methods to be able to handle moving domains while maintaining accuracy and conservation. Moreover, the interactions of flows with moving boundaries resulting in unsteady phenomena, such as wakes and vortices, necessitate high-order accurate approximations to resolve and capture the unsteadiness of flows. These requirements become one of the main challenges in developing a robust and efficient numerical method for this class of problems.

The arbitrary Lagrangian–Eulerian (ALE) description is most commonly used in flow problems involving large deformation due to moving boundaries. It was primarily developed for finite difference methods for fluid flows (Hirt et al., 1997) and later extended in the context of finite element and finite volume methods for both fluid and structure dynamics [see Donea and Huerta (2003) for an extensive review]. Combining the best features from the traditional

*Tel.: +65 6419 1591; fax: +65 6467 4350.
E-mail address: nguyenvt@ihpc.a-star.edu.sg

0889-9746/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jfluidstructs.2009.11.002

Lagrangian and Eulerian descriptions, the ALE method uses a computational mesh which can move with arbitrary velocity independent of the material velocity. It thus gives flexibility in handling moving domain problems, especially for fluid–structure interactions. On one hand, it allows flexible grids with material flowing through. On the other hand, it avoids problems of tracking interfaces in the Eulerian framework as well as large distortion encountered in the Lagrangian approach.

Although the grid is allowed to move arbitrarily, the velocity at domain boundaries is usually known in practice, e.g. moving boundary problems and fluid–structure interactions. The computational grid thus may be distorted with just the movement of grid points on the boundaries. One of the main issues in the ALE description is to decide when to regenerate the mesh in order to preserve its quality. However, regeneration of the grids usually requires more computational resources and thus should be avoided if possible. Alternatively, a grid velocity smoothing scheme is employed to update the grid and minimize the distortion. In principle, the grid velocity is constructed for the whole domain such that the grid quality is assured and the velocity at the boundaries matches the given boundary velocity. One may refer to Lohner and Yang (1996) for a brief description on various techniques used in literature.

Apart from handling moving grids in those deformable domain applications, accuracy is also another concern in designing a numerical scheme for this class of problems. Moreover, with the advent of increasing computational power, the requirement of high-order accuracy in numerical simulations has become more critical in obtaining solutions for large-scale complex problems such as aeroacoustics, turbulence flows, etc. Development of high-order methods and their application for solving real-world problems have emerged as one of the most challenging research topics in computational mechanics over the last decade. In recent years, discontinuous Galerkin (DG) methods have made considerable progress in computational mechanics. The concept of combining finite element and finite volume discretization makes DG methods suitable for a wide range of applications in many areas including fluid dynamics, aeroacoustics and electromagnetics (Cockburn and Shu, 2001), mainly due to its advantages of high-order approximation on arbitrary unstructured grids as well as compactness and conservation. In this article, high-order DG discretizations are combined with the ALE approach to provide a new way of simulating compressible Navier–Stokes flows over deformable domains and moving boundaries.

For variable geometry problems, a number of DG methods in the ALE framework have been developed for many applications of moving boundaries, free surface flows, fluid–structure interactions; see, for instance, Lomtev et al. (1999) and van de Vegt and Xu (2007). An ALE matrix-free discontinuous Galerkin method was first introduced in Lomtev et al. (1999) for compressible Navier–Stokes flows in moving domains. There, the DG discretization is applied for the conservation laws written for arbitrarily moving domains in the ALE framework as described in Ventakasubban (1995) with an additional source term involving the grid moving velocity. It uses a force-directed algorithm from graph theory to update the grid moving velocity. An alternative approach is presented in Persson et al. (2009) for high-order DG methods on deformable domains. In that work, a continuous mapping between a fixed reference configuration and a time-varying domain is constructed with explicit mapping using blending functions. The equations of conservation laws are then cast in the reference domain with the introduction of variable geometries. An additional equation related to the Jacobian of transformations is introduced to ensure the preservation of constant solutions. The method has been shown to obtain high-order accurate solutions for many applications with variable geometries. However, the construction of explicit mapping is not always trivial for some complex applications.

In this work, we introduce a DG approach for compressible Navier–Stokes flows in variable-geometry domains. The governing equation is discretized in the moving frame using high-order discontinuous Galerkin spatial discretization. In our presented discretization, the divergence of grid velocity is eliminated and the geometric conservation law is always satisfied without introducing a grid velocity source term, as suggested in Lomtev et al. (1999). This implies the preservation of constant solutions as well as the accuracy and stability in solving the moving domain problems. The explicit TVD Runge–Kutta time stepping (Gottlieb and Shu, 1998) is applied for time integration. As for the mesh movement, the grid velocity is updated in such a way that the nodes very close to the moving boundaries have a velocity almost equivalent to that of the boundaries, as suggested in Lohner and Yang (1996). However, instead of solving for the Laplacian equation with variable diffusion, the node moving velocity is constructed locally from its neighbouring vertices as a modification of the force-directed algorithm presented in Lomtev et al. (1999) with the weights evaluated from element size functions. A combination of variable diffusivity approach and spring analogy results in a simple and effective algorithm in handling deformable meshes with minimum overhead. Although the proposed approach is developed for general three-dimensional applications, for the sake of simplicity in presentation, it is applied here to problems in two-dimensional space.

For numerical simulations, we first present some of the results on accuracy as well as the preservation of constant solutions to assure the high-order accuracy and conservation of the approach. As an application of the proposed approach for moving domain problems, we consider some unsteady flow problems including flows over moving boundaries with prescribed motions. The most common example is that of a circular cylinder oscillating in direction transverse to the free stream. The structure of the flow, as well as forces acting on the cylinder, depends on Reynolds number and frequency ratio. When the oscillating frequency of the cylinder matches the natural Karman frequency of the stationary cylinder wake, a lock-on phenomena is occurred. This problem has been extensively investigated in the literature experimentally (Tanida et al., 1973; Gu et al., 1994) and numerically (Lu and Dalton, 1996; Patnaik et al., 1999). The results from our simulations for the problems of an oscillating cylinder and a pitching airfoil are presented and are found to compare well with the earlier published results. Examples of a pitching and hovering airfoil, similar to an insect flight mechanism (Wang, 2005), are also presented in this work. The results compare well with earlier experiments and numerical studies.

2. ALE discontinuous Galerkin for variable geometries

2.1. Governing equations

Consider the system of conservation laws

$$\boldsymbol{u}_t + \nabla \cdot \boldsymbol{F}(\boldsymbol{u}) = 0, \tag{1}$$

over the domain Ω with the appropriate boundary conditions applied on the domain boundary $\Gamma \equiv \partial\Omega = \Gamma_D \cup \Gamma_N$. In general, the boundaries are allowed to move with velocity $\boldsymbol{v}_\Gamma$. Here, $\boldsymbol{u}(\boldsymbol{x}, t)$ is the conserved variable and $\boldsymbol{x} = (x_1, \ldots, x_d)$ is the position vector in d-dimensional space. In the present work, we consider a compressible Navier–Stokes flow, where the conserved variables are given by

$$\boldsymbol{u} = (\rho \;\; \rho u_1 \;\; \rho u_2 \;\; \rho u_3 \;\; \rho e)^{\mathrm{T}}. \tag{2}$$

The flux function $\boldsymbol{F} = \boldsymbol{F}_e - \boldsymbol{F}_v$ includes the Euler flux $\boldsymbol{F}_e(\boldsymbol{u})$ and the viscous flux $\boldsymbol{F}_v(\boldsymbol{u})$ described in three-dimensional space as

$$\boldsymbol{F}_e^i(\boldsymbol{u}) = \begin{pmatrix} \rho u_i \\ \rho u_i u_1 + p\delta_{i1} \\ \rho u_i u_2 + p\delta_{i2} \\ \rho u_i u_3 + p\delta_{i3} \\ \rho h u_i \end{pmatrix} \quad \text{for } i = 1, 2, 3, \tag{3}$$

$$\boldsymbol{F}_v^i(\boldsymbol{u}, \nabla\boldsymbol{u}) = \begin{pmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ u_k\tau_{ki} - Q_i \end{pmatrix} \quad \text{for } i = 1, 2, 3, \tag{4}$$

where ρ is the density, u_i is the velocity component, p is the pressure, e is the internal energy, h is the enthalpy and $\delta_{i,j}$ is the Kronecker function. The stresses are given by

$$\tau_{ij} = \mu\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \lambda\delta_{ij}\frac{\partial u_k}{\partial x_k}\right] \quad \text{for } i,j = 1, 2, 3 \tag{5}$$

and the heat flux is

$$Q_i = -\kappa\frac{\partial T}{\partial x_i} \quad \text{for } i = 1, 2, 3. \tag{6}$$

