Computers & Fluids 48 (2011) 84–97

Contents lists available at ScienceDirect

Computers & Fluids

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

A high-order spectral difference method for unstructured dynamic grids

M.L. Yu *, Z.J. Wang, H. Hu

Department of Aerospace Engineering, Iowa State University, Ames, IA 50011, United States

ARTICLE INFO

Article history:
Received 8 May 2010
Received in revised form 22 February 2011
Accepted 31 March 2011
Available online 12 April 2011

Keywords:
High-order
Unstructured dynamic grids
Spectral difference
Navier–Stokes
Bio-inspired flow

ABSTRACT

A high-order spectral difference (SD) method has been further extended to solve the three dimensional compressible Navier–Stokes (N–S) equations on deformable dynamic meshes. In the SD method, the solution is approximated with piece-wise continuous polynomials. The elements are coupled with common Riemann fluxes at element interfaces. The extension to deformable elements necessitates a time-dependent geometric transformation. The Geometric Conservation Law (GCL), which is introduced in the time-dependent transformation from the physical domain to the computational domain, has been discussed and implemented for both explicit and implicit time marching methods. Accuracy studies are performed with a vortex propagation problem, demonstrating that the spectral difference method can preserve high-order accuracy on deformable meshes. Further applications of the method to several moving boundary problems including bio-inspired flow problems are shown in the paper to demonstrate the capability of the developed method.

1. Introduction

Computational fluid dynamics (CFD) has attracted a surge of research activities during the last three decades, and it has become a routine tool in the aerodynamic design of aircraft, wind turbines, centrifugal pumps, etc. For general engineering applications, nearly all production flow solvers are based on at most second-order numerical methods. Although they proved very useful, the second-order methods may not be accurate enough for problems requiring high accuracy, such as vortex-dominated flows, and acoustic noise predictions. Therefore, there has been a growing interest in the development of high-order methods for unstructured grids in recent years. The reasons for this are obvious. High-order methods enjoy remarkably high accuracy with low numerical dissipations, and unstructured grids can provide flexibility in handling complex geometries. A review of the high-order methods for the Euler and Navier–Stokes equations can be found in [31].

The spectral difference (SD) method [12] is a recently developed high-order method to solve compressible flow problems on simplex meshes. Its precursor is the conservative staggered-grid Chebyshev multi-domain method [11]. The general formulation of the SD method was first described in [12] and applied for computational electromagnetic problems. It is then extended to 2D Euler [33] and Navier–Stokes equations [14,34]. After that, Sun et al. [22,23] implemented the SD method for 3D N–S equations on unstructured hexahedral meshes. Later, a weak instability in the original SD method was found independently by Vanden Adeele et al. [26] and Huynh [8]. Huynh [8] further found that the use of Legendre–Gauss quadrature points as flux points results in a stable SD method. This was later proved by Jameson [9] for the one dimensional linear advection equation. The present study is based on Sun et al. [22,23] and further extends the method to 3D deformable meshes. The basic idea to achieve high-order accuracy in the SD method is to use a high degree polynomial to approximate the exact solution in a standard element (a local cell). However, unlike the discontinuous Galerkin (DG) [3] method and spectral volume (SV) method [32], the SD method is in the differential form, which is efficient and simple to implement. As all the computations are performed on the fixed standard element in the computational domain, it is reasonable to expect that the SD method can preserve high-order features for moving boundary problems in the physical domain.

Since a time-dependent curvilinear transformation from the physical domain to the standard element is needed in the SD method, the Geometric Conservation Law (GCL), first discussed in [25], should be strictly enforced in order to eliminate the grid motion induced errors. For high-order methods, an approach to guarantee GCL for the finite difference method has been proposed in [27]. It is straightforward to extend this approach to the present SD method. In addition, a GCL compliant high-order time integration method is developed for the implicit scheme with a similar method used in [13]. Note that there is an alternative way to deal with moving boundary problems, which is called the arbitrary Lagrangian–Eulerian (ALE) method [4]. In that approach, a mapping from a fixed reference configuration to the physical domain is needed. In

* Corresponding author.
E-mail addresses: mlyu@iastate.edu (M.L. Yu), zjw@iastate.edu (Z.J. Wang), huhui@iastate.edu (H. Hu).

0045-7930/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2011.03.015

the mapping, a time-dependent GCL is introduced for the reference domain [17–19]. It is quite similar to the coordinate transformation approach aforementioned in the SD or the finite difference methods in [24,27]. It can be shown that the final form of the time-dependent GCL is exactly the same for both approaches.

The remainder of the paper is organized as follows. In Section 2, the SD method is briefly reviewed including both the space discretization procedure and time integration approach. The GCL of the transformation from the physical domain to the computational one is then discussed in detail. After that, the implementation of GCL into the numerical schemes is described for different time marching methods. An algebraic grid deformation method together with the corresponding blending strategy is given in Section 2 as well. Then several numerical test cases are presented in Section 3. For a single flapping airfoil, the numerical results are obtained with both a rigid moving grid and a deformable grid. The comparisons of these results with experimental data are also presented. Moreover, some superior features of high-order methods over the lower ones are also illustrated in Section 3. Section 4 briefly concludes the paper.

2. Numerical method

2.1. Governing equations

We consider the unsteady compressible Navier–Stokes (N–S) equations in conservation form in the physical domain (t,x,y,z)

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}+\frac{\partial H}{\partial z}=0, \tag{2.1}$$

where Q is the vector of conservative variables, and F, G, H are the total fluxes including both the inviscid and viscous flux vectors.

After introducing a time-dependent coordinate transformation (Fig. 1a) from the physical domain (t,x,y,z) to the computational domain (τ,ξ,η,ζ), Eq. (2.1) can be rewritten as

$$\frac{\partial \widetilde{Q}}{\partial \tau}+\frac{\partial \widetilde{F}}{\partial \xi}+\frac{\partial \widetilde{G}}{\partial \eta}+\frac{\partial \widetilde{H}}{\partial \zeta}=0, \tag{2.2}$$

where

$$\begin{cases} \widetilde{Q}=|J|Q \\ \widetilde{F}=|J|(Q\xi_t+F\xi_x+G\xi_y+H\xi_z) \\ \widetilde{G}=|J|(Q\eta_t+F\eta_x+G\eta_y+H\eta_z) \\ \widetilde{H}=|J|(Q\zeta_t+F\zeta_x+G\zeta_y+H\zeta_z) \end{cases}. \tag{2.3}$$

Herein, $\tau = t$, and $(\xi,\eta,\zeta)\in[-1,1]^3$, are the local coordinates in the computational domain. In the transformation shown above, the Jacobian matrix J takes the following form:

$$J=\frac{\partial(x,y,z,t)}{\partial(\xi,\eta,\zeta,\tau)}=\begin{bmatrix} x_\xi & x_\eta & x_\zeta & x_\tau \\ y_\xi & y_\eta & y_\zeta & y_\tau \\ z_\xi & z_\eta & z_\zeta & z_\tau \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.4}$$

For a non-singular transformation, its inverse transformation must also exist, and the transformation matrix is

$$J^{-1}=\frac{\partial(\xi,\eta,\zeta,\tau)}{\partial(x,y,z,t)}=\begin{bmatrix} \xi_x & \xi_y & \xi_z & \xi_t \\ \eta_x & \eta_y & \eta_z & \eta_t \\ \zeta_x & \zeta_y & \zeta_z & \zeta_t \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.5}$$

It should be noted that all the information concerning grid velocity $\vec{v}_g=(x_\tau,y_\tau,z_\tau)$ is contained in ξ_t, η_t and ζ_t, which can be written as

$$\begin{cases} \xi_t=-\vec{v}_g\cdot\nabla\xi \\ \eta_t=-\vec{v}_g\cdot\nabla\eta \\ \zeta_t=-\vec{v}_g\cdot\nabla\zeta \end{cases}. \tag{2.6}$$

2.2. Space discretization

A brief review of the SD method is given here for completeness. A more detailed description of this numerical method is available in [22]. In the SD method, two sets of points are given, namely the solution and flux points, as shown in Fig. 1b. Conservative variables are defined at the solution points, and then interpolated to flux points to obtain local fluxes. In this study the flux points are selected to be the Legendre–Gauss points plus both end points −1 and 1.

The fluxes are computed at the flux points using Lagrange interpolation polynomials. It should be pointed out that this solution polynomial is only continuous within a standard element, but discontinuous at the cell interfaces. Therefore, for the inviscid flux, a Riemann solver is necessary to compute a common flux on the interface. For a moving boundary problem, since the eigenvalues of the Euler equations are different from those for a fixed boundary problem by the grid velocity, the design of the Riemann solver should consider the grid velocity. Taking the Rusanov flux [22] as an example, the reconstructed fluxes in three directions can be written as

$$\begin{cases} \widetilde{F^i}=\frac{1}{2}[\widetilde{F^i_L}+\widetilde{F^i_R}-(|\overline{V_n}-v_{gn}|+\bar{c})\cdot(Q_R-Q_L)\cdot|J||\nabla\xi|\cdot sign(\vec{n}\cdot\nabla\xi)] \\ \widetilde{G^i}=\frac{1}{2}[\widetilde{G^i_L}+\widetilde{G^i_R}-(|\overline{V_n}-v_{gn}|+\bar{c})\cdot(Q_R-Q_L)\cdot|J||\nabla\eta|\cdot sign(\vec{n}\cdot\nabla\eta)] \\ \widetilde{H^i}=\frac{1}{2}[\widetilde{H^i_L}+\widetilde{H^i_R}-(|\overline{V_n}-v_{gn}|+\bar{c})\cdot(Q_R-Q_L)\cdot|J||\nabla\zeta|\cdot sign(\vec{n}\cdot\nabla\zeta)], \end{cases} \tag{2.7}$$

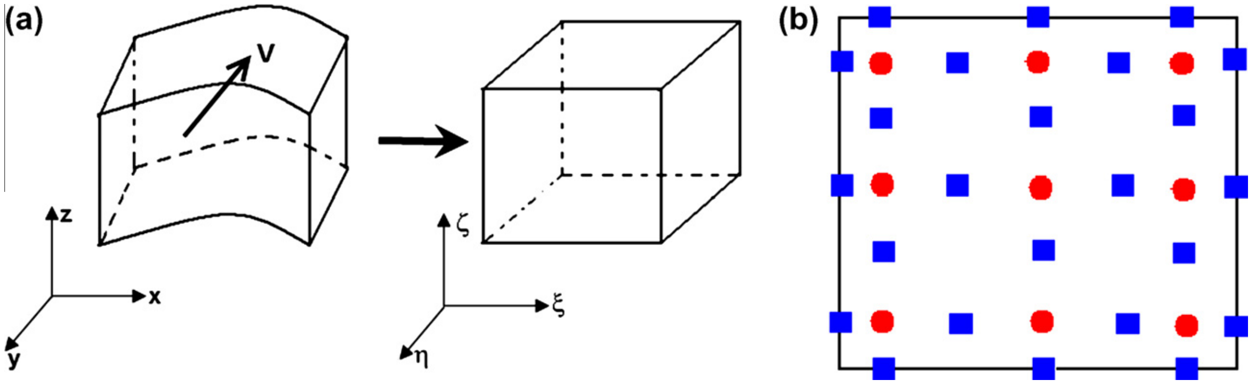

Fig. 1. (a) Transformation from a moving physical domain to a fixed computational domain. (b) Distribution of solution points (as denoted by circles) and flux points (as denoted by squares) in a standard quadrilateral element for a third-order SD scheme.

where superscript i indicates the inviscid flux, subscript n indicates the normal direction of the interface. It should be noted that $Q\xi_t$, $Q\eta_t$ and $Q\zeta_t$ are included in the inviscid fluxes. The reconstruction of the viscous flux can be found in [22].

