Journal of Computational Physics 256 (2014) 55–68

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An efficient correction procedure via reconstruction for simulation of viscous flow on moving and deforming domains

Chunlei Liang [a,*], Koji Miyaji [b], Bin Zhang [a]

[a] Department of Mechanical and Aerospace Engineering, George Washington University, DC 20052, United States
[b] Department of Ocean and Space System Engineering, Yokohama National University, Japan

ARTICLE INFO

Article history:
Received 19 February 2013
Received in revised form 27 June 2013
Accepted 25 August 2013
Available online 4 September 2013

Keywords:
Correction procedure via reconstruction
Unstructured grid
Quadrilateral element
Moving and deforming grids
Parallel Navier–Stokes solver

ABSTRACT

In this paper, we report the development of a new parallel solver using the Correction Procedure via Reconstruction (CPR) for viscous flows on moving and deforming grids. By employing an accurate treatment of flux derivatives for moving and deforming unstructured grids consisting of all quadrilateral cells, it is found that the Geometric Conservation Law is not explicitly required, the free-stream preservation is automatically satisfied. *The CPR code is verified using a benchmark case for a moving inviscid vortex on moving and deforming grids. The optimal orders of accuracy are obtained. It is subsequently employed to study viscous flows on moving and deforming grids. The CPR method is faster than and nearly as accurate as the SD method for solving viscous flow problems with moving boundaries*.

Published by Elsevier Inc.

1. Introduction

In 2007, Huynh [5] introduced a new approach to high-order accuracy by solving the equations in differential form. The approach, originally called flux reconstruction (FR), results in numerous schemes with favorable properties including an earlier developed spectral difference (SD) method [9,14] for quadrilateral elements. In 2009, Wang and Gao [20] extended the FR idea to 2D triangular and mixed meshes with the lifting collocation penalty (LCP) framework. The involved authors later combined the names of “FR” and “LCP” to call them CPR (Correction Procedure via Reconstruction). The CPR formulation is believed to be among the most efficient discontinuous methods in terms of the number of operations in Wang et al. [21]. *Correction functions are used to correct the discontinuous flux function within an element in order to ensure flux continuity across element interfaces. The g_2 scheme is the most efficient method in the family of CPR methods. The correction function of the g_2 scheme proposed by Huynh [6] and further developed here for moving and deforming domains is expressed in terms of a combination of Radau polynomials, where the zeros of the derivative of the correction function coincide with the Legendre–Lobatto points. Recently, Vincent et al. [19] reported a stability proof using a flux reconstruction formulation with energy estimates for a one parameter family of schemes where the family was expressed in terms of the Legendre polynomials instead of the Radau polynomials, proving stability for all orders of accuracy on unstructured grids*. Recently, Liang et al. [12] reported that the 4th-order CPR method could be over 40% faster than the SD method for viscous flow on 2D stationary grids with all quadrilateral cells.

While the SD or CPR method is quite established for simulations on stationary grids with complex geometries, its potential for simulating unsteady flow on moving and deforming domains could be very helpful for modeling flows associated with flapping wing flights, micro-air vehicles, oscillating wing energy harvesters, etc. To the best knowledge of the authors, the CPR method has not yet been applied for computations on moving and deforming grids. *The SD method is a staggered*

[*] Corresponding author. Tel.: +1 202 994 7073.
E-mail address: chliang@gwu.edu (C. Liang).

0021-9991/$ – see front matter Published by Elsevier Inc.
http://dx.doi.org/10.1016/j.jcp.2013.08.046

method requiring two grids for distributing flux and solution points at different locations. We consider a particular CPR method to co-locate the flux and solution points one-on-one within the standard element. The resulting discrete Geometric Conservation Law is satisfied automatically. In this paper, we extend the g_2 scheme originally proposed in Huynh [5,6] to solve both inviscid and viscous flows on moving and deforming grids consisting of all quadrilateral cells. By employing an accurate treatment of flux derivatives, we find that an enforcement of the Geometric Conservation Law by adding source terms [16] for free-stream preservation is no longer necessary.

The paper is arranged as the following. Section 2 presents the mathematical formulation of equations in both physical and computational domains. Section 3 gives the numerical formulation of the CPR method on moving and deforming grids. In Section 4, we test the accuracy of CPR method by solving the inviscid Euler vortex propagation problem on stationary and deforming grids. In Section 5, we validate our methods and solver by studying viscous flow over an oscillating cylinder, we also test the efficiency of CPR method on moving and deforming grids as well as the scalability of the parallelization. After that, in Section 6 and Section 7, we further validate the solver through the study of viscous subsonic flows around pitching and plunging airfoils. Section 8 concludes this paper.

2. Mathematical formulation

We consider compressible Euler as well as Navier–Stokes equations in 2D. The conservative form is given by

$$\frac{\partial \mathbf{Q}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=0, \tag{1}$$

where $\mathbf{Q}$ is the vector of conserved variables; $\mathbf{F}$ and $\mathbf{G}$ are flux vectors in two Cartesian directions which can include both inviscid and viscous flux terms.