Here, μ is the dynamic viscosity coefficient, $\lambda = -\frac{2}{3}$, κ is the coefficient of thermal conductivity and T is the temperature. The Prandtl number is computed as

$$\mathrm{Pr} = \frac{c_p\mu}{\kappa}. \tag{7}$$

2.2. ALE formulation

Let $\mathcal{V}_h^p(\Omega)$ be the space of polynomials of degree p on the subdivision $\mathcal{T}_h$ of the domain Ω into elements $\Omega = \bigcup_{\kappa \in \mathcal{T}_h} \kappa$. Movement of the domain boundaries requires the change of grid in order to maintain the conforming, connectivity and quality of the elements. Denote $\varphi_i(\boldsymbol{x}, t) \in \mathcal{P}^p$ to be the basis of $\mathcal{V}_h^p$, and write the approximate solution $\boldsymbol{u}_h$ in terms of the discrete solution vector $\overline{\boldsymbol{u}}_h$ as

$$\boldsymbol{u}_h(\boldsymbol{x}, t) = \sum_{j=0}^{N(p)} \overline{\boldsymbol{u}}_{h,j}(t)\varphi_j(\boldsymbol{x}, t), \tag{8}$$

where $N(p)$ is the number of degrees of freedom and $\varphi_i(\boldsymbol{x}, t)$ is taken as nodal basis functions (Koornwinder, 1992) dependent on both space and time to take into account mesh motions. For the sake of simplicity, we will drop the arguments $(\boldsymbol{x}, t)$ in the presentation of the basis function φ for the rest of the paper. For a cell with volume κ bounded by the surface $\partial\kappa$, which may be moving in time with velocity $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x}, t)$, where $\boldsymbol{x} \in \partial\kappa$, the DG discretization is written in the form: find $\boldsymbol{u}_h \in \mathcal{V}_h^p(\Omega)$ such that

$$\int_\kappa \varphi_i \frac{\partial \boldsymbol{u}_h}{\partial t}\,\mathrm{d}\boldsymbol{x} + \int_\kappa \varphi_i \nabla \cdot \boldsymbol{F}(\boldsymbol{u}_h)\,\mathrm{d}\boldsymbol{x} = 0, \quad \forall \kappa \in \mathcal{T}_h,\ \forall \varphi_i \in \mathcal{V}_h^p,\ \forall i = 1, \ldots, N(p). \tag{9}$$

It is noted that the state values $\boldsymbol{u}_h(\boldsymbol{x}, t)$ and the basis $\varphi_i(\boldsymbol{x}, t)$ are changing with time; thus, the Reynolds transport theorem can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\kappa(t)} \varphi_i \boldsymbol{u}_h\,\mathrm{d}\boldsymbol{x} = \int_{\kappa(t)} \varphi_i \frac{\partial \boldsymbol{u}_h}{\partial t}\,\mathrm{d}\boldsymbol{x} + \int_{\kappa(t)} \frac{\partial \varphi_i}{\partial t}\boldsymbol{u}_h\,\mathrm{d}\boldsymbol{x} + \oint_{\partial\kappa(t)} \varphi_i \boldsymbol{u}_h v_n\,\mathrm{d}s, \quad \forall \varphi_j \in \mathcal{V}_h^p,\ \forall \kappa \in \mathcal{T}_h(t),\ \forall i = 1, \ldots, N(p), \tag{10}$$

where $v_n = \boldsymbol{v} \cdot \boldsymbol{n}$ is the moving velocity of the cell interface in normal direction. Substitute (10) into the Eq. (9) and, via integration by parts, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\kappa(t)} \varphi_i \boldsymbol{u}_h\,\mathrm{d}\boldsymbol{x} = \int_{\kappa(t)} \frac{\partial \varphi_i}{\partial t}\boldsymbol{u}_h\,\mathrm{d}\boldsymbol{x} + \int_{\kappa(t)} \nabla\varphi_i \cdot \boldsymbol{F}(\boldsymbol{u}_h)\,\mathrm{d}\boldsymbol{x} - \oint_{\partial\kappa(t)} \varphi_i(\mathcal{F}_n(\boldsymbol{u}_h) - \boldsymbol{u}_h v_n)\,\mathrm{d}s, \quad \forall \varphi_i \in \mathcal{V}_h^p,\ \forall \kappa \in \mathcal{T}_h(t),\ \forall i = 1, \ldots, N(p). \tag{11}$$

The discontinuous Galerkin formulation for a moving grid can now be expressed as follows: find $\boldsymbol{u}_h \in \mathcal{V}_h^p$ such that $\forall \varphi_i \in \mathcal{V}_h^p$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\kappa \varphi_i \boldsymbol{u}_h\,\mathrm{d}\boldsymbol{x} - \int_\kappa \frac{\partial \varphi_i}{\partial t}\boldsymbol{u}_h\,\mathrm{d}\boldsymbol{x} - \int_\kappa \nabla\varphi_i \cdot \boldsymbol{F}(\boldsymbol{u}_h)\,\mathrm{d}\boldsymbol{x} + \oint_{\partial\kappa} \varphi_i \mathcal{F}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}, v_n)\,\mathrm{d}s = 0, \quad \forall \kappa \in \mathcal{T}_h,\ \forall i = 1, \ldots, N(p), \tag{12}$$

where $\mathcal{F}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}, v_n)$ approximating $\boldsymbol{F}_n(\boldsymbol{u}) - \boldsymbol{u}v_n$ is the numerical flux at an interior element boundary or a domain boundary which is moving with velocity v_n in the normal direction. The $()^+$ and $()^-$ notation indicates the trace of the solution taken from the interior and exterior of the element, respectively, and $\boldsymbol{n}$ is the outward normal vector to the boundary of the element. Along the domain boundaries, the exterior state of the solution is constructed by weakly imposing the boundary conditions. In the computation of the flux across element boundaries, various numerical flux functions including Godunov's, Lax-Friedrichs', Roe's can be used for evaluating of inviscid flux; for example, we use Lax-Friedrichs flux in this work. As for the viscous flux requiring the approximation of solution derivatives, we use a local discontinuous Galerkin procedure (Cockburn and Shu, 2001). At the domain boundaries, boundary conditions are weakly imposed by evaluating the interior and exterior solution states. Details of how to implement the boundary conditions are presented in the Appendix A.

The fundamental ALE relation between material time derivatives, referential time derivatives and spatial derivatives of basis functions is written as

$$\frac{\mathrm{d}\varphi_i}{\mathrm{d}t} = \frac{\partial \varphi_i}{\partial t} + \boldsymbol{v} \cdot \nabla\varphi_i, \quad \forall i = 1, \ldots, N(p). \tag{13}$$

As the basis functions move with the grid velocity, it is then observed that the substantial derivative of the basis functions vanishes with the grid motion, i.e. $\mathrm{d}\varphi/\mathrm{d}t = 0$. Therefore, we have

$$\frac{\partial \varphi_i}{\partial t} = -\boldsymbol{v} \cdot \nabla\varphi_i, \quad \forall i = 1, \ldots, N(p), \tag{14}$$

where $\boldsymbol{v}=(v_1,\ldots,v_d)$ is the moving velocity of the given region in d-dimensional space. Eq. (12) can then be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\kappa(t)}\varphi_i\boldsymbol{u}_h\,\mathrm{d}\boldsymbol{x}=\int_{\kappa(t)}\nabla\varphi_i\cdot(\boldsymbol{F}(\boldsymbol{u}_h)-\boldsymbol{v}\boldsymbol{u}_h)\,\mathrm{d}\boldsymbol{x}-\oint_{\partial\kappa(t)}\varphi_i\mathcal{F}(\boldsymbol{u}_h^+,\boldsymbol{u}_h^-,\boldsymbol{n},v_n)\,\mathrm{d}s,\quad\forall\kappa\in\mathcal{T}_h(t)\ \forall i=1,\ldots,N(p).\tag{15}$$

In the above expression, the original flux function is modified to reflect the movement of the grid. Substituting the discrete approximation of the solution (8) into the above expression, one can obtain a semi-discrete system of ODEs as

$$(\boldsymbol{M}\overline{\boldsymbol{U}_h})_t=\boldsymbol{R}(\overline{\boldsymbol{U}}_h),\tag{16}$$

where $\overline{\boldsymbol{U}}_h$ is the global discrete solution vector, and $\boldsymbol{M}$ is the block diagonal mass matrix for which each block is an element mass matrix computed as

$$\boldsymbol{M}_{ij|\kappa}=\int_\kappa\varphi_i\varphi_j\,\mathrm{d}\boldsymbol{x},\quad\forall i=1,\ldots,N,\ \forall\kappa\in\mathcal{T}_h.\tag{17}$$

Here, $\boldsymbol{R}(\overline{\boldsymbol{U}}_h)$ is the residual vector given as

$$\boldsymbol{R}(\overline{\boldsymbol{U}}_h)_{i|\kappa}=\int_\kappa\nabla\varphi_i\cdot(\boldsymbol{F}(\boldsymbol{u}_h)-\boldsymbol{v}\boldsymbol{u}_h)\,\mathrm{d}\boldsymbol{x}-\oint_{\partial\kappa}\varphi_i\mathcal{F}(\boldsymbol{u}_h^+,\boldsymbol{u}_h^-,\boldsymbol{n},v_n)\,\mathrm{d}s,\quad\forall\kappa,\ \forall i=1,\ldots,N.\tag{18}$$

The system of ordinary differential equations (16) can be integrated in time using either explicit or implicit time-stepping algorithms. In this work, an explicit fourth order TVD Runge–Kutta method (Gottlieb and Shu, 1998) is employed to advance solutions in time. The integrals in (17) and (18) are evaluated using Gaussian quadrature rules (Solin et al., 2004). A quadrature rule for the integral over a domain of line segment, triangle or tetrahedral used in our computation is generally described as

$$\int_\Omega f(\boldsymbol{x})\,\mathrm{d}\Omega\approx\sum_g w_g f(\boldsymbol{x}_g),\tag{19}$$

where w_g denotes the weight and $\boldsymbol{x}_g$ is the coordinate of the quadrature point. The number of quadrature points depends on the order of the approximation and the quadrature rule used. It is known that in order to obtain $(p+1)$ order of accuracy for p th order DG methods (DGPp), the quadrature rules over each of the faces of the elements must be exact for polynomial of order $(2p+1)$ and the quadrature approximation for the integral over the interior of the elements (element boundaries) must be exact for polynomial of order $2p$ (Cockburn and Shu, 2001).