2.3. Geometric Conservation Law (GCL)

The GCL for the metrics of the transformation from the physical domain to the computational one can be expressed as

$$\begin{cases} \frac{\partial}{\partial\xi}(|J|\xi_x)+\frac{\partial}{\partial\eta}(|J|\eta_x)+\frac{\partial}{\partial\zeta}(|J|\zeta_x)=0 \\ \frac{\partial}{\partial\xi}(|J|\xi_y)+\frac{\partial}{\partial\eta}(|J|\eta_y)+\frac{\partial}{\partial\zeta}(|J|\zeta_y)=0 \\ \frac{\partial}{\partial\xi}(|J|\xi_z)+\frac{\partial}{\partial\eta}(|J|\eta_z)+\frac{\partial}{\partial\zeta}(|J|\zeta_z)=0 \\ \frac{\partial|J|}{\partial t}+\frac{\partial}{\partial\xi}(|J|\xi_t)+\frac{\partial}{\partial\eta}(|J|\eta_t)+\frac{\partial}{\partial\zeta}(|J|\zeta_t)=0. \end{cases} \tag{2.8}$$

It is obvious that the first three formula of the GCL only depend on the accuracy of the space discretization, while the last one is related to the time evolution of the moving grid. Since the spatial metrics are computed exactly, the first three equations are automatically satisfied. If the mesh undergoes rigid-body motion without deformation, $|J|$ is independent of time. Due to the discretization error, the time-dependent GCL may not be strictly satisfied if one does not pay attention to how the mesh velocity is computed. However, for a dynamic mesh, spurious flows can be induced if the GCL is not strictly enforced. Therefore, GCL is a critical element for dynamic meshes.

In the present study, the GCL error in the numerical simulation is canceled by adding a source term to the N–S equations in the computational domain. In [17–19], the enforcement of GCL is achieved by using the same time integration form for the Jacobian as the conservative variables. An extra equation for the Jacobian needs to be solved iteratively. However, the present approach calculates the Jacobian directly, and then eliminates the errors generated by the disagreements between Jacobian and the corresponding grid velocity through a source term. Herein, treatments of the GCL are introduced separately for explicit and implicit schemes due to their different characteristics.

2.3.1. Explicit scheme

The semi-discrete form of the N–S equation in the computational domain reads

$$\frac{\partial\widetilde{Q}}{\partial t}=R(\widetilde{Q}^n)=-\left(\frac{\partial\widetilde{F}}{\partial\xi}+\frac{\partial\widetilde{G}}{\partial\eta}+\frac{\partial\widetilde{H}}{\partial\zeta}\right). \tag{2.9}$$

The equation is solved with a multi-stage strong-stability-preserving (SSP) Runge–Kutta scheme.

The following equation is obvious by the chain rule,

$$\frac{\partial\widetilde{Q}}{\partial t}=\frac{\partial|J|Q}{\partial t}=|J|\frac{\partial Q}{\partial t}+Q\frac{\partial|J|}{\partial t} \tag{2.10}$$

Substitute the last formula of Eqs. (2.8) into Eq. (2.10), we obtain

$$\frac{\partial\widetilde{Q}}{\partial t}=|J|\frac{\partial Q}{\partial t}-Q\left[\frac{\partial}{\partial\xi}(|J|\xi_t)+\frac{\partial}{\partial\eta}(|J|\eta_t)+\frac{\partial}{\partial\zeta}(|J|\zeta_t)\right] \tag{2.11}$$

Thus Eq. (2.9) is changed to the following form,

$$\begin{aligned}\frac{\partial Q}{\partial t}&=\frac{1}{|J|}\left\{-\left(\frac{\partial\widetilde{F}}{\partial\xi}+\frac{\partial\widetilde{G}}{\partial\eta}+\frac{\partial\widetilde{H}}{\partial\zeta}\right)+Q\left[\frac{\partial}{\partial\xi}(|J|\xi_t)+\frac{\partial}{\partial\eta}(|J|\eta_t)+\frac{\partial}{\partial\zeta}(|J|\zeta_t)\right]\right\}\\&=\frac{1}{|J|}\left\{-\left(\frac{\partial\widetilde{F}}{\partial\xi}+\frac{\partial\widetilde{G}}{\partial\eta}+\frac{\partial\widetilde{H}}{\partial\zeta}\right)+source\right\}\end{aligned} \tag{2.12}$$

where

$$source=Q\left[\frac{\partial}{\partial\xi}(|J|\xi_t)+\frac{\partial}{\partial\eta}(|J|\eta_t)+\frac{\partial}{\partial\zeta}(|J|\zeta_t)\right]. \tag{2.13}$$

Note that $\frac{\partial\widetilde{F}}{\partial\xi}+\frac{\partial\widetilde{G}}{\partial\eta}+\frac{\partial\widetilde{H}}{\partial\zeta}$ contains a term as $Q\left[\frac{\partial}{\partial\xi}(|J|\xi_t)+\frac{\partial}{\partial\eta}(|J|\eta_t)+\frac{\partial}{\partial\zeta}(|J|\zeta_t)\right]$. It is clear that GCL is satisfied strictly as this term will be canceled by the 'source' term when Q is a constant (i.e. the free stream flow). The benefits of this method are that the source term is easy to compute and implement for the original solver for stationary grids and the calculation of $\partial|J|/\partial t$ can be avoided, which might generate additional errors and increase the computational cost.

2.3.2. Implicit scheme

At each cell 'c', using the backward Euler scheme for the time derivative,

$$\frac{\widetilde{Q_c}^{n+1}-\widetilde{Q_c}^n}{\Delta t}-\left[R_c(\widetilde{Q}^{n+1})-R_c(\widetilde{Q}^n)\right]=R_c(\widetilde{Q}^n), \tag{2.14}$$

further performing the Taylor expansion and keeping the first-order term, we obtain

$$R_c(\widetilde{Q}^{n+1})-R_c(\widetilde{Q}^n)=\frac{\partial R_c}{\partial\widetilde{Q_c}}\Delta\widetilde{Q_c}+\sum_{nb\neq c}\frac{\partial R_c}{\partial\widetilde{Q_{nb}}}\Delta\widetilde{Q_{nb}}, \tag{2.15}$$

where $\Delta\widetilde{Q_c}=\widetilde{Q_c}^{n+1}-\widetilde{Q_c}^n$, 'nb' indicates all the neighboring cells contributing to the residual of cell'c'.

Combining (2.14) and (2.15), we obtain

$$\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial\widetilde{Q_c}}\right)\Delta\widetilde{Q_c}-\sum_{nb\neq c}\frac{\partial R_c}{\partial\widetilde{Q_{nb}}}\Delta\widetilde{Q_{nb}}=R_c(\widetilde{Q}^n). \tag{2.16}$$

However, it is expensive in memory to store the full LHS implicit Jacobian matrices. Therefore, a preconditioned LU-SGS scheme is adopted in the development of the implicit scheme. Herein, we just introduce a preconditioning matrix as

$$D=\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial\widetilde{Q_c}}\right), \tag{2.17}$$

and the iterative scheme becomes

$$\begin{aligned}D\Delta\widetilde{Q_c}^{(k+1)}&=\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial\widetilde{Q_c}}\right)\Delta\widetilde{Q_c}^{(k+1)}\\&=R_c(\widetilde{Q}^n)+\sum_{nb\neq c}\frac{\partial R_c}{\partial\widetilde{Q_{nb}}}\Delta\widetilde{Q_{nb}}^*,\end{aligned} \tag{2.18}$$

where superscript $(k+1)$ is an iterative index, and $*$ indicates the most recently updated solutions. It should be noted that $\Delta\widetilde{Q_c}^{(k+1)}$ can be written as

$$\begin{aligned}\Delta\widetilde{Q_c}^{(k+1)}&=\widetilde{Q_c}^{(k+1)}-\widetilde{Q_c}^n\\&=\left(\widetilde{Q_c}^{(k+1)}-\widetilde{Q_c}^{(k)}\right)\\&\quad+\left(\widetilde{Q_c}^{(k)}-\widetilde{Q_c}^n\right),\quad\text{with}\quad\widetilde{Q_c}^{(k)}=\widetilde{Q_c}^*.\end{aligned} \tag{2.19}$$

Since we do not want to store the matrices $\partial R_c/\partial\widetilde{Q_{nb}}$, (2.18) is further manipulated as follows:

$$\begin{aligned}R_c(\widetilde{Q}^n)+\sum_{nb\neq c}\frac{\partial R_c}{\partial\widetilde{Q_{nb}}}\Delta\widetilde{Q_{nb}}^*&=R_c\left(\widetilde{Q}_c^n,\{\widetilde{Q}_{nb}^n\}_{nb\neq c}\right)+\sum_{nb\neq c}\frac{\partial R_c}{\partial\widetilde{Q_{nb}}}\Delta\widetilde{Q_{nb}}^*\\&\approx R_c(\widetilde{Q}_c^n,\{\widetilde{Q}_{nb}^*\}_{nb\neq c})\\&\approx R_c(\widetilde{Q}_c^*,\{\widetilde{Q}_{nb}^*\}_{nb\neq c})-\frac{\partial R_c}{\partial\widetilde{Q_c}}\Delta\widetilde{Q_c}^*\\&=R_c(\widetilde{Q}^*)-\frac{\partial R_c}{\partial\widetilde{Q_c}}\Delta\widetilde{Q_c}^*\left(\text{or }R_c(\widetilde{Q}^*)-\frac{\partial R_c}{\partial\widetilde{Q_c}}\Delta\widetilde{Q_c}^{(k)}\right)\end{aligned} \tag{2.20}$$

In (2.20), note that both approximations can be obtained using the first-order Taylor series expansion. Combining (2.18)–(2.20), we obtain

$$D\left(\widetilde{Q_c}^{(k+1)} - \widetilde{Q_c}^{(k)}\right) = \left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \widetilde{Q_c}}\right)\left(\widetilde{Q_c}^{(k+1)} - \widetilde{Q_c}^{(k)}\right) = R_c\left(\widetilde{Q}^*\right) - \frac{\Delta \widetilde{Q_c}^*}{\Delta t}, \tag{2.21}$$

Since matrix D merely serves as a preconditioner, the accuracy of the iteration will be determined by the right-hand side (RHS) of the Eq. (2.21).

Note that

$$\frac{\partial \widetilde{F}}{\partial \xi} = \frac{\partial\left[|J|(Q\xi_t + F\xi_x + G\xi_y + H\xi_z)\right]}{\partial \xi} = Q\frac{\partial |J|\xi_t}{\partial \xi} + |J|\xi_t\frac{\partial Q}{\partial \xi} + \frac{\partial\left[|J|(F\xi_x + G\xi_y + H\xi_z)\right]}{\partial \xi}, \tag{2.22}$$

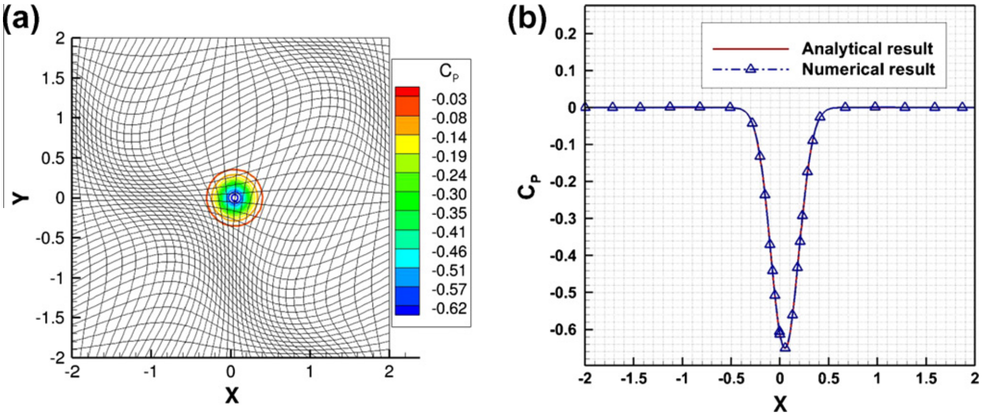

Fig. 2. (a) Pressure coefficient distribution and grid deformation; (b) comparison between numerical and analytical solutions of pressure coefficient along $y = 0$ at $t = 0.1$. The solid line denotes the analytical result, and the dash–dot line with triangles indicates the numerical result.