In both CPR and SD methods, we employ an iso-parametric mapping to transform the fully-conservative equations from the physical domain onto a computational domain which allows universal reconstruction via polynomials. The computational domain is represented by a standard square element $0 \leqslant \xi \leqslant 1$, $0 \leqslant \eta \leqslant 1$, and a pseudo time τ. The Jacobian matrix for the mapping is,

$$\mathcal{J}=\frac{\partial(x,y,t)}{\partial(\xi,\eta,\tau)}=\begin{bmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ t_\xi & t_\eta & t_\tau \end{bmatrix}, \tag{2}$$

where $t_\xi = 0$, $t_\eta = 0$, and $t_\tau = 1$.

The inverse of matrix $\mathcal{J}$ is

$$\mathcal{J}^{-1}=\frac{\partial(\xi,\eta,\tau)}{\partial(x,y,t)}=\begin{bmatrix} \xi_x & \xi_y & \xi_t \\ \eta_x & \eta_y & \eta_t \\ \tau_x & \tau_y & \tau_t \end{bmatrix}=\frac{1}{|\mathcal{J}|}\mathcal{S}, \tag{3}$$

where $\tau_x = 0$, $\tau_y = 0$, and $\tau_t = 1$. And $\mathcal{S}$ is the transpose of the cofactor matrix of $\mathcal{J}$ which can be written as

$$\mathcal{S}=\begin{bmatrix} y_\eta & -x_\eta & A \\ -y_\xi & x_\xi & B \\ 0 & 0 & |\mathcal{J}| \end{bmatrix}, \tag{4}$$

with $A = x_\eta y_\tau - x_\tau y_\eta$, $B = x_\tau y_\xi - x_\xi y_\tau$ and $|\mathcal{J}| = x_\xi y_\eta - x_\eta y_\xi$.

The Geometric Conservation Law (GCL) can be expressed as

$$\begin{cases} \dfrac{\partial}{\partial \xi}\big(|\mathcal{J}|\xi_x\big)+\dfrac{\partial}{\partial \eta}\big(|\mathcal{J}|\eta_x\big)=0, \\ \dfrac{\partial}{\partial \xi}\big(|\mathcal{J}|\xi_y\big)+\dfrac{\partial}{\partial \eta}\big(|\mathcal{J}|\eta_y\big)=0, \\ \dfrac{\partial |\mathcal{J}|}{\partial \tau}+\dfrac{\partial}{\partial \xi}\big(|\mathcal{J}|\xi_\tau\big)+\dfrac{\partial}{\partial \eta}\big(|\mathcal{J}|\eta_\tau\big)=0. \end{cases} \tag{5}$$

The first two equations are automatically satisfied since the iso-parametric mapping is analytic and the metrics can be calculated exactly. We only have to consider the last equation which is time dependent.

In the following, we first give the classical form of Eq. (1) in the computational domain as was reported in Yu et al. [22] and Liang et al. [11]. Then we present the equation in a new form, namely the Liang–Miyaji form, which has been used in this paper.

2.1. The classical form

By directly using the chain rule, we can rewrite Eq. (1) into the following form,

$$\frac{\partial \mathbf{F}}{\partial \xi}\xi_x+\frac{\partial \mathbf{F}}{\partial \eta}\eta_x+\frac{\partial \mathbf{F}}{\partial \tau}\tau_x+\frac{\partial \mathbf{G}}{\partial \xi}\xi_y+\frac{\partial \mathbf{G}}{\partial \eta}\eta_y+\frac{\partial \mathbf{G}}{\partial \tau}\tau_y+\frac{\partial \mathbf{Q}}{\partial \xi}\xi_t+\frac{\partial \mathbf{Q}}{\partial \eta}\eta_t+\frac{\partial \mathbf{Q}}{\partial \tau}\tau_t=0. \tag{6}$$

Using the relations in Eq. (3), the above equation can be written as

$$\frac{1}{|\mathcal{J}|}\left\{\frac{\partial \mathbf{F}}{\partial \xi}\mathcal{S}_{1,1}+\frac{\partial \mathbf{F}}{\partial \eta}\mathcal{S}_{2,1}+\frac{\partial \mathbf{G}}{\partial \xi}\mathcal{S}_{1,2}+\frac{\partial \mathbf{G}}{\partial \eta}\mathcal{S}_{2,2}+\frac{\partial \mathbf{Q}}{\partial \xi}\mathcal{S}_{1,3}+\frac{\partial \mathbf{Q}}{\partial \eta}\mathcal{S}_{2,3}+\frac{\partial \mathbf{Q}}{\partial \tau}\mathcal{S}_{3,3}\right\}=0, \tag{7}$$

where the expressions for $\mathcal{S}_{i,j}$ can be found in Eq. (4).

If we use the product rule of differentiation and the last GCL equation, the above equation can be further written as,

$$\frac{1}{|\mathcal{J}|}\left\{\frac{\partial \mathbf{F}y_\eta}{\partial \xi}-\frac{\partial \mathbf{F}y_\xi}{\partial \eta}-\frac{\partial \mathbf{G}x_\eta}{\partial \xi}+\frac{\partial \mathbf{G}x_\xi}{\partial \eta}