2.3. Geometric conservation law

In simulations of flow problems involving moving boundaries, it is important to ensure that a numerical scheme reproduces exactly a constant solution. This preservation of constant solution is referred to as the geometric conservation law (GCL), which simply states that a solution of a uniform flow under any numerical discretization scheme should be preserved exactly with an arbitrary mesh motion. The concept of GCL was first introduced in Thomas and Lombard (1979) and later applied to the design of numerical methods for simulations of flows over moving domains in the context of the finite difference as well as the volume method (Guillard and Farhat, 2000). The necessity of the GCL condition was also studied in Lomtev et al. (1999) and Persson et al. (2009) for DG discretization with moving boundaries. Compliance to the GCL condition in the framework of ALE is believed to ensure at least first-order accuracy in time (Guillard and Farhat, 2000) and possibly time stability of numerical schemes.

Mathematically, it must be shown that the ALE formulation (15) satisfies the uniform flow exactly. Inserting a constant solution, $\boldsymbol{u}(x,t)=\boldsymbol{u}_0$, into (15) and using the consistency property of numerical fluxes,

$$\mathcal{F}(\boldsymbol{u}_0,\boldsymbol{u}_0,\boldsymbol{n},v_n)=\boldsymbol{F}(\boldsymbol{u}_0)-\boldsymbol{v}\cdot\boldsymbol{n}\boldsymbol{u}_0,\tag{20}$$

we obtain the following expression after rearrangement:

$$\boldsymbol{u}_0\frac{\mathrm{d}}{\mathrm{d}t}\int_{\kappa(t)}\varphi_i\,\mathrm{d}\boldsymbol{x}=\boldsymbol{F}(\boldsymbol{u}_0)\left(\int_{\kappa(t)}\nabla\varphi_i\,\mathrm{d}\boldsymbol{x}-\oint_{\partial\kappa(t)}\varphi_i\boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}s\right)+\boldsymbol{u}_0\left(\oint_{\partial\kappa(t)}\varphi_i\boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}s-\int_{\kappa(t)}\nabla\varphi_i\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}\right),\quad\forall\kappa\in\mathcal{T}_h(t),\ \forall i=1,\ldots,N(p).\tag{21}$$

Applying the divergence theorem, the integrals associated with the flux function vanish and the above equation is rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\kappa(t)} \varphi_i \,\mathrm{d}\boldsymbol{x} = \oint_{\partial\kappa(t)} \varphi_i \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}s - \int_{\kappa(t)} \nabla\varphi_i\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}, \quad \forall\kappa\in\mathcal{T}_h(t),\ \forall i. \tag{22}$$

The time derivative of the integral on the left can be further expanded as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\kappa(t)} \varphi_i \,\mathrm{d}\boldsymbol{x} = \int_{\kappa(t)} \frac{\partial\varphi_i}{\partial t}\,\mathrm{d}\boldsymbol{x} + \oint_{\partial\kappa(t)} \varphi_i \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}s, \quad \forall i. \tag{23}$$

Substituting (23) into (22), we have

$$\int_{\kappa(t)} \left(\frac{\partial\varphi_i}{\partial t} + \nabla\varphi_i\cdot\boldsymbol{v}\right)\mathrm{d}\boldsymbol{x} = 0, \quad \forall i. \tag{24}$$

It is observed that expression (24) is always satisfied due to the fact that the basis functions move with the grid velocity as stated in (14). In other words, the ALE DG discretization (15) preserves the constant solution exactly and thus implies the GCL condition.

2.4. Moving mesh velocity

In the ALE formulation, the moving mesh velocity $\boldsymbol{v}$ can be arbitrarily found such that the moving velocity at the interface must be equal to the boundary velocity $\boldsymbol{v}_\Gamma$, either prescribed for fixed boundaries or obtained from the structure response in applications of fluid–structure interaction. Due to the movement of the boundaries, the computational mesh is deformed and possibly fails to preserve its quality. In the context of the ALE framework, one has to either regularize the grid moving velocity or decide to do the remeshing after certain steps, which often leads to increased computational cost, loss of accuracy and conservation. Deriving a robust and efficient scheme of constructing mesh velocity could help in reducing the effort of remeshing and thus the overhead. This has motivated many different approaches for computing the grid velocity, including the velocity smoothing technique.

Generally, the mesh velocity is directly computed from the velocity of the moving boundaries in order to minimize the grid distortion. The mesh velocity can be obtained from solving the equation

$$\nabla\cdot(k(\boldsymbol{x})\nabla\boldsymbol{v}(\boldsymbol{x})) = 0, \tag{25}$$

with diffusivity of $k(\boldsymbol{x})$ and Dirichlet boundary condition

$$\boldsymbol{v} = \boldsymbol{v}_\Gamma, \quad \boldsymbol{x}\in\Gamma. \tag{26}$$

One can discretize Eq. (25) using the finite element method to obtain the grid velocity. A simple uniform Laplacian smoothing is recovered if one sets $k(\boldsymbol{x}) = 1$; however, it does not avoid edge crossing and element collapsing, especially for the nonuniform grid under motion. Alternatively, variable diffusivity approaches (Lohner and Yang, 1996) can be used to alleviate the edge crossing by setting the diffusive coefficient to be a function of the mesh characteristics, for example mesh sizes and node coordinates. In the current work, the grid velocity is updated using a simple approach similar to the one presented in Lomtev et al. (1999). Velocity at a node is obtained from the weighted average of velocity of neighbouring nodes shown in Fig. 1,

$$\boldsymbol{v}_0 = \sum_{i=1}^{N_e} w_i\boldsymbol{v}_i, \tag{27}$$

where N_e is the number of edges meeting at the current node and w_i is the associate weighted coefficient of corresponding the neighbouring node i. The coefficients w_i associated with a node are computed from the prescribed mesh size function $h(\boldsymbol{x})$, giving the desired element size. The mesh size function, usually specified by users, is not an actual size but a relative distribution of nodes over the domain. It could also be created from the distance function $d(\boldsymbol{x})$ to the boundaries or expressed in terms of curvatures computed from the distance function. The weight coefficient at a node is simply computed as

$$w(\boldsymbol{x}_i) = \frac{h(\boldsymbol{x}_i)}{\sum_{j=1}^{N_e} h(\boldsymbol{x}_j)}. \tag{28}$$

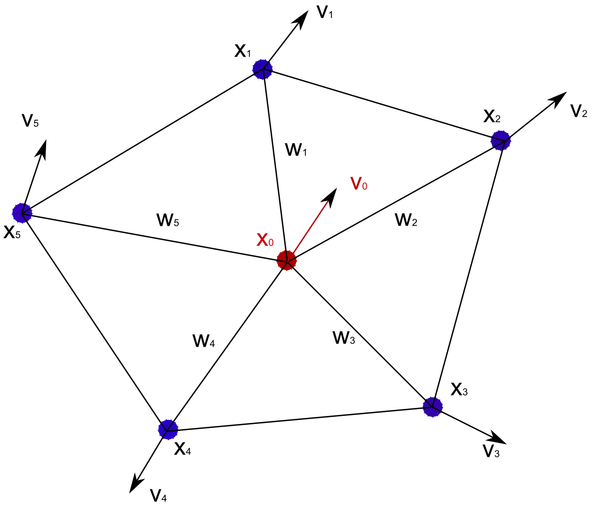

Fig. 1. Connectivity and velocity update at a nodes from its neighbours.

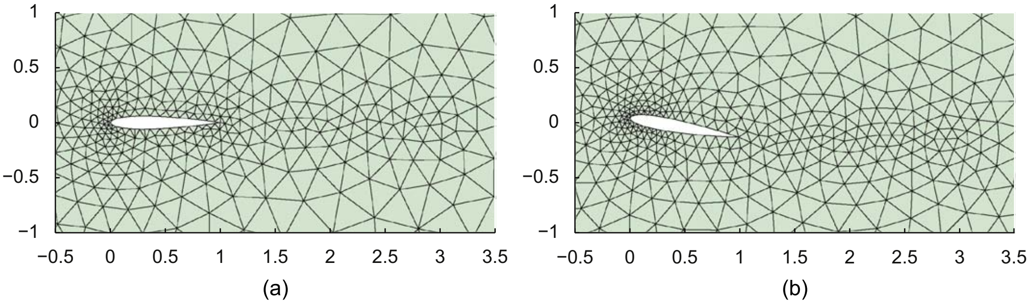

Fig. 2. Grids for pitching airfoil using moving mesh velocity technique for updating the grid nodes: (a) grid at low angle of attack; (b) grid at higher angle of attack. The grid quality is well maintained as well as the local features, adaptivity.

For a given interface velocity, the mesh velocity is iteratively computed, as in (27), until convergence, which often requires only a few iterations. The velocity of nodes on the boundary is directly obtained from the boundary conditions. This approach for mesh smoothing performs fairly well for problems with simple boundary movement, for example the pitching airfoil as shown in Fig. 2. It can maintain the mesh quality as well as the gird features. For a more complicated boundary deformation, this technique may not prevent elements from collapsing as well as badly shaped elements. However, it works well for most of our current simulations of moving domain problems with reasonably large deformations.