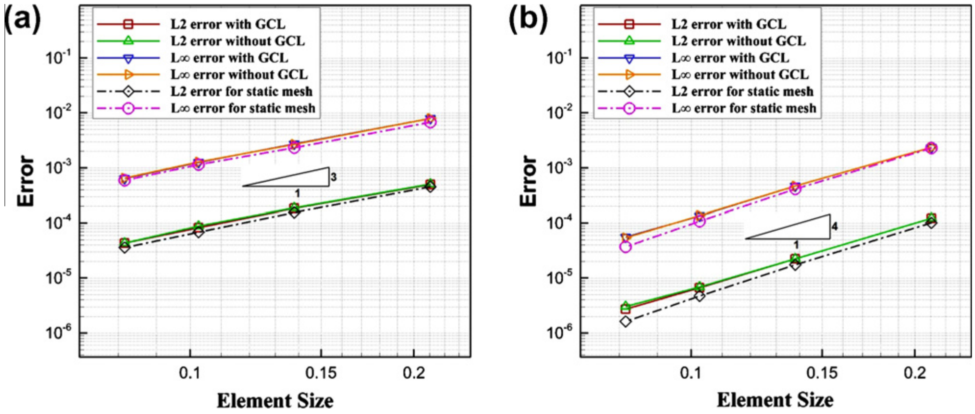

Fig. 3. The convergence of the vortex propagation problem using the deformable grid with and without GCL correction, as well as for the stationary grid. Figure (a) and (b) displays results from the third-order and fourth-order SD methods respectively. In both cases, four mesh sizes are used and error representations in both 2-norm (as denoted by L_2) and infinity-norm (as denoted by L_∞) are given.

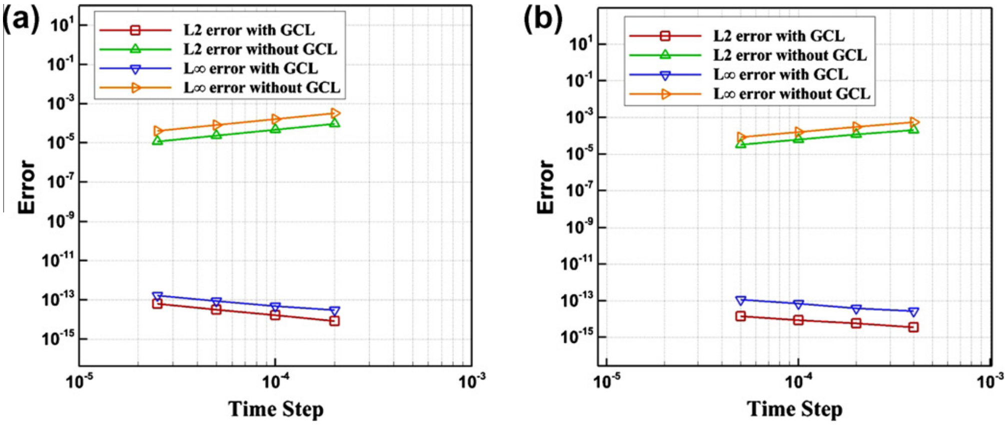

Fig. 4. The convergence of the free stream preservation test using the deformable grid with and without GCL correction. Results from the fourth-order SD method with (a) explicit SSP-RKS and (b) implicit BDF2 time integration schemes are displayed. In both cases, four time steps are used and error representations in both 2-norm (as denoted by L_2) and infinity-norm (as denoted by L_∞) are given.

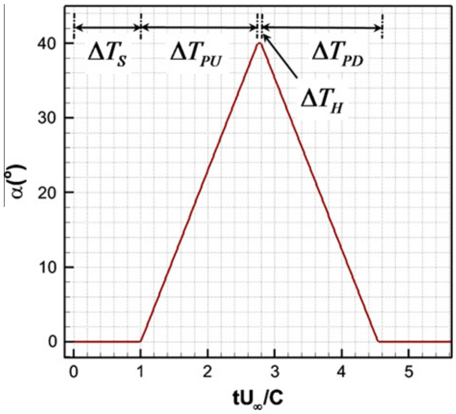

Fig. 5. Pitching angle evolution during the hold-pitch-up-hold-pitch down process.

which is contained in $R(\widetilde{Q})$.

Thus, the GCL is introduced in the RHS as follows.

$$\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial\widetilde{Q_c}}\right)(\widetilde{Q_c}^{(k+1)}-\widetilde{Q_c}^{(k)})$$
$$=R_c(\widetilde{Q}^*)-\frac{\Delta\widetilde{Q_c}^*}{\Delta t}+Q_c^*\left(\frac{\Delta|J|^*}{\Delta t}+\frac{\partial}{\partial\xi}(|J|^*\xi_t^*)+\frac{\partial}{\partial\eta}(|J|^*\eta_t^*)+\frac{\partial}{\partial\zeta}(|J|^*\zeta_t^*)\right) \tag{2.23}$$

It should be noted that in the above equation the discrete form of $\Delta|J|^*/\Delta t$ is exactly the same as $\Delta\widetilde{Q_c}^*/\Delta t$. This consistency can help minimize the errors induced by discretization schemes. For example, the second-order backward difference scheme (BDF2) for the two derivatives can be written as below,

$$\frac{\Delta\widetilde{Q_c}^*}{\Delta t}=\frac{3\widetilde{Q_c}^*-4\widetilde{Q_c}^n+\widetilde{Q_c}^{n-1}}{2\Delta t},\quad \frac{\Delta|J|^*}{\Delta t}=\frac{3|J|^*-4|J|^n+|J|^{n-1}}{2\Delta t} \tag{2.24}$$

2.4. General grid deformation strategies

In order to solve problems with moving grids, it is necessary to design a grid moving algorithm. As the first step, the boundary motion of the physical domain is specified according to the physical problem. Then traditionally two methods can be used to manipulate the rest of the mesh nodes. The first one is to use the algebraic procedure to smooth the whole field [5,17–19,30]. Another approach is to solve differential equations (usually elliptic, like equations of linear elasticity) with the specified boundary conditions [21,30]. For the sake of computational efficiency, an algebraic methodology is performed in the present study, which has been widely used by other researchers [17–19].

The first implementation of the algebraic method is to make the whole physical domain perform a rigid-body motion. Obviously, this approach cannot handle relative motions among several

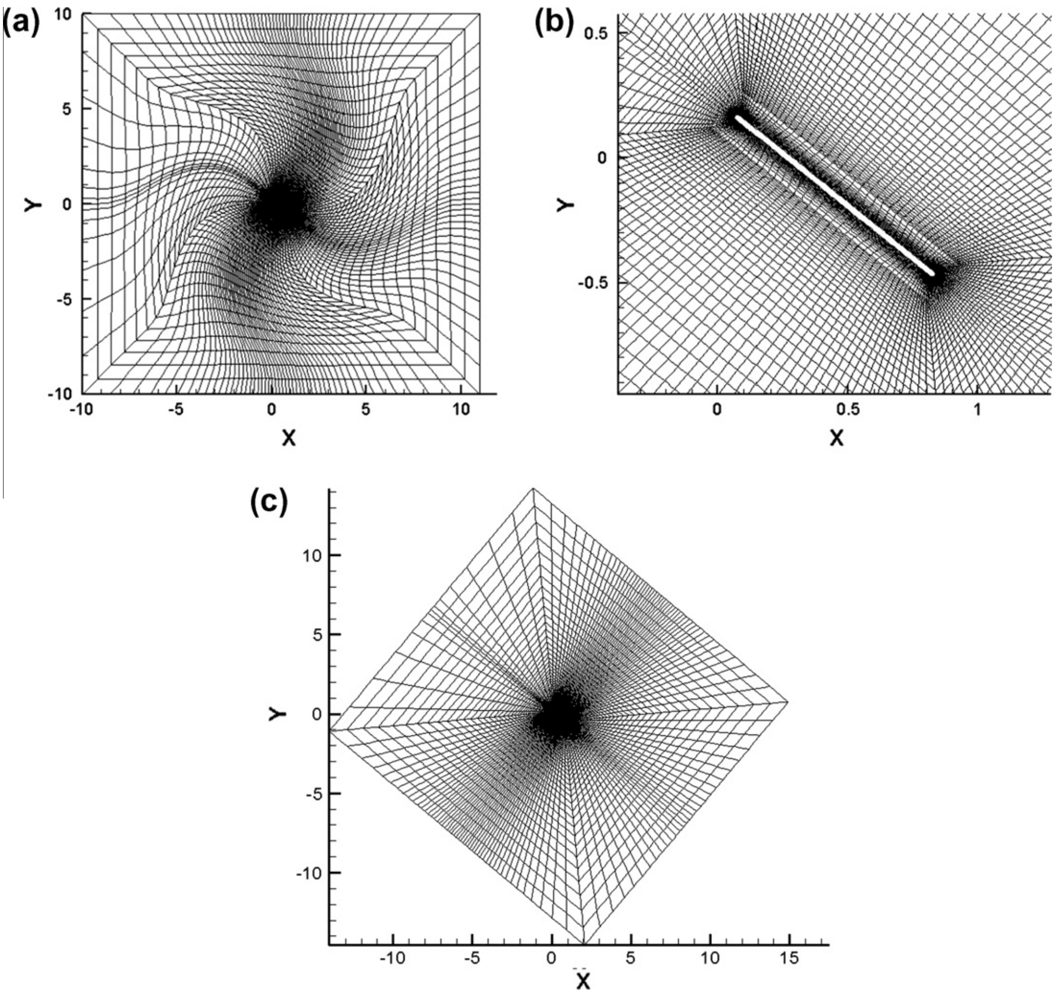

Fig. 6. (a) Overview of the deformable grid; (b) close-up view of the deformable grid near the moving boundary; (c) overview of the rigidly moving grid.

components. Another implementation is to use blending functions to reconstruct the whole physical domain. In the present study, a fifth-order polynomial blending function proposed in [19],

$$r_5(s) = 10s^3 - 15s^4 + 6s^5, s \in [0,1] \tag{2.25}$$

is adopted. It is obvious that $r_5'(0) = 0, r_5'(1) = 0$, which can generate a smooth variation at both end points during the mesh reconstruction. Herein, 's' is a normalized arc length, which reflects the 'distance' between the present node and the moving boundaries. Specifically, $s = 0$ means that the present node will move with the moving boundary, while $s = 1$ means that the present node will not move. Therefore, for any motion (transition, rotation), the change of the position vector $\vec{P}$ is

$$\Delta\vec{P}_{present} = (1 - r_5)\Delta\vec{P}_{rigid}. \tag{2.26}$$

After these manipulations, a new set of mesh nodes can be calculated based on $\Delta\vec{P}$. In the present study, for the deformable grid approach, in order to maintain the grid quality near the wall boundaries, rigid motions are enforced in the vicinity of the wall boundaries. The outer boundaries far from the wall are specified as stationary reference. Between the rigidly displaced grid and the stationary grid, the blending function (2.25) is used to interpolate and smooth the grid motion.