3. Numerical examples

3.1. Validation

The presented approach is validated first for a moving domain problem where the motion is prescribed. We consider a deformable domain with the grid motion defined as

$$x(t) = x_0 + X_0 \sin(n_t 2\pi t/t_0)\sin(n_x 2\pi x/L_x)\sin(n_y 2\pi y/L_y), \tag{29}$$

$$y(t) = y_0 + Y_0 \sin(n_t 2\pi t/t_0)\sin(n_x 2\pi x/L_x)\sin(n_y 2\pi y/L_y), \tag{30}$$

where (x, y) are the grid coordinates in two-dimensional space, $L_{x,y}$ are the domain sizes in the x and y direction, respectively, and t_0 denotes the reference time. Here, $n_{x,y}$ and n_t are the number of periods in space and time, respectively. Fig. 3 shows the motion of the grid in the space domain of $L_x = L_y = 20$ and time domain of $t_0 = \sqrt{5^2 + 5^2}$ during space and time period of $n_x = n_y = n_t = 1$ with amplitude of $X_0 = Y_0 = 0.5$.

3.1.1. Free stream preservation

In order to check the geometric conservation law for the proposed approach, we consider a uniform inviscid compressible flow in the square domain of $(x, y) \in [10 \times 10]$. The mesh motion is set as in above with $t_0 = 10$, $n_x = n_y = n_t = 1$ and amplitudes $X_0 = Y_0 = 0.5$. It was found that the error of the solution at $t = 0.5t_0$ with the initial condition $\boldsymbol{u}_0$ is very small and in the order of 10^{-8} for different approximation polynomials as shown in Table 1. This error, moreover, remains of the same order for various grid refinements in both L_2 and L_∞ norms. As the solution is constant, refinement of the polynomial interpolation order does not provide more accuracy but rather incurs more error in the approximation of integrations, as shown in Table 1 for L_2 norm of polynomial approximation order $p = 2$. However, the errors are found to be relatively small for various interpolation orders. It is again verified that the geometric conservation law is satisfied in the framework of the proposed ALE approach.

3.1.2. Propagation of isentropic vortex

Next, we consider the propagation of the isentropic vortex in Erlebacher et al. (1997), described as a perturbation to a mean flow of density ρ_∞, pressure p_∞, velocity (u_∞, v_∞) and temperature $T = p/\rho$. The perturbation in $(\tilde{u}, \tilde{v}, \tilde{T})$ centred at (x_0, y_0) has the form of

$$(\tilde{u}, \tilde{v}) = \varepsilon\tau e^{\alpha(1-\tau^2)}(-(y-y_0), (x-x_0)),$$

$$\tilde{T} = -\frac{(\gamma-1)\varepsilon^2 e^{2\alpha(1-\tau^2)}}{4\alpha\gamma}, \tag{31}$$

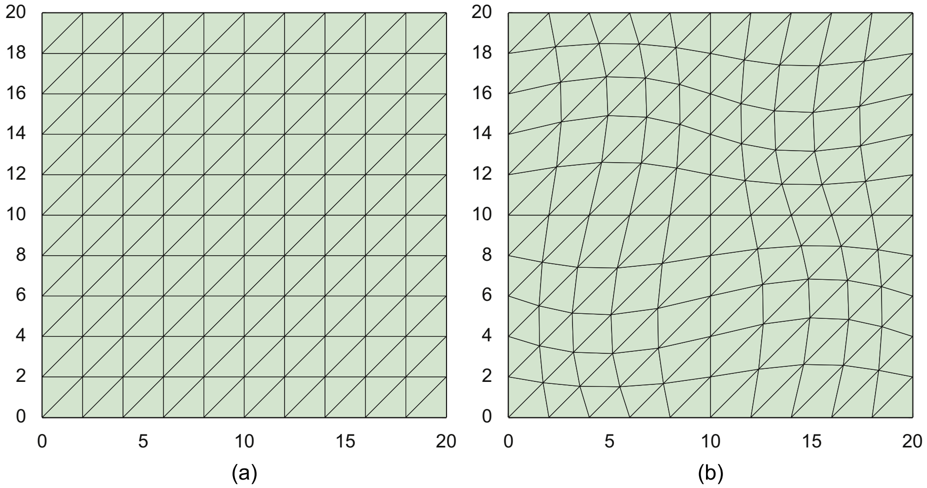

Fig. 3. Motion of the mesh with time period $n_x = 1.0$ and space period $n_x = n_y = 1.0$: (a) t=0; (b) t=5.0.

Table 1
Freestream flow preservation: error of computed solution versus free flow condition.

h	Elements	DGP1		DGP2		DGP3	
		$\lvert e\rvert_{L_2}$	$\lvert e\rvert_{L_\infty}$	$\lvert e\rvert_{L_2}$	$\lvert e\rvert_{L_\infty}$	$\lvert e\rvert_{L_2}$	$\lvert e\rvert_{L_\infty}$
4.0	50	5.36×10^{-7}	7.33×10^{-8}	1.19×10^{-6}	1.33×10^{-7}	7.22×10^{-7}	9.80×10^{-8}
2.0	200	3.48×10^{-7}	2.67×10^{-8}	9.41×10^{-7}	5.25×10^{-8}	4.75×10^{-7}	3.50×10^{-8}
1.0	800	2.51×10^{-7}	1.12×10^{-8}	8.52×10^{-7}	2.69×10^{-8}	3.47×10^{-7}	1.90×10^{-8}
0.5	3200	1.93×10^{-7}	4.89×10^{-9}	7.93×10^{-7}	1.26×10^{-8}	2.45×10^{-7}	7.72×10^{-9}

where $\tau = r/r_c$ and $r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$. Here, ε denotes the strength of the vortex, α and r_c are the decaying rate and the strength radius of the vortex, respectively.

It is known that the analytic solution at (x, y, t) is given as (Erlebacher et al., 1997)

$$u = u_\infty\left(1 - \frac{\varepsilon(y-y_0-v_\infty t)}{2\pi r_c}\mathrm{e}^{\alpha f(x,y,t)}\right), \quad v = v_\infty\left(1 + \frac{\varepsilon(x-x_0-u_\infty t)}{2\pi r_c}\mathrm{e}^{\alpha f(x,y,t)}\right),$$

$$\rho = \rho_\infty\left(1 - \frac{\varepsilon^2(\gamma-1)M_\infty^2}{16\alpha\pi^2}\mathrm{e}^{2\alpha f(x,y,t)}\right)^{1/(\gamma-1)}, \quad p = p_\infty\left(1 - \frac{\varepsilon^2(\gamma-1)M_\infty^2}{16\alpha\pi^2}\mathrm{e}^{2\alpha f(x,y,t)}\right)^{\gamma/(\gamma-1)}, \tag{32}$$

where $f(x, y, t) = (1-(x-x_0-u_\infty t)^2-(y-y_0-v_\infty t)^2)/r_c^2$. In this work, the vortex is initially located at $(x_0, y_0) = (2.5, 2.5)$ in the domain of 10×10. The free stream flow condition is $M_\infty = 1/\sqrt{\gamma}$ in the diagonal direction of $\theta = \pi/4$, with the pressure and density of being $p_\infty = 1$ and $\rho_\infty = 1$, respectively. The vortex has the strength of $\varepsilon = 5.0$, $\alpha = 0.5$ and $r_c = 1.0$.

The solution is integrated in time using the explicit fourth-order Runge–Kutta method. Fig. 4 shows errors between computed solutions and exact solution in L_2 norm for different orders of approximation $p = 1, 2, 3$. As the mesh is refined, it is observed that the solution converges at the expected rate.

3.2. Forced oscillation cylinder

We now consider a viscous flow over an oscillating cylinder. Motion of the cylinder is prescribed via position of the centre $y_c(t) = a\sin(2\pi ft)$, where f and a are the frequency and amplitude of the oscillation, respectively. The forced frequency of the oscillation is chosen to be in the range of the vortex-shedding frequency f_s of the wake expressed as non-dimensional Strouhal number, $\mathrm{St}_s = f_s D/U_0$, where U_0 is the free stream velocity and D is the diameter of the cylinder.

First, the cylinder is held fixed in space until the flow is fully developed. As shown in Braza et al. (1986), for a range of Reynolds numbers ($\mathrm{Re} = U_0 D/\nu$), a well-known Karman vortex pattern is formed in the wake of the cylinder due to the breaking of flow symmetry. In this study, simulations were performed for different Reynolds numbers ranging from $\mathrm{Re} = 80$ to 1000. Table 2 shows the results of the average drag coefficient and Strouhal number at $\mathrm{Re} = 500$ on different grids, as depicted in Fig. 5. The refinement of the grid is done by doubling the number of points along the circular surface, from a coarse grid with 32 points to a medium grid consisting 64 points and a fine grid with 128 points. The average drag coefficient and Strouhal number obtained from the medium grid is closer to the results obtained on the fine grid, which converges to the experimental data from Roshko (1954). The drag coefficient and Strouhal number obtained using the proposed approach are compared with the experimental data and other numerical calculations. In Fig. 6, the present results are plotted against the experimental data and numerical computation reported in Braza et al. (1986). As reported in Roshko (1954), the Strouhal number is in the range of 0.2–0.22 for the same Reynolds number

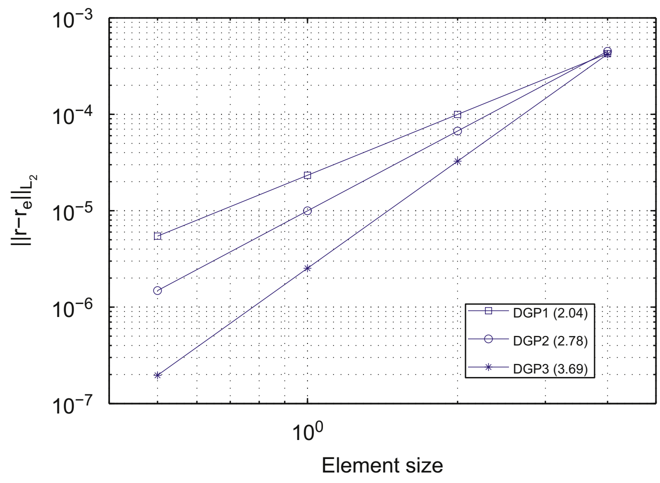

Fig. 4. Convergence of the ALE DG approach for the vortex problem. Solutions converge at the expected rate in the order of approximation polynomial $(p+1)$.

Table 2
Drag coefficient and Strouhal number at Re = 500.

	Mesh size ($h_{\min}$)	$\overline{C_d}$	St
Coarse mesh	0.1	1.4704	0.241
Medium mesh	0.05	1.4502	0.223
Fine mesh	0.025	1.4413	0.218
Experiment (Roshko, 1954)	–	–	0.20–0.22

Comparison of computational results on refined grids with experimental data extracted from Braza et al. (1986).

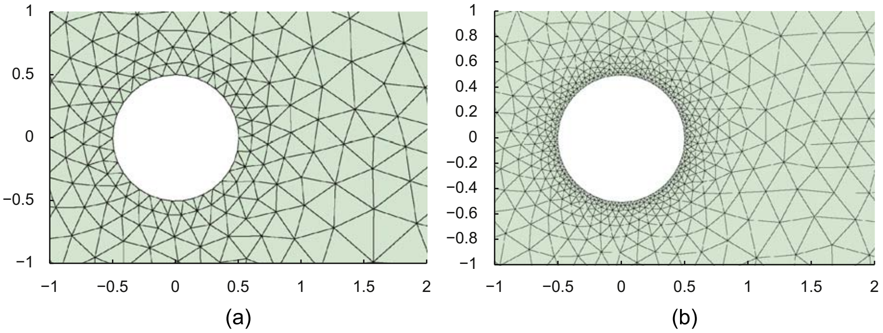

Fig. 5. Zooming in around the cylinder for different computational grids of (a) coarse grid with 32 nodes on the circular surface and (b) fine grid of 128 nodes on the circular surface.

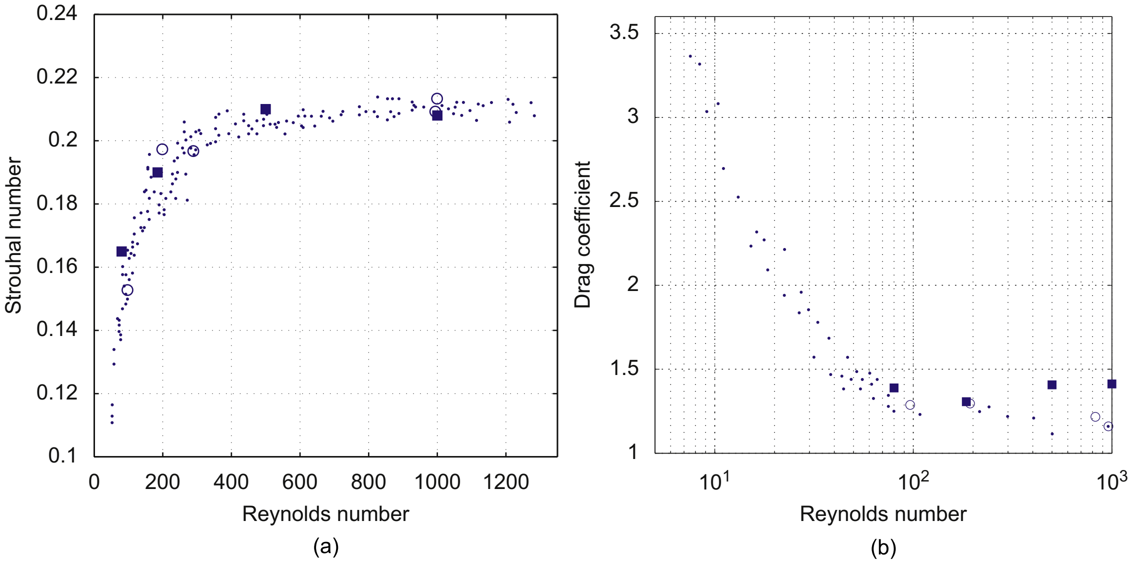

Fig. 6. Comparison of the present computational results with experiment and earlier simulations for flow over a fixed circular cylinder. (a) Strouhal number versus Reynolds number. (b) Average drag coefficient versus Reynolds number: •, experimental data of Roshko (1954); ○, Braza et al. (1986); ■, present.

Re ≥ 200. A good comparison of the vortex shedding frequency obtained from the current computation with the experimental results can be seen in Fig. 6(a). The average drag coefficient agrees well with the experimental data for lower Reynolds numbers, Re < 500. It is also observed that there is a discrepancy between the present result with the experimental data for Re = 500 and 1000. This could be due to the three-dimensional flow nature at higher Reynolds numbers and the fact that the present simulation is two-dimensional.

The cylinder is now forced to oscillate in the transverse direction with frequency f and amplitude a. This problem has been studied extensively over the past decades with a lot of both experimental and numerical results, including the experimental work in Tanida et al. (1973) as well as numerical study in Patnaik et al. (1999) and Lu and Dalton (1996) for two-dimensional flows over an oscillating cylinder. In this work, simulations are carried out for various combinations of frequencies and amplitude ratio in a range of Reynolds number $\mathrm{Re}=80-1000$. For the sake of brevity and comparison, only the results of variation in frequency ratio f/f_s at $\mathrm{Re}=80$ and amplitude ratio of $a/D=0.14$ are discussed in detail. Some comparison can be made with the available experimental and numerical data.

Fig. 7 shows the temporal history of lift and drag coefficients. It is well-known that, in a certain range of frequency ratio f/f_s, the vortex shedding is entrained by the cylinder motion. Thus, the excitation frequency is able to drive the vortex shedding frequency to match the cylinder oscillation frequency. This phenomenon is called lock-in. In the lock-in regime, the variation of the force coefficients, drag and lift, is in phase with the movement of the cylinder, as shown in Fig. 7 for frequency $0.8 \le f/f_s \le 1.2$. The stable periodic history of force coefficients implies a synchronization flow regime. The spectral plot of lift coefficient in Fig. 8 presents a single peak as a unique dominant frequency, indicating a lock-in regime. For $f/f_s=1.3$, it can be seen that there are two dominant frequencies driving the response of the system, as depicted in the spectral plot. The combination of these two dominant frequencies causes a ‘‘non-lock-in’’ response shown in a ‘‘beat’’ pattern of the force history. The range of the lock-in frequency ratio varies with Reynolds number and the amplitude of the oscillation of the cylinder. At $\mathrm{Re}=80$ and amplitude ratio of $a/D=0.14$, the present result of lock-in frequency range ($0.8 \le f/f_s \le 1.2$) compares well with the results reported in Patnaik et al. (1999). It is also noticed that the force coefficients are increased in the lock-in regime. Fig. 9 shows the mean drag coefficient versus excitation frequency. The present result is compared with the experimental data of Tanida et al. (1973) and the

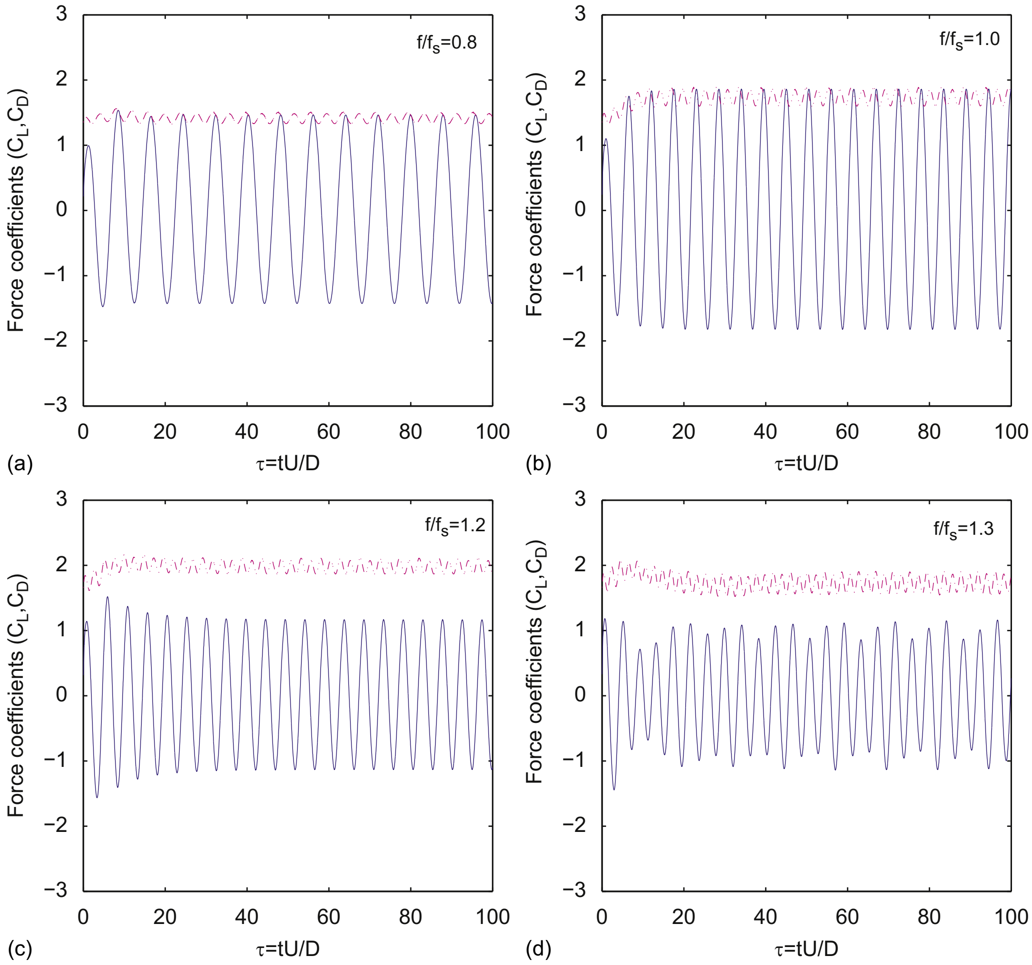

Fig. 7. Temporal history of lift and drag coefficients at $\mathrm{Re}=80$ for different frequency ratio of $f/f_s=0.8, 1.0, 1.2, 1.3$.