It should be mentioned that the same smoothing method can also be used in problems with two or more objects with relative motions. In the present study, a tandem airfoil problem is investigated using this approach, as will be discussed in the next section. In that case, the change of the position vector $\vec{P}$ can be written as

$$\begin{aligned}
&\Delta\vec{P}_{present} = \Delta\vec{P}_{rigid1}, && \textit{if } s_1 = 0\\
&\Delta\vec{P}_{present} = \Delta\vec{P}_{rigid2}, && \textit{if } s_2 = 0\\
&\Delta\vec{P}_{present} = \tfrac{s_2^n}{s_1^n + s_2^n}[1 - r_5(s_1)]\Delta\vec{P}_{rigid1} + \tfrac{s_1^n}{s_1^n + s_2^n}[1 - r_5(s_2)]\Delta\vec{P}_{rigid2}, && \textit{otherwise}
\end{aligned} \tag{2.27}$$

and it is made sure that there is no region with both $s_1 = 0$ and $s_2 = 0$.

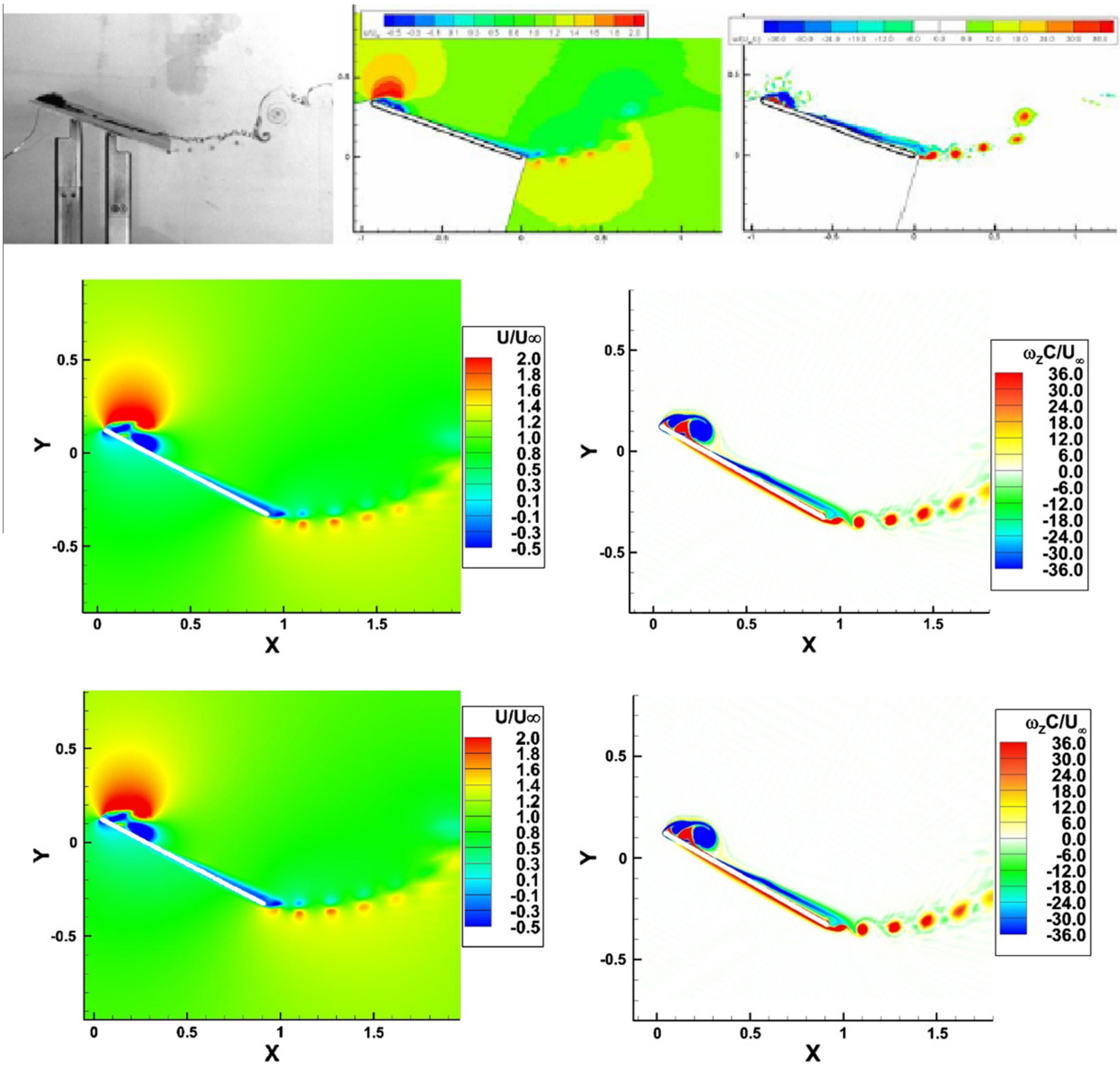

Fig. 7. Comparison between numerical and experimental results for Re = 10,000, $k = 0.2$, $\alpha = 20°$ when $tU_\infty/C = 1.8725$. (a) Experimental results (courtesy of OL [15]). From left to right: flow visualization with dye; u velocity contour (PIV); vorticity contour in the spanwise direction (PIV). (b) Numerical results with deformable grids. Left: u velocity contour (u/U_∞); right: vorticity contour in the spanwise direction. (c) Numerical results with rigidly moving grids. Left: u velocity contour; right: vorticity contour in the spanwise direction.

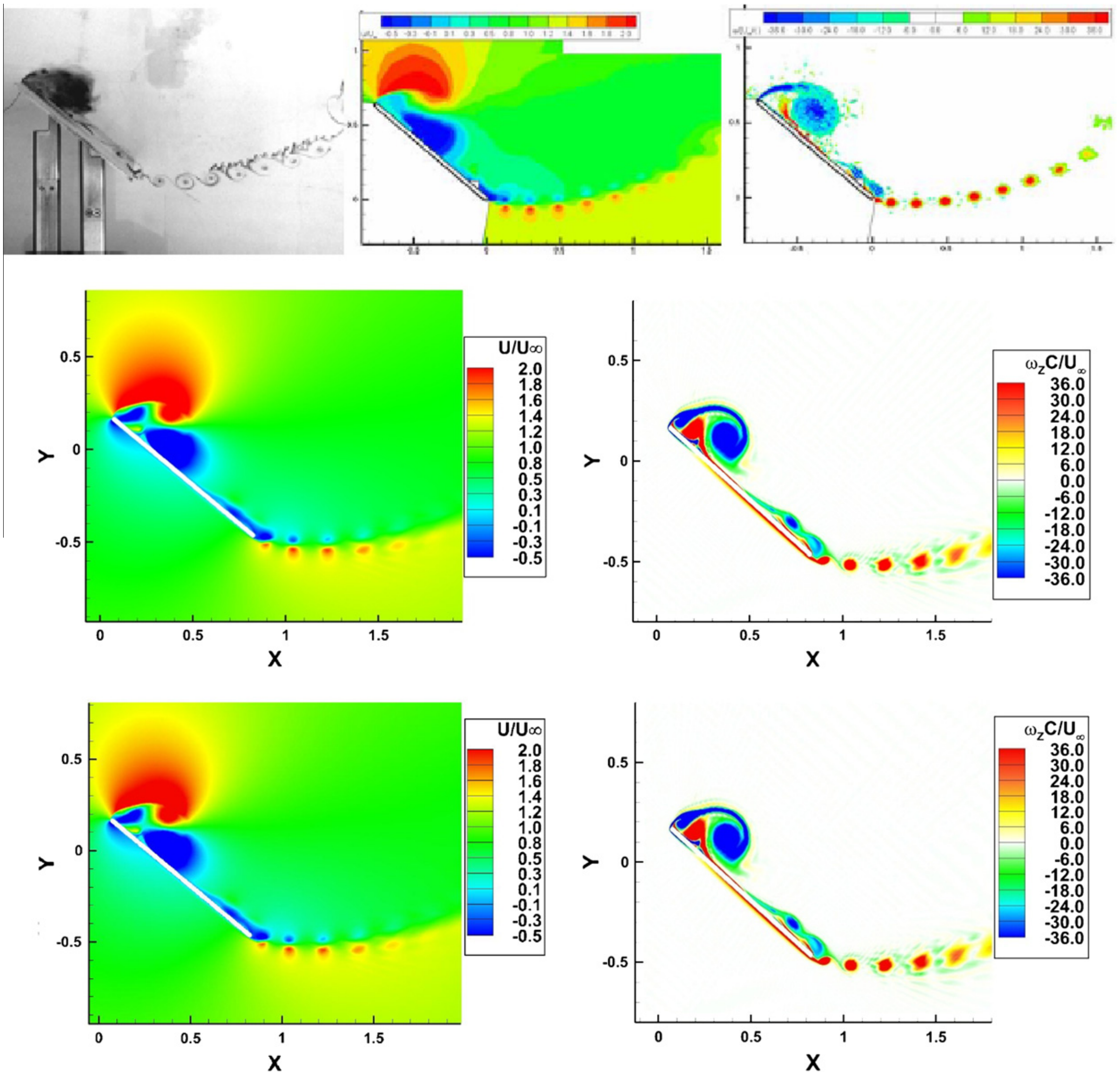

Fig. 8. Comparison between numerical and experimental results for Re = 10000, k = 0.2, α = 40° when tU_∞/C = 2.745. (a) Experimental results (courtesy of OL [15]). From left to right: flow visualization with dye; u velocity contour (PIV); vorticity contour in the spanwise direction (PIV). (b) Numerical results with deformable grids. Left: u velocity contour (u/U_∞); right: vorticity contour in the spanwise direction. (c) Numerical results with rigidly moving grids. Left: u velocity contour; right: vorticity contour in the spanwise direction.

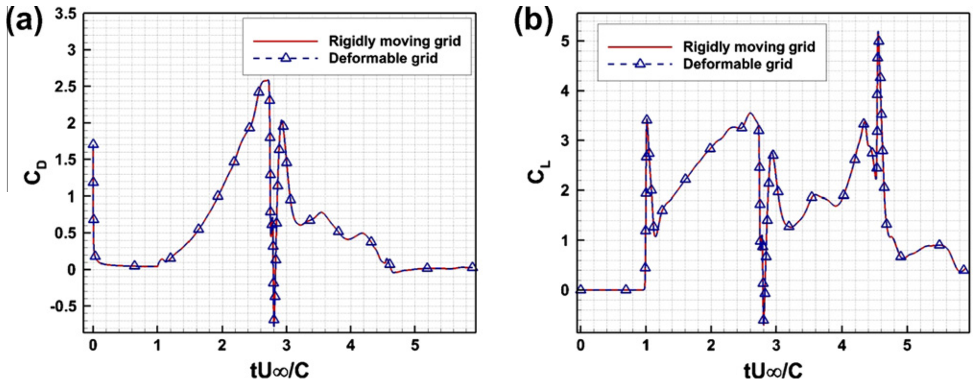

Fig. 9. (a) Drag coefficient history and (b) lift coefficient history for Re = 10,000, k = 0.2, calculated using both the rigidly moving grid (as denoted by the solid line) and the deformable grid (as denoted by the dash–dot line with triangles).