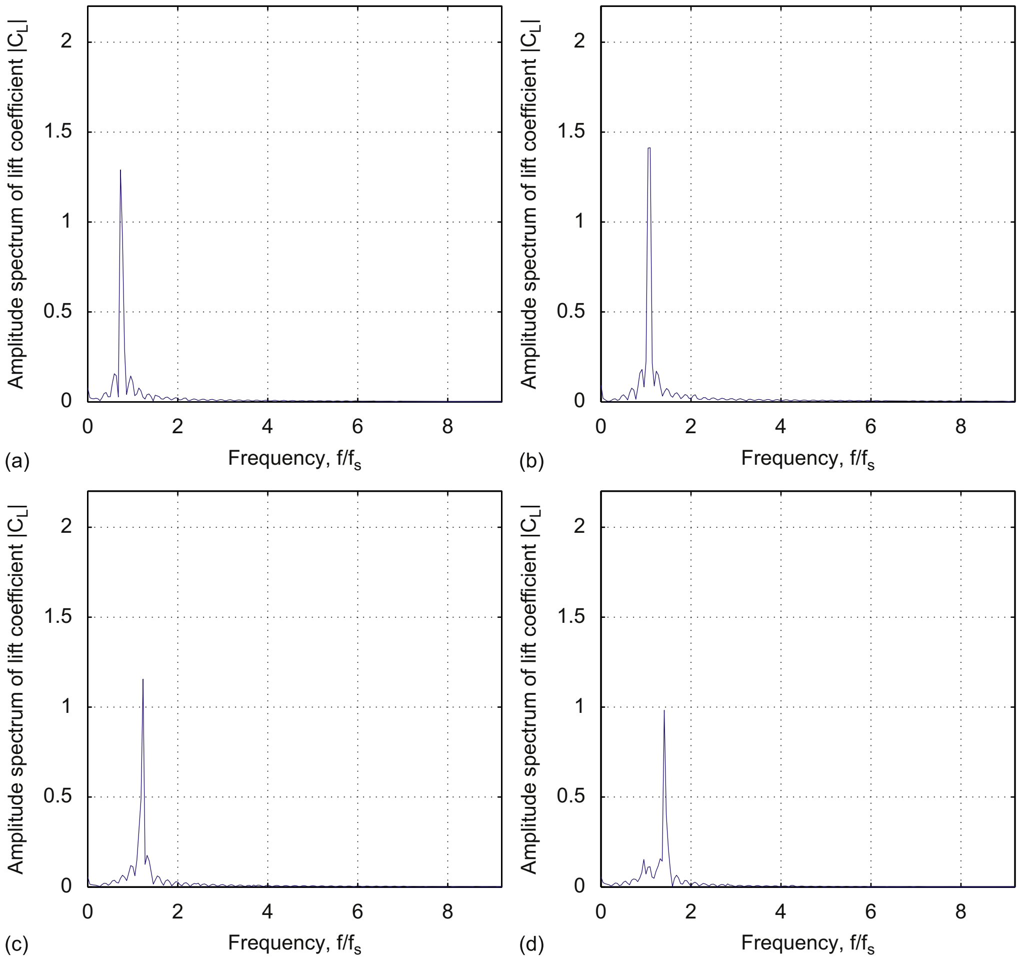

Fig. 8. Frequency response of lift coefficients for an oscillating cylinder at $\mathrm{Re} = 80$, amplitude $a/D = 0.14$, for a range of frequency. Note that there are two dominant frequencies in the non lock-in case ($f/f_s = 1.3$) including the natural shedding frequency and forcing frequency while there is only one dominant frequency in the lock-in case: (a) $f/f_s = 0.8$; (b) $f/f_s = 1.0$; (c) $f/f_s = 1.2$; (d) $f/f_s = 1.3$.

numerical result reported in Patnaik et al. (1999) for $\mathrm{Re} = 80$ and $a/D = 0.14$. The drag coefficient is sharply increased in the vicinity of $f/f_s = 1$. Compared to the experimental data, the peak is overestimated and shifted to the right by the present approach; this is probably due to the difference in the natural vortex shedding frequency between the present result ($f_s = 1.60$) with the earlier experiment ($f_s = 1.54$) and simulation ($f_s = 1.43$). However, the current computation matches fairly well with the experiment and the current result is able to predict the sharp increase in the drag coefficients more favourably than the earlier numerical result.

Interaction between the oscillation of the cylinder and vortex shedding also results in a change of flow structure in the wake. Williamson and Govardhan (2004) classified different shedding modes according to the number of vortex pairs (P) or single vortices (S). As described in Williamson and Govardhan (2004), there are different vortex wake patterns for a cylinder translating in a sinusoidal fashion. The wake vortex patterns may be formed by a combination of single vortices (S) and vortex pairs (P). It is also found that the vortex street in the downstream of the cylinder experiences a basic change at a certain threshold value of excitation frequency. A quantitative study by Gu et al. (1994) and the later numerical work of Lu and Dalton (1996) show that the initially formed concentration of vorticity moves closer to the cylinder as the excitation frequency is increased relatively to the natural vortex shedding frequency. As the limiting position is reached, the concentration of vorticity abruptly switches to the opposite side of the cylinder. Fig. 10 shows instantaneous vorticity contours for various excitation frequencies at the same Reynolds number of $\mathrm{Re} = 80$ and amplitude $a/D = 0.14$ when the cylinder is at its upper limit position. It can be seen that the length of the elongated vortex attached to the upper surface of the cylinder decreases as the excitation frequency is increased. The lower vortex

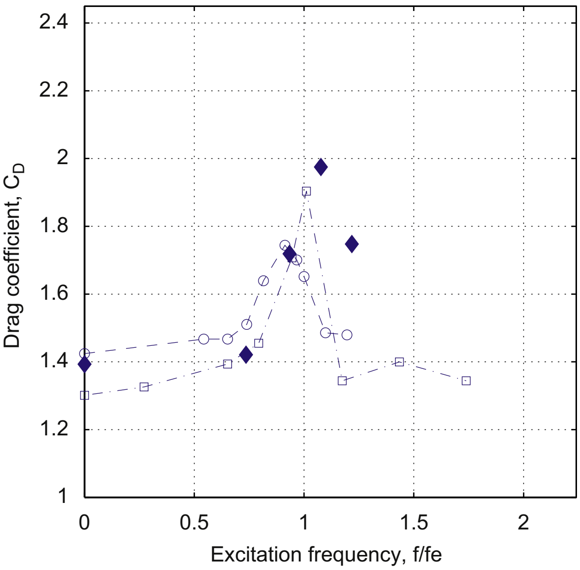

Fig. 9. Comparison of drag coefficient varying with excitation frequency at Re = 80 and $a/D = 0.14$: □, Tanida et al. (1973); ○, Patnaik et al. (1999); ◆, present.

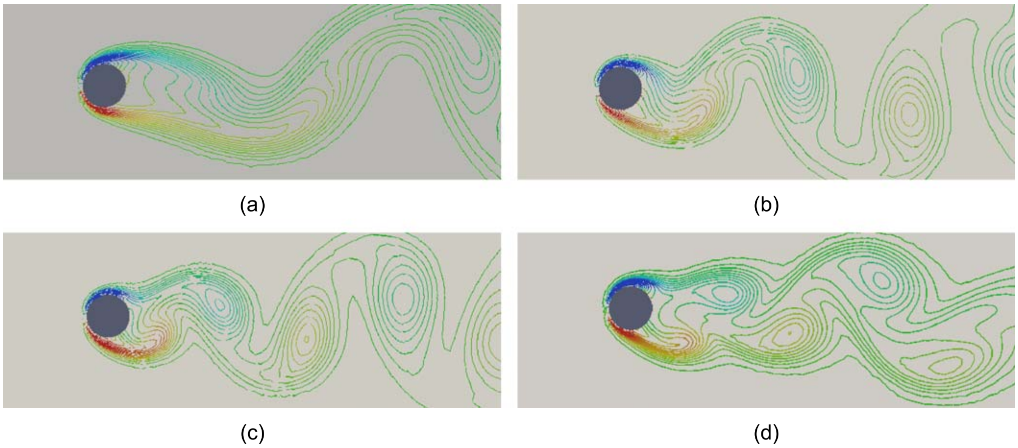

Fig. 10. Vorticity contours at Re = 80, amplitude $a/D = 0.14$; (a) f/f_s=0.9; (b) f/f_s=1.0; (c) f/f_s=1.2; (d) f/f_s=1.3. Note that the vorticity contours are plotted as the cylinder at its upper limit position.

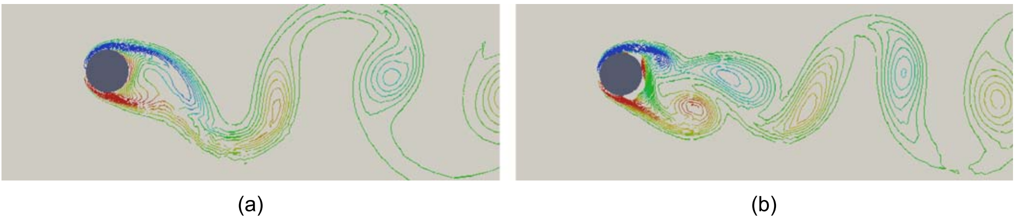

Fig. 11. Vorticity contours at Re = 185, amplitude $a/D = 0.4$; (a) f/f_s=0.9; (b) f/f_s=1.2. Note that the vorticity contours are plotted as the cylinder at its upper limit position.

formed due to the motion of the cylinder interacts with the upper vortex to diminish the vorticity. As the frequency is increased, the lower vortex becomes the dominant vortex and the upper vortex rolls up tightly in the wake of the cylinder. This phenomenon was described in the earlier experimental and numerical studies. It is also found that there is a critical excitation frequency corresponding to a particular Reynolds number and oscillation amplitude at which the switching occurs. Fig. 11 shows instantaneous vorticity contours with different excitation at $\mathrm{Re}=185$ and $a/D=0.4$ when the cylinder is reaching its upper limit position. The critical excitation frequency f/f_e^* found in Gu et al. (1994) is $f/f_e^*=1.12$ and the switching of the flow patterns in the wake of the cylinder is captured in our current numerical results. Fig. 11 shows the difference in flow structure in the wake of the cylinder for low frequency ($f/f_e=0.9$) and high frequency ($f/f_e=1.2$). The contour plots compare fairly well with the one reported in Lu and Dalton (1996) at the same Reynolds number and amplitude ratio where the smaller lower vortex at low frequency becomes the bigger vortex compared with the upper vortex at higher frequency.

3.3. Forward flapping flight

In this example, we consider a forward flapping flight, one of the basic modes of flying in insects. The proposed ALE DG method is applied for simulation of viscous flow over a flapping wing section in two-dimensional space at Reynolds number in the range of insect flight. The results of unsteady effects and forces generated on the wing section are computed for different flapping modes for varying flapping frequency and amplitude. The present simulation is verified with earlier studies in Wang (2000).

As a simple model in two dimensions, the wing section in the chord direction is modelled as an ellipse of thickness ratio 12.5% and chord length of $c=1$. The wing is moving forward with velocity u_0 while flapping with velocity

$$u_f(t)=2\pi fA\sin(2\pi ft) \tag{33}$$

at an inclined angle β to the horizontal direction; f and A are the flapping frequency and amplitude, respectively. The unsteady effect of flow over the flapping foil is characterized by nondimensional numbers including the advance ratio and Strouhal number. The ratio of forward and maximum flapping velocity is defined as the advance ratio,

$$J=\frac{2\pi fA}{u_0}. \tag{34}$$

The Strouhal number or nondimensional flapping frequency is given as

$$\mathrm{St}_c=\frac{fc}{u_0}. \tag{35}$$

In this test, we study a viscous flow of $\mathrm{Re}=1000$ over the flapping wing with $\beta=\pi/2$, zero angle of attack in forward flight and a fixed advance ratio of $J=1.0$. A circular computational domain is discretized into an unstructured grid, which is refined at the leading edge and trailing edge as depicted in Fig. 12. By varying the flapping frequency, the coefficients of lift and drag acting on the ellipse, measured in the parallel and orthogonal directions to the forward flying velocity u_0 can be obtained. However, in studying flapping flight, it is more convenient to show the forces in the

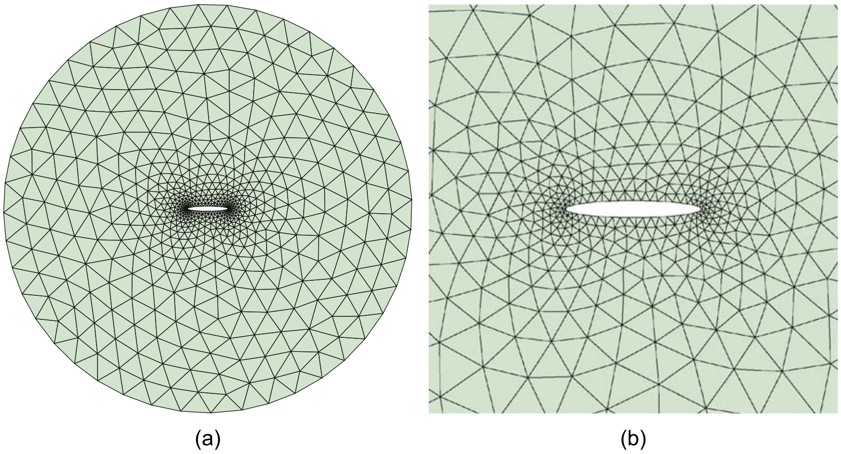

Fig. 12. Computational grid of flow over a forward flapping wing section (a) and zoomed in region (b) near the foil surface where the grid points are clustered in the leading edge and trailing edge region.

direction of effective flow velocity U at angle of attack α, with U as the sum of u_0 and u_f, $\alpha = \tan^{-1}(u_f(t)/u_0)$. The forces measured in the parallel and orthogonal direction to the velocity U are denoted as axial and normal forces (C_A and C_N) in this study. Thus the time-averaged forces can be computed as

$$\langle C_A \rangle = \frac{1}{T}\int_0^T C_A(t)\,\mathrm{d}t, \tag{36}$$

where T is the time of oscillation. This average force in mean flow direction corresponds to the thrust in flapping flight.

In Fig. 13, the history of lift and drag coefficients as well as axial and normal force coefficients in nondimensional time at flapping frequency of $\mathrm{St}_c = 1.0$ are shown. It can be seen that due to the symmetric flapping motion of the wing at this frequency, the lift coefficient (C_L) symmetrically varies about zero mean, and thus the average lift is zero. The drag coefficient is asymmetric and oscillating with a frequency twice that for the lift coefficient. As projected in the effective flow direction, thrust is defined as the negative value of axial force (C_A). It can be seen that the thrust is

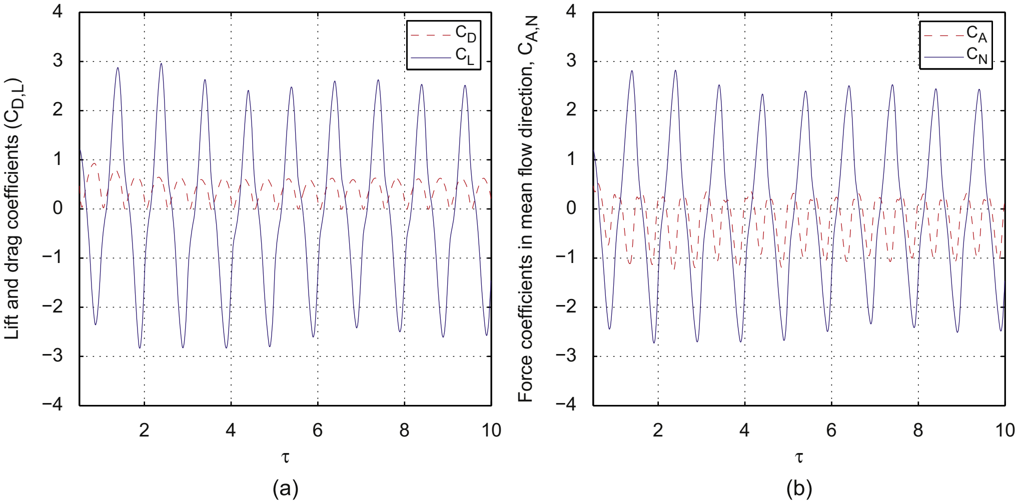

Fig. 13. Force coefficients acting on the flapping foil at $\mathrm{Re} = 1000$, $J = 1$ and $\mathrm{St}_c = 1$.

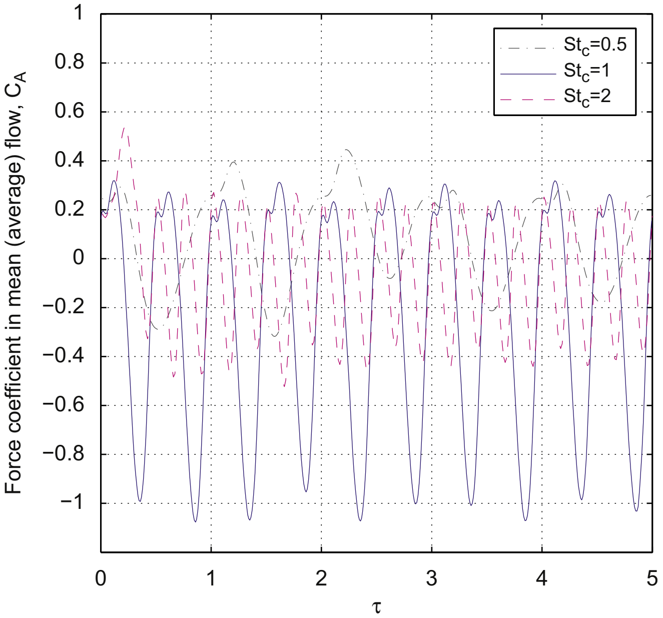

Fig. 14. Thrust coefficients at $\mathrm{Re} = 1000$, $J = 1$ and different flapping frequencies of $\mathrm{St}_c = 0.5, 1.0, 2.0$.

Table 3
Average thrust coefficient for different Strouhal number and comparison between the present results with the earlier study.