Another point is that if ‘s’ in the blending function (2.25) is set to be 0 at any grid point, then a rigidly moving grid approach is achieved. In this case, the whole domain will have the same motion. Generally speaking, a rigid grid is only suitable for simple motions of one object. For the case of multiple objects with relative motions, it will generate overset cells. From a numerical perspective, the Jacobian of the transformation from the physical domain to the computational domain will be the same all the time, and theoretically this will introduce less error when performing simulations, as Jacobian needs to be calculated only once. On the other hand, a deformable grid is desirable in more general cases. But extra efforts are needed to calculate the changing Jacobian as the grid evolves.

It is clear that for systems with complex relative motions, the algebraic algorithm for the grid motion can be hard to design. However, for many cases this method enjoys its remarkable simplicity and efficiency. Several examples will be shown in the next section.

3. Numerical results

3.1. Accuracy study using an isentropic vortex propagating problem

In order to verify that the SD method can preserve its high-order accuracy for deformable meshes, a 2D Euler vortex propagation case is performed in the present study. SSP third-order RungeKutta (SSP-RK3) time integration is used for this study. The definition of the isentropic vortex and its evolution process can be described as [7]

$$u(r) = \frac{U'_{max}}{b} r e^{\frac{1}{2}\left(1-\frac{r^2}{b^2}\right)}, \quad \rho(r) = \left(1 - \frac{1}{2}(\gamma-1)U'^2_{max} e^{1-\frac{r^2}{b^2}}\right)^{1/(\gamma-1)}, p(r) = \left(1 - \frac{1}{2}(\gamma-1)U'^2_{max} e^{1-\frac{r^2}{b^2}}\right)^{\gamma/(\gamma-1)},$$

and

$$\begin{pmatrix} \rho(x,y,t) \\ u(x,y,t) \\ v(x,y,t) \\ p(x,y,t) \end{pmatrix} = \begin{pmatrix} 0 \\ U_0 \\ V_0 \\ 0 \end{pmatrix} + \begin{pmatrix} \rho(r) \\ -u(r)\sin\theta \\ u(r)\cos\theta \\ p(r) \end{pmatrix},$$

where $u(r)$, $\rho(r)$, $p(r)$ are the velocity, density and pressure distribution of the vortex respectively; U_0 and V_0 are the advection velocities of the main stream in the x- and y-directions; $r = \sqrt{(x-U_0t)^2 + (y-V_0t)^2}$, is the radial distance from the vortex center; b is a constant.

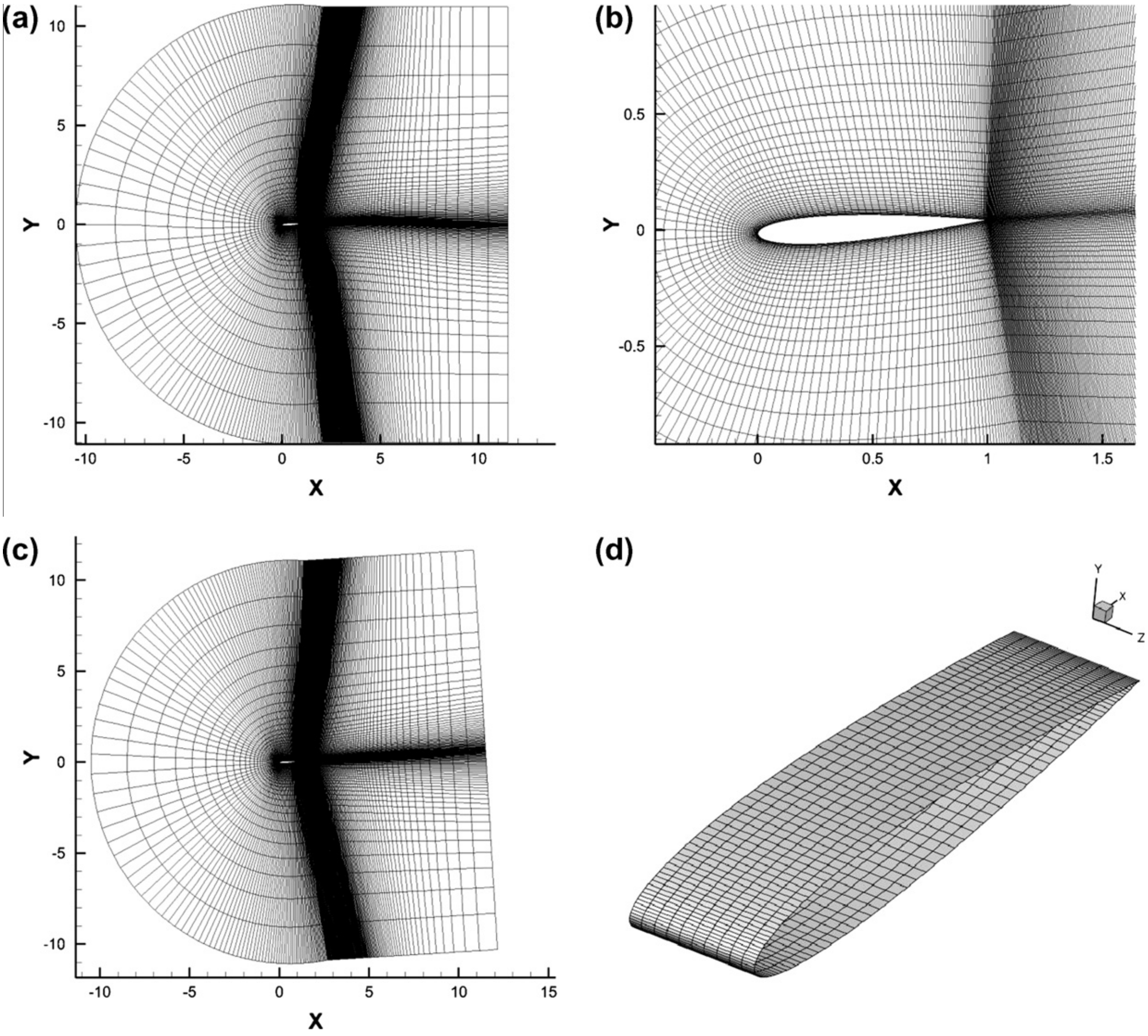

Fig. 10. Grids used for the simulations of the sinusoidally pitching airfoil. (a) Overview of the deformable grid; (b) close-up view of the deformable grid near the moving boundary; (c) overview of the rigidly moving grid; (d) airfoil surface grid for the 3D simulations.

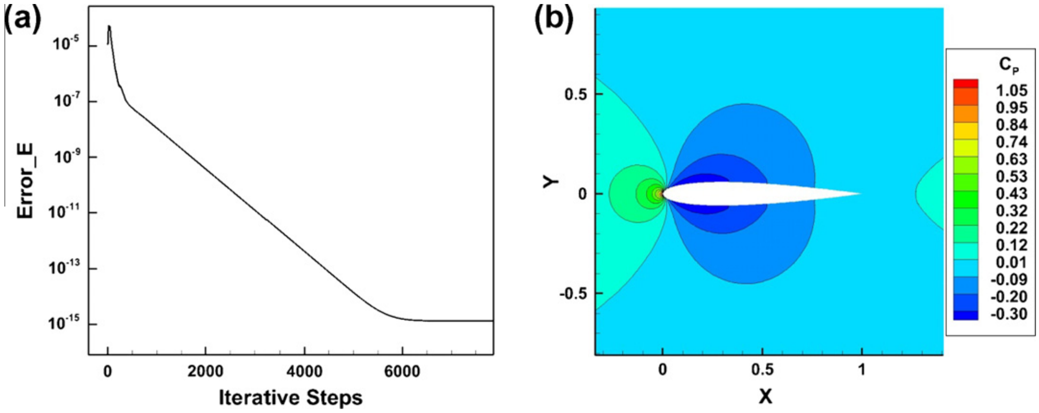

Fig. 11. (a) Convergence history of the energy error for the steady solution of the flow over a stationary NACA0012 airfoil with implicit (LU-SGS) time integration; (b) pressure coefficient contours for the converged steady flow.

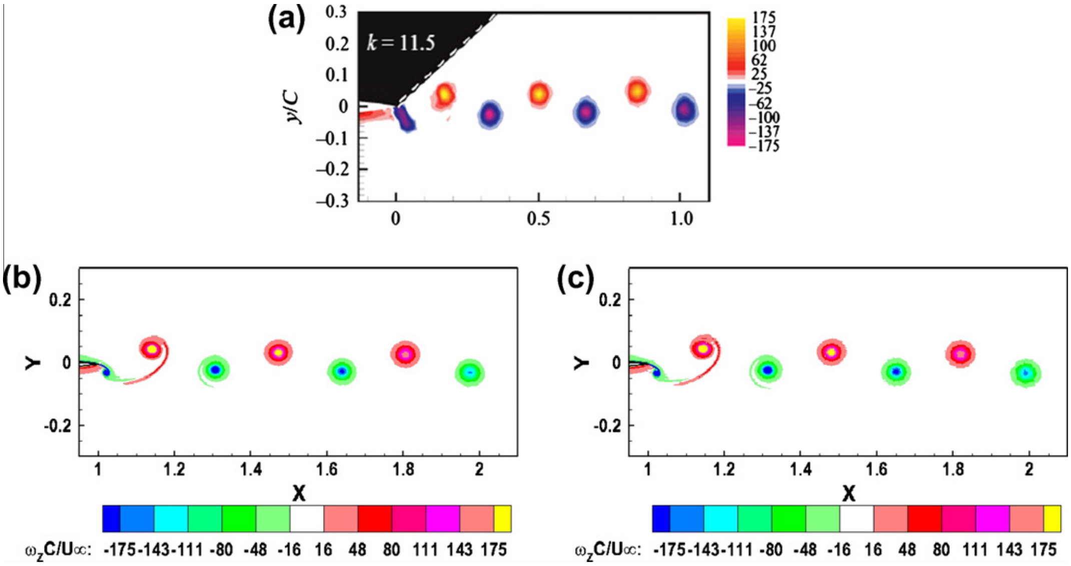

Fig. 12. Vorticity field for Re = 12600, k = 11.5, S_t = 0.19. (a) Phase-averaged experimental results (courtesy of Bohl and Koochesfahani [2]). (b) Instantaneous numerical results with deformable grid. (c) Instantaneous numerical results with rigidly moving grid.

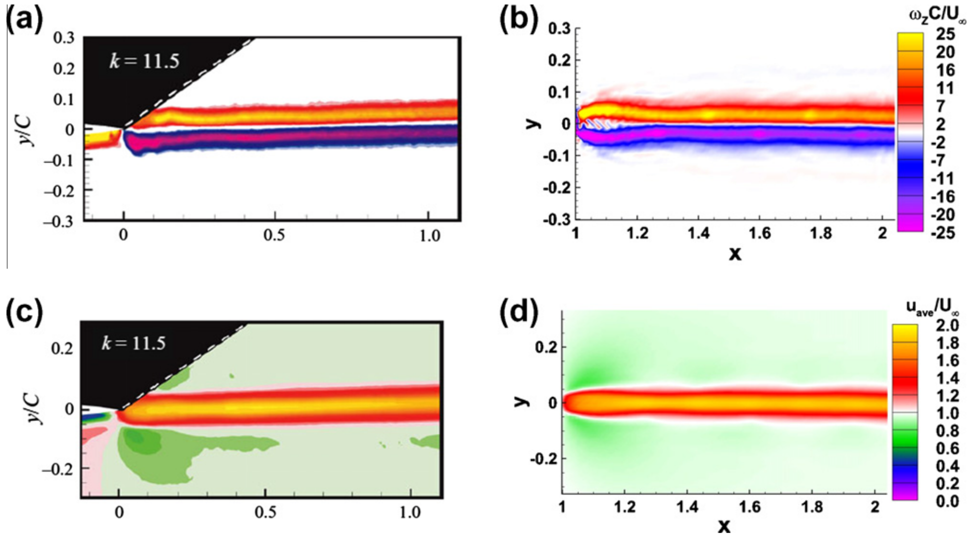

Fig. 13. Averaged flow fields for Re = 12600, k = 11.5, S_t = 0.19. (a) Vorticity field, experimental results, (courtesy of Bohl and Koochesfahani [2]); (b) vorticity field, numerical results; (c) u velocity field, experimental results, (courtesy of Bohl and Koochesfahani [2]); (d) u velocity field, numerical results.

The isentropic vortex was originally centered at*(0,0)*, with the initial condition given by $(U_0, V_0) = (0.5{,}0)$, $U'_{\max} = 0.5U_0$, $b = 0.2$. The physical domain of this problem is set to be $[-2{,}2] \times [-2{,}2]$ with one cell in the z-direction. The grid deformation strategy follows [13], which analytically defines the grid motion as

$$\vec{x}(t) = \vec{x}(t) + d\vec{x}(t)$$

with

$$dx(t) = A_x L_x dt/t_{max} \sin(f_n t) \sin(f_x x) \sin(f_y y)$$
$$dy(t) = A_y L_y dt/t_{max} \sin(f_n t) \sin(f_x x) \sin(f_y y)$$

for 2D problems. Herein, $A_{x,y}$ is the amplitude in x and y directions; $L_{x,y}$ and $t_{\max}$ depict the reference length and time; dt is the time step, and

$$f_n = n_t \pi / t_{max}, \quad f_x = n_x \pi / L_x, \quad f_y = n_y \pi / L_y.$$

The motion control parameters of the deformable grid are set as $L_x = L_y = 4$, $t_{\max} = 0.1$, and $n_x = n_y = 2$, $n_t = 1$, and $A_x = A_y = 0.2$. Since at $t = 0.1$ the grid has the largest deformation, the errors are analyzed at that instance. In order to ensure that the time integration errors have no effects on the accuracy analyses, a fixed time step is chosen as $\Delta t = 5 \times 10^{-5}$. Pressure coefficient (defined as $C_p = (p - p_\infty)/(0.5\rho U_\infty^2)$) distribution of the vortex is displayed in Fig. 2a at $t = 0.1$. From Fig. 2b, it is obvious that the analytical result agrees well with the numerical one. Results of the grid refinement study are displayed in Fig. 3, which demonstrate the accuracy of the SD method for the deformable domain. The errors are measured with both L_2 and L_∞norms, and an optimal convergence has been achieved in all cases. It is also found that the schemes with and without GCL for the isentropic vortex propagation tests almost obtain the same error values and accuracy. However, for the free stream preservation test, it is obvious from Fig. 4 that for both explicit (SSP-RK3) and implicit (BDF2) schemes, if the GCL is not enforced, the error level can reachup to nine-orders larger than machine zero. But with a GCL compliant scheme, machine zero can be achieved. In this test, the fourth-order scheme is used on the grid with 19 × 19 × 1 cells, and the errors are computed at $t = 0.1$ as well.

3.2. Bio-inspired flow simulations

Recently, there is a growing interest in the study of bio-inspired flows in the fluid dynamics community. One of the major objectives is to investigate the wake structures after flapping airfoils

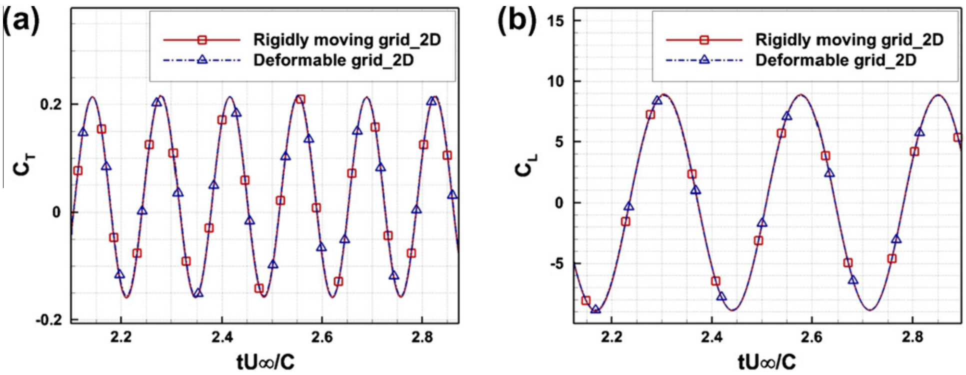

Fig. 14. (a) Thrust coefficient history and (b) lift coefficient history for Re = 12600, $k = 11.5$, $S_t = 0.19$, calculated using both the rigidly moving grid (as denoted by the solid line with squares) and the deformable grid (as denoted by the dash–dot line with triangles).

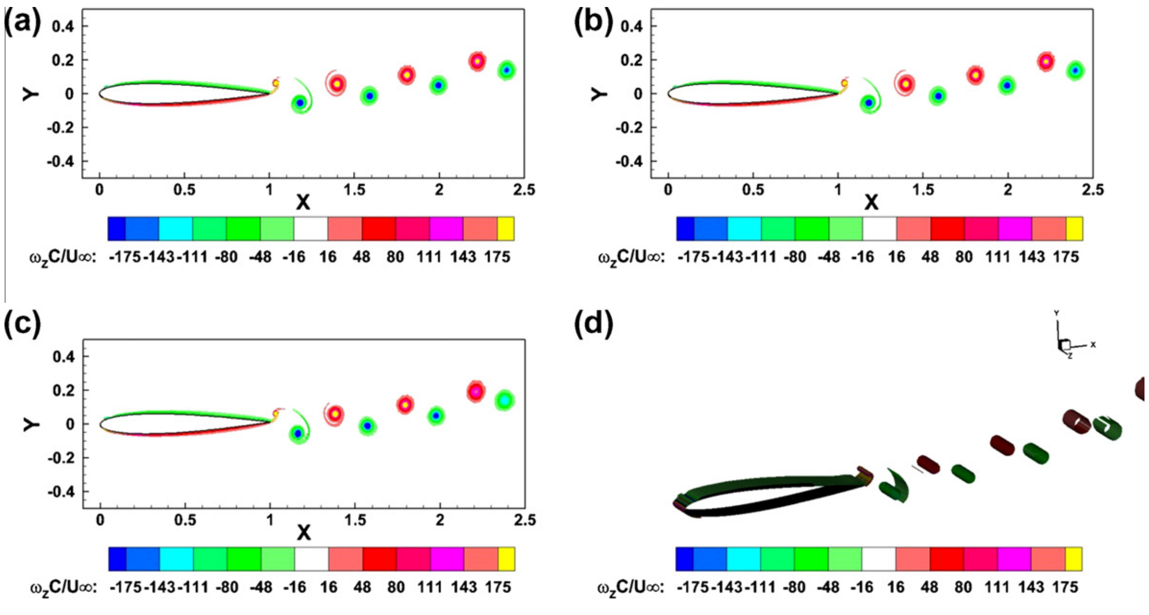

Fig. 15. Instantaneous spanwise vorticity field for Re = 12,600, $k = 11.5$, $S_t = 0.33$. (a) 2D simulation with the deformable grid; (b) 2D simulation with the rigidly moving grid; (c) 3D simulation with the rigidly moving grid; (d) iso-surface of Q colored by the spanwise vorticity from the 3D simulation results.

or wings [1,2,6,10,15,16,20,28,29,35]. The reason is that based on the evolution of these wake structures, the thrust and lift generation mechanism in agile flight can be clearly revealed. As mentioned before, such flows are unsteady vortex-dominated flows. In order to resolve the subtle vortex structures, a high-order method is necessary, as first- and second-order flow solvers may dissipate the unsteady vortices quickly. Moreover, these problems all involve moving boundaries. Therefore, several numerical simulations of the flapping-related motions are carried out to examine the performance of the high-order SD method for deformable meshes. Unless otherwise noted, the default numerical scheme used in the simulations is the third-order SD scheme. For the two dimensional simulations, the implicit BDF2 time integration is used; and for the three dimensional simulations, the explicit SSP-RK3 time integration is employed. For all the simulations presented in this section, the free stream Mach number is chosen as 0.1.

3.2.1. Flat plate pitch-up process

A series of canonical unsteady experimental studies on the flat plate pitch-up problem was conducted in [15,16]. This problem is also studied using the high-order SD method. The aim of the study

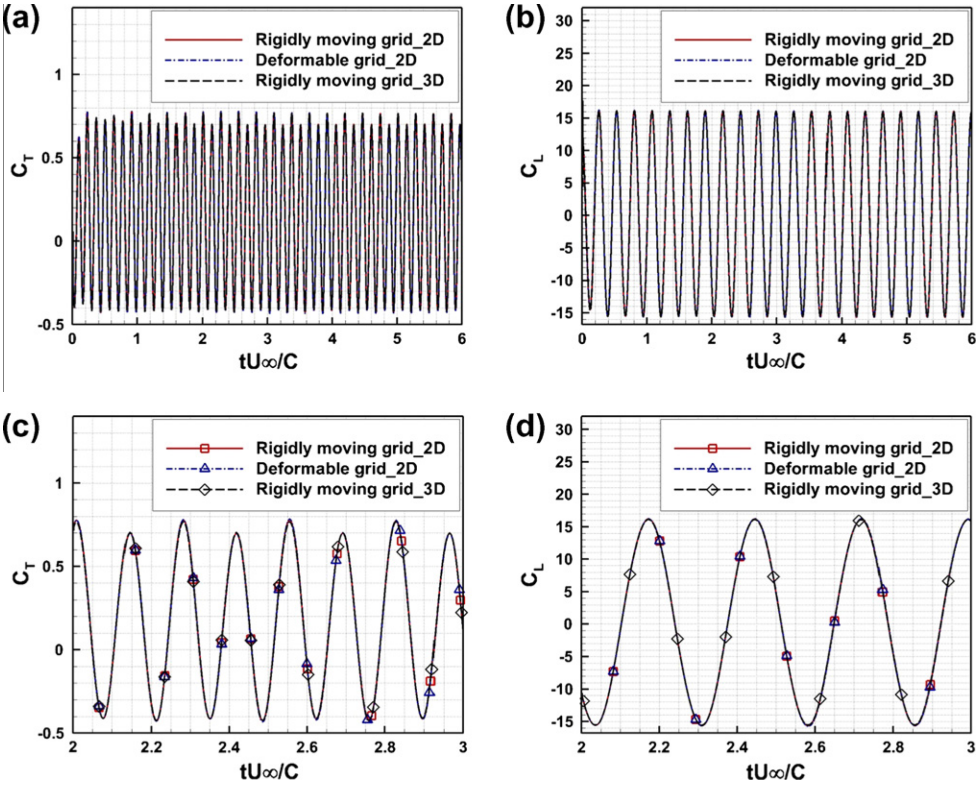

Fig. 16. (a) Thrust coefficient convergence history and (b) lift coefficient convergence history for Re = 12,600, k = 11.5, S_t = 0.33, for 2D simulations using the rigidly moving grid (as denoted by the solid line with squares) and the deformable grid (as denoted by the dash–dot line with triangles) and 3D simulations using rigidly moving grid (as denoted by the dash line with diamonds). (c) and (d) are the corresponding close-up views of (a) and (b).