St_c	0.5	1.0	2.0
$\langle C_A \rangle$ (Wang, 2000)	−0.08	−0.24	−0.06
$\langle C_A \rangle$ (present)	0.0176	−0.225	−0.058

generated in both up and down strokes and a positive average value of thrust is observed in this case. The thrust coefficients for different flapping frequency are shown in Fig. 14 for $\mathrm{St}_c = 0.5$, 1.0, 2.0. In Table 3, the average force coefficients are computed and compared with those reported in Wang (2000), in which the author employed a compact finite difference scheme for simulations of flow over a flapping wing in elliptic coordinates by solving the incompressible Navier–Stokes equations in vorticity-stream function formulation. The present results agree fairly well with this earlier study. The average thrust coefficient is maximum at $\mathrm{St} = 1.0$, which is also observed in the earlier numerical experiment. It is of interest to note the agreement of the results from the present work with the earlier numerical experiment. The two approaches are basically different, as the earlier study solved the Navier–Stokes equations in vorticity form in elliptical coordinates attached to the body frame. The matching of results using the proposed approach again shows the flexibility and accuracy of the method for resolving unsteady aerodynamic phenomena.

4. Conclusion

We have developed a new approach for simulations of compressible Navier–Stokes flows over deformable domains and boundaries. The method employs high-order discontinuous Galerkin spatial discretization in the ALE framework together with the explicit TVD Runge–Kutta scheme. Using the current approach, we are able to obtain high-order accuracy for moving domain problems while the geometry conservation law is satisfied to ensure the preservation of constant solutions. As for the grid movement, we have presented a simple and effective approach of grid velocity smoothing based on local smoothing with variable diffusivity computed from mesh size functions. It has been shown that this technique is able to handle variable geometries with reasonably large deformation. The application of the proposed approach to some moving boundary problems in two dimensions has shown the accuracy of the methods especially in resolving the flow features in the wake which result from interactions between flow and moving boundaries. The results compare fairly well with the available published results as well as experiment.

Appendix A. Boundary conditions

It is considered as one of the advantages of the DG method that the boundary conditions are much simpler to implement than the other high-order approaches, which require more complex treatment near the boundaries. In the implementation of the DG method, the boundary conditions are weakly imposed by evaluating the left and right states of solutions along the boundaries to compute the numerical fluxes. We briefly describe in detail the frequently used boundary conditions in this work, including inflow/outflow boundary conditions and solid boundary conditions.

A.1. Inflow/outflow boundary condition

For inflow/outflow conditions, the state value at the boundaries ($\boldsymbol{u}_h^b$) is determined by using the outgoing Riemann invariants in the normal direction to the boundary and given boundary data dependent on the nature of the flow at the boundaries. For example, if the inflow is in the subsonic regime ($0 < u_n < c$), the boundary solution is computed from the total temperature and total pressure set at the boundary and the interior invariants. In the case of subsonic outflow ($-c < u_n < 0$), the boundary solution is computed from only the set boundary pressure and the invariants on the interior. As the boundary state is determined, the numerical flux across the boundary can be computed using the numerical flux formulations, e.g. Lax-Friedrichs flux. It is not trivial to implement the viscous flux from the above set of conditions applied for $\boldsymbol{u}_h^b$. The viscous flux is usually computed by interpolating the solution derivatives across the boundary.

A.2. Solid boundary condition

The solid wall boundary condition is usually imposed in flow problems using the reflection technique. All of the interior state is reflected symmetrically to the "ghost" state outside the domain in the normal direction. Then the Riemann problem is solved on the boundary. Due to the symmetry of the reflection, the boundary flux obtained from the Riemann solver involves only the pressure contribution. Essentially, it is stated via the reflecting boundary that no flow can penetrate a solid wall, i.e.,

$$u_i \boldsymbol{n}_i^{\text{wall}} = u_i^{\text{wall}} \boldsymbol{n}_i^{\text{wall}}, \tag{37}$$

where u_i^{wall} and $\boldsymbol{n}^{\text{wall}}$ are the wall velocity and normal vector, respectively. At the stationary boundary for inviscid flow calculation, the normal velocity is set to zero, $u_i \boldsymbol{n}_i^{\text{wall}} = 0$. A symmetric boundary condition is applied to specify the state condition which has the same density, internal energy and tangential velocity as the internal components.

For viscous flow the particles are stuck to the wall; therefore, $u_i = 0$. In addition to the conditions applied to the velocity, temperature needs to be specified at the boundary. Basically there are two types of conditions: isothermal and adiabatic. For an isothermal wall, the static temperature is set $T_b = T_{\text{wall}}$. This condition, together with the static pressure specified from the interior condition, $P_b = (\gamma - 1)\rho E$, are sufficient to determine the full state solution at the boundary as

$$\boldsymbol{u}_h^b = \left[\frac{P_b}{RT_b} \;\; 0 \;\; 0 \;\; 0 \;\; P_b/(\gamma - 1) \right]^{\mathrm{T}}. \tag{38}$$

No physical condition is set for viscous flux; therefore, it is computed by interpolating the gradient of solution across the boundary.

For an adiabatic wall, the heat transfer across the boundary is specified instead. Thus, the Neumann boundary condition is prescribed for temperature

$$\left. \frac{\partial T}{\partial \boldsymbol{n}} \right|_{\text{wall}} = 0. \tag{39}$$

By specifying two other conditions, static pressure and enthalpy, which are interpolated from the interior

$$P_b = (\gamma - 1)\rho E, \quad H_b = \frac{\rho E + P_b}{\rho}, \tag{40}$$

the full state vector of solution at the boundary is determined

$$\boldsymbol{u}_h^b = \left[\frac{\gamma}{\gamma - 1} \frac{P_b}{H_b} \;\; 0 \;\; 0 \;\; 0 \;\; \rho E \right]^{\mathrm{T}}. \tag{41}$$

Under the adiabatic condition at the solid wall, it is required that the viscous flux associated with the energy equations is set to zero. The other quantities are determined from interpolation as usual.

References

Braza, M., Chassaing, P., Ha Minh, H., 1986. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. Journal of Fluid Mechanics 165, 79–130.

Cockburn, B., Shu, C.-W., 2001. Runge–Kutta discontinuous Galerkin methods for convective-dominated problems, review article. Journal of Scientific Computing 16, 173–261.

Donea, J., Huerta, A., 2003. Finite Element Methods for Flow Problems. Wiley, New York.

Erlebacher, G., Hussaini, M.Y., Shu, C.-W., 1997. Interaction of a shock with a longitudinal vortex. Journal of Fluid Mechanics 337, 129–153.

Gottlieb, S., Shu, C.-W., 1998. Total variation diminishing Runge–Kutta schemes. Mathematics Contemporary 67, 73–85.

Guillard, H., Farhat, C., 2000. On the significance of the geometric conservation law for flow computations on moving meshes. Computer Methods in Applied Engineering 190, 1467–1482.

Gu, W., Chyu, C., Rockwell, D., 1994. Timing of vortex formation from an oscillating cylinder. Physics of Fluids 6, 3677–3682.

Hirt, C.W., Amsden, A.A., Cook, J.L., 1997. An arbitrary Lagrangian–Eulerian computing method for all flow speed. Journal of Computational Physics 135, 203–216.

Koornwinder, T.H., 1992. Askey–Wilson polynomials for root systems of type BC. Contemporary Mathematics, vol. 138, pp. 189–204.

Lohner, R., Yang, C., 1996. Improved ALE mesh velocities for moving bodies. Communications in Numerical Methods in Engineering 12, 599–608.

Lomtev, I., Kirby, R.M., Karniadakis, G.E., 1999. A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. Journal of Computational Physics 155, 128–159.

Lu, X.-Y., Dalton, C., 1996. Calculation of the timing of vortex shedding from an oscillating cylinder. Journal of Fluids and Structures 10, 527–541.

Patnaik, B.S.V., Narayana, P.A.A., Seetharamu, K.N., 1999. Numerical simulation of laminar flow past a transversely vibrating circular cylinder. Journal of Sound and Vibration 288, 459–475.

Persson, P.O., Peraire, J., Bonet, J., 2009. Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains. Computer Methods in Applied Mechanics and Engineering 198, 1585–1595.

Roshko, A., 1954. Vortex shedding from circular cylinder at low Reynolds number. NACA Report No. 1191.

Solin, P., Segeth, K., Dolezel, I., 2004. Higher-Order Finite Element Methods. Chapman & Hall, New York.

Tanida, Y., Okajima, A., Watanabe, Y., 1973. Stability of a circular cylinder oscillating in uniform flow and in wake. Journal of Fluid Mechanics 61, 769–784.

Thomas, P.D., Lombard, C.K., 1979. Geometric conservation law and its applications to flow computations on moving grids. AIAA Journal 17, 1030–1037.

van de Vegt, J.J.W., Xu, Y., 2007. Space-time discontinuous Galerkin method for nonlinear water waves. Journal of Computational Physics 224, 17–39.

Ventakasubban, C.S., 1995. A new finite element formulation for ALE (arbitrary Lagrangian Eulerian) compressible fluid mechanics. International Journal of Engineering Science 33, 1743–1762.

Wang, Z.J., 2000. Vortex shedding and frequency selection in flapping flight. Journal of Fluid Mechanics 410, 323–341.

Wang, Z.J., 2005. Dissecting insect flight. Annual Review of Fluid Mechanics 37, 183–210.

Williamson, C.H.K., Govardhan, R., 2004. Vortex-induced vibrations. Annual Review of Fluid Mechanics 36, 413–455.