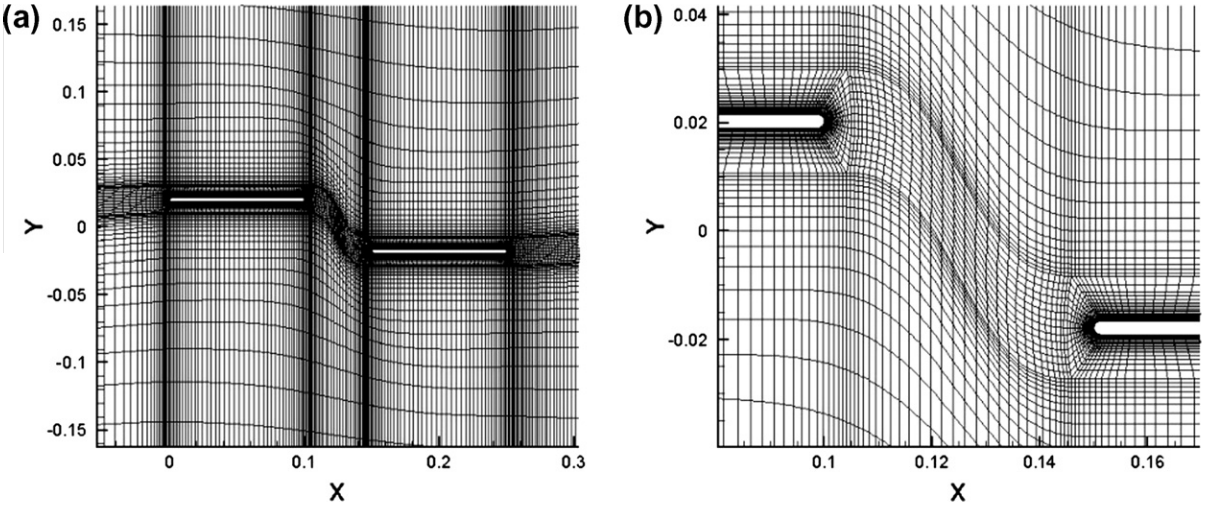

Fig. 17. (a) Overview of the deformable grid; (b) close-up view of the deformable grid between the two moving boundaries.

is to investigate the aerodynamic responses of maneuvering flights, such as perching. The main features of these problems can be generalized as high-frequency and high-amplitude pitching processes, which can be used to verify the efficiency of the SD method for deformable meshes. In order to compare the numerical results with the experimental ones, the functions and parameters used in the present study are defined to be consistent with the experiment.

The maximum pitching angle α_m is set to be 40°, and α is computed according to

$$\alpha(T) = \alpha_m \frac{G(T)}{Max(G(T))},$$

with a smoothing function defined in [6] as

$$G(T) = \ln\left[\frac{\cosh(a(T-T_1))\cosh(a(T-T_4))}{\cosh(a(T-T_2))\cosh(a(T-T_3))}\right],$$

where a is a function shape parameter, which is set to be 11.0, $T_1 = \Delta T_s$, $T_2 = T_1 + \Delta T_{pu}$, $T_3 = T_2 + \Delta T_h$ and $T_4 = T_3 + \Delta T_{pd}$ as shown in Fig. 5. Herein, T is a non-dimensional time with respect toC/U_∞, where 'C' stands for the chord length. The start-up interval ΔT_s is set to be 1.0, the reduced pitch rate $K = (C\alpha_m/\Delta T_{pu,d})/2U_\infty$ is specified as 0.2, and the hold interval ΔT_h is set to be 0.05. The Reynolds number based on the plate chord length is 10,000. The non-dimensional time step used for the simulations is $\Delta tU_\infty/C = 7.5 \times 10^{-5}$.

Fig. 6 shows the details of the deformable grid and the rigidly displaced grid. The grid has 77 × 78 × 1 cells, and the minimum cell size normalized by the plate chord length in the transverse direction is 0.0015. The numerical results for two instances during the pitch-up process, namely $tU_\infty/C = 1.8725$ (corresponding pitch angle 20°) and $tU_\infty/C = 2.745$ (corresponding pitch angle 40°), are compared with the experimental results. From Figs. 7 and 8, it is obvious that the computed instantaneous vorticity and velocity fields agree well with the experimental data. The corresponding force histories for both deformable and rigidly moving grids are displayed in Fig. 9. Note that the results with different grid deformation algorithms are nearly identical.

3.2.2. Flow over a sinusoidally pitching airfoil

An experimental investigation of the flow over a NACA-0012 airfoil performing a pitching motion with small amplitude and high reduced frequency has been conducted in [2]. The aim of the study is to find the critical point at which the von Karman vortex street turns into a reverse von Karman street and to study the parameter dependencies of the thrust generation during the pitching motion. Following this experimental study, a numerical research is completed with the same parameter setting. And some cases are verified both with rigidly moving and deformable grid strategies.

In the present study, the airfoil performs a pitching motion expressed as

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t + \phi), \quad \omega = 2\pi f$$

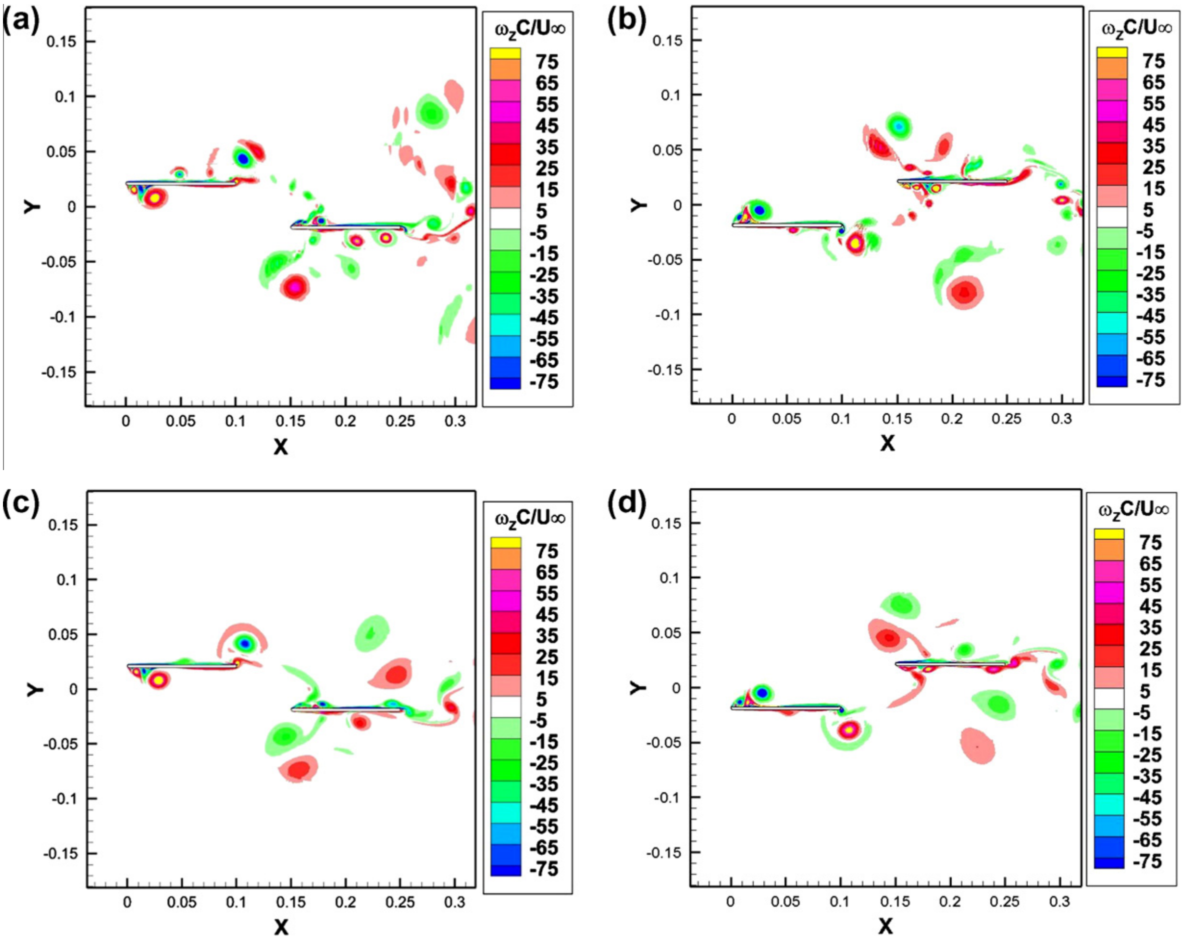

Fig. 18. Instantaneous vorticity fields of a tandem airfoil configuration. (a) and (c) display the vorticity fields calculated at the phase of the fore plate up and hind plate down position using the third-order and second-order accuracy schemes respectively; (b) and (d) display the vorticity fields calculated at the phase of the fore plate down and hind plate up position using the third-order and second-order accuracy schemes respectively.

where α_m is the mean angle of attack, α_0 is the amplitude of the pitching angle, ϕ is the initial phase. Also, the reduced frequency k and the Strouhal number S_t are defined respectively as

$$k = \frac{\omega C}{2U_\infty}, \quad S_t = \frac{fA}{U_\infty},$$

where C is the chord length of the airfoil, A is the pitching amplitude. The Reynolds number based on the airfoil chord length for all the simulations in this section is 12,600. The non-dimensional time step used for the two dimensional simulations is $\Delta\ tU_\infty/C = 1 \times 10^{-4}$; while that for the three dimensional simulations is $\Delta tU_\infty/C = 1 \times 10^{-5}$.

For the rigidly moving grid approach, the computational grid moves with the body and is updated using

$$\begin{cases} x_{present} - x_c = (x_{former} - x_c)\cos(\Delta\alpha) - (y_{former} - y_c)\sin(\Delta\alpha) \\ y_{present} - y_c = (x_{former} - x_c)\sin(\Delta\alpha) + (y_{former} - y_c)\cos(\Delta\alpha) \end{cases},$$

where (x_c, y_c) is the pitching center, and $\Delta\alpha\ = \alpha_0(\sin(\omega(t + dt) + \phi_0) - \sin(\omega t + \phi_0))$.

The deformable grid and the rigidly moving grid at maximum displacements for the St = 0.33 case are displayed in Fig. 10. There the grid with 341 × 47 × 1 cells for the two dimensional simulations and that with 341 × 47 × 10 cells for the three dimensional simulations are shown. The minimum cell size normalized by the airfoil chord length in the transverse direction is 0.001 and that in the spanwise direction for the three dimensional simulations is 0.02. A grid refinement study has been performed in [35] to determine this grid setup. The initial conditions for all simulations on the dynamic grids in the present section are set as the steady solutions of the flow fields under the same Reynolds number (Re = 12600) and inlet Mach number (Ma = 0.1). The effects of initial conditions on the bio-inspired flow simulations are discussed in [35], and it is found that the present initial conditions can best imitate the general experimental setups. The convergence history of the steady flow over the stationary NACA 0012 airfoil and the pressure coefficient (defined as $C_p = (p - p_\infty)/(0.5\rho U_\infty^2)$) contour are shown in Fig. 11.

The phase-averaged vorticity field from the experiment [2] and the corresponding instantaneous vorticity fields from the numerical simulations with different grid deformation algorithms are displayed in Fig. 12. In addition, the experimental and numerical results for the time-averaged vorticity and velocity fields are shown in Fig. 13. The numerical results are found to agree well with the experimental results. Thrust and lift coefficient histories for both deformable and rigidly moving grids are plotted in Fig. 14. According to [2], the mean thrust coefficient for the case Re = 12600, k = 11.5, S_t = 0.19 is around 0.024. In the present study, the mean thrust coefficient is calculated to be 0.031, and it is obtained by averaging the data in the continuous four cycles after twenty-four cycles. In addition, an interesting phenomenon discovered in the numerical simulation is that if the pitching amplitude is further increased, which means that the Strouhal number is increased, an asymmetric wake structure appears during the pitching motion. This was first reported in [10] for the plunging motion and has been experimentally studied in [29]. The vorticity fields with both deformable and rigidly moving grid are described in Fig. 15. The initial phase ϕ is set to be 180°. A three dimensional simulation is then conducted using the same parameters as that of the two dimensional simulations, except that in the spanwise direction, periodic boundary conditions are specified. From Fig. 15c and d and Fig. 16, it can be found that results from the 3D simulation are almost the same as those from the 2D simulations. This demonstrates that under the flow conditions specified in the present study, the flow is laminar and 2D simulations can predict the flow features well. The vortex structures in Fig. 15d are indicated by Q-criterion, which is described by

$$Q = \frac{1}{2}(R_{ij}R_{ij} - S_{ij}S_{ij}) = \frac{1}{2}\frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i},$$

where $R_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_i}{\partial x_j}\right)$ is the angular rotation tensor, and $S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_i}{\partial x_j}\right)$ is the rate-of-strain tensor. It also can be discovered from Fig. 16a that the thrust generation process appears certain unsteady features accompanying with the asymmetric wake structures. Again, it can be found from Fig. 16 that the numerical results do not depend on the grid deformation algorithms.

3.2.3. Flow over Tandem airfoils with inverse initial plunging phases

In order to enhance the thrust or lift generation and increase the propulsive efficiency, the tandem airfoil configuration has been studied by some researchers [1,20]. In these problems, the two airfoils have relative motions, which can be utilized to verify the grid deformation strategy for the SD method. Two flat plates performing plunging motions are studied here. The Reynolds number based on the plate chord length is 10,000. The motions of the two plates are specified as follows.

$$\begin{aligned} &\text{Fore plate}: \quad y = h\sin(\omega t + \phi_1) \\ &\text{Hind plate}: \quad y = h\sin(\omega t + \phi_2) \end{aligned}$$

where $h/C = 0.2$, the reduced frequency $k = 1.5$, $\phi_1\ = 0°$, and $\phi_2 = 180°$. The non-dimensional time step used for the simulations is $\Delta tU_\infty/C = 2 \times 10^{-4}$.

The deformable grid is displayed in Fig. 17. In order to compare the performances of high-order methods and their low-order

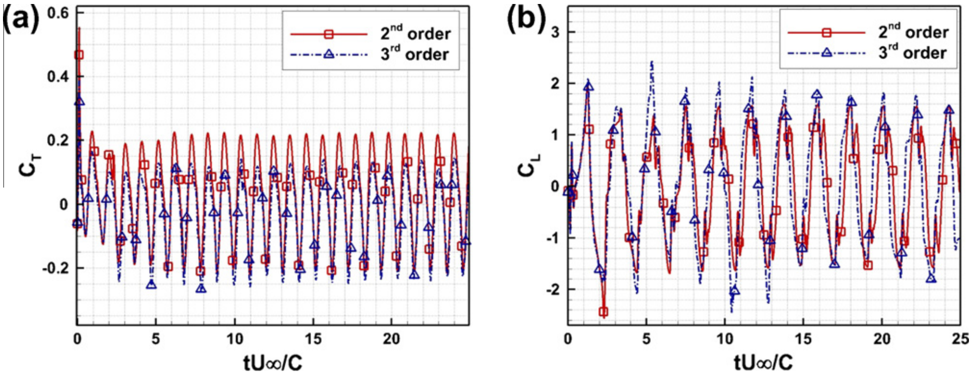

Fig. 19. (a) Thrust coefficient convergence history and (b) lift coefficient convergence history calculated using both the third-order scheme (as denoted by the dash–dot line with triangles) and the second-order scheme (as denoted by the solid line with squares).

counterparts, two sets of grids with almost the same degrees of freedom (DOFs) for the third- and second-order schemes are used in the simulations. A grid with 46,270 cells (185,080 DOFs) is designed for the second-order scheme; while another grid with 20,056 cells (180,504 DOFs) is designed for the third-order scheme. The computed vorticity fields from both third- andsecond-order accuracy schemes are shown in Fig. 18, and remarkable differences of small vortex structures near the moving wall boundaries can be observed for different accuracy approaches. Further, Fig. 19 displays the different aerodynamic force convergence histories for methods of different accuracy. The second-order scheme shows certain quasi-steady features after several cycles, which is not found from the results of the third-order scheme. This can be explained as follows. Due to the relatively high numerical dissipation, the second-order scheme can only capture the large vortex structures as seen from Fig. 18. As a comparison, the third-order scheme can resolve fine vortex structures near the wall boundaries with the same DOFs. These observations further demonstrate the necessity of high-order methods in vortex-dominated flows.

4. Conclusions

A high-order spectral difference method has been extended to solve compressible Navier–Stokes equations on deformable meshes. Since the present method is based on unstructured grids, it can handle complex geometries. Moreover, the differential form of the SD method makes the implementation straightforward even for high-order curved boundaries. Because a time-dependent transformation from the physical domain to the computational one has been made in the application of the method, the Geometric Conservation Law (GCL) has been carefully considered during the process and implemented for both the explicit and implicit time integration methods. It has been demonstrated that the developed algorithm preserved the high-order accuracy and works efficiently for several bio-inspired flow problems. Numerical tests clearly show that the high-order method with low numerical dissipation can resolve much more elaborate vortex structures than the low-order method, and can then help better illuminate the underlying physics of the vortex-dominated flow.

References

[1] Akhtar I, Mittal R, Lauder GV, Drucker E. Hydrodynamics of a biologically inspired tandem flapping foil configuration. Theor Comput Fluid Dyn 2007;21:155–70.

[2] Bohl DG, Koochesfahani MM. MTV measurements of the vertical field in the wake of an airfoil oscillating at high reduced frequency. J Fluid Mech 2009;620:63–88.

[3] Cockburn B, Shu C-W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. Math Comput 1989;52:411–35.

[4] Donea J. Arbitrary Lagrangian–Eulerian finite element methods. Computational methods for transient analysis (A84-29160 12-64). Amsterdam: North-Holland; 1983. p. 473–516.

[5] Dubuc L, Cantariti F, Woodgate M, Gribben B, Badcock KJ, Richards BE. A grid deformation technique for unsteady flow computations. Int J Numer Methods Fluids 2000;32:285–311.

[6] Eldredge JD, Wang CJ and OL MV. A computational study of a canonical pitch-up, pitch-down wing maneuver. AIAA Paper, 2009-3687; 2009.

[7] Hu FQ, Li XD, Lin DK. Absorbing boundary conditions for nonlinear Euler and Navier–Stokes equations based on the perfectly matched layer technique. J Comput Phys 2008;227:4398–424.

[8] Huynh HT. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. AIAA Paper, 2007-4079; 2007.

[9] Jameson A. A proof of the stability of the spectral difference method for all orders of accuracy. J Sci Comput 2010, doi:10.1007/s10915-009-9339-4.

[10] Jones KD, Dohring CM, Platzer MF. Experimental and computational investigation of the Knoller–Betz effect. AIAA J 1998;36(7):1240–6.

[11] Kopriva DA, Kolias JH. A conservative staggered-grid Chebyshev multi-domain method for compressible flows. J Comput Phys 1996;125(1):244–61.

[12] Liu Y, Vinokur M, Wang ZJ. Discontinuous spectral difference method for conservation laws on unstructured grids. J Comput Phys 2006;216:780–801.

[13] Mavriplis DJ, Nastase CR, On the geometric conservation law for high-order discontinuous GalerkinDiscretizations on dynamically deforming meshes. AIAA Paper, 2008-778; 2008.

[14] May G, Jameson A. A spectral difference method for the Euler and Navier–Stokes Eqs. AIAA Paper No. 2006–304; 2006.

[15] OL MV. The high-frequency, high-amplitude pitch problem: airfoils, plates and wings. AIAA Paper, 2009-3686; 2009.

[16] OL MV, Altman A, Eldredge JD, Garmann DJ, Lian YH. AIAA Paper, Résumé of the AIAA FDTC low Reynolds number discussion group's canonical cases. 2010-1085; 2010.

[17] Ou K, Jameson A. On the temporal and spatial accuracy of spectral difference method on moving deformable grids and the effect of geometric conservation law. AIAA Paper, 2010-5032; 2010.

[18] Ou K, Liang CH and Jameson A. A high-order spectral difference method for the Navier–Stokes equations on unstructured moving deformable grids. AIAA Paper, 2010-541; 2010.

[19] Persson PO, Peraire J, Bonet J. Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains. Comput Methods Appl Mech Eng 2009;198:1585–95.

[20] Platzer MF, Jones KD, Young J, Lai JCS. Flapping-wing aerodynamics: progress and challenges. AIAA J 2008;46(9):2136–49.

[21] Stein K, Tezduyar T, Benney R. Mesh moving techniques for fluid–structure interactions with large displacements. J Appl Mech 2003;70(1):58–63.

[22] Sun YZ, Wang ZJ, Liu Y. High-order multidomain spectral difference method for the Navier–Stokes equations on unstructured hexahedral grids. Commun Comput Phys 2006;2(2):310–33.

[23] Sun YZ, Wang ZJ, Liu Y. Efficient implicit non-linear LU-SGS approach for compressible flow computation using high-order spectral difference method. Commun Comput Phys 2009;5(2–4):760–78.

[24] Tannehill J, Anderson D, Pletcher R. Computational fluid mechanics and heat transfer. 2nd ed. Taylor & Francis; 1997.

[25] Thomas PD, Lombard CK. Geometric conservation law and its application to flow computations on moving grids. AIAA J 1979;17:1030–7.

[26] Vanden Abeele K, Lacor C, Wang ZJ. On the stability and accuracy of the spectral difference method. J Sci Comput 2008;37(2):162–88.

[27] Visbal MR, Gaitonde DV. On the use of high-order finite-difference schemes on curvilinear and deforming meshes. J Comput Phys 2002;181:155–85.

[28] Visbal MR. High-fidelity simulation of transitional flows past a plunging airfoil (2009), AIAA Paper No. 2009-391; 2009.

[29] von Ellenrieder KD, Pothos S. PIV measurement of the asymmetric wake of a two dimensional heaving hydrofoil. Exp Fluids 2007;43(5).

[30] Wuilbaut T. Algorithmic developments for a multi-physics framework, PhD. Thesis; 2008.

[31] Wang ZJ. High-order methods for the Euler and Navier Stokes equations on unstructured grids. Prog Aerosp Sci 2007;43:1–41.

[32] Wang ZJ. Spectral(finite)volume method for conservation laws on unstructured grids: basic formulation. J Comput Phys 2002;178:210–51.

[33] Wang ZJ, Liu Y, May G, Jameson A. Spectral difference method for unstructured grids II: extension to the Euler equations. J Sci Comput 2007;32:45–71.

[34] Wang ZJ, Sun Y, Liang C, Liu Y. Extension of the SD method to viscous flow on unstructured grids. In: Proceedings of the 4th international conference on computational fluid dynamics, Ghent, Belgium, July 2006.

[35] Yu ML, Hu H, Wang ZJ. A numerical study of vortex-dominated flow around an oscillating airfoil with high-order spectral difference method. AIAA Paper, 2010-726; 2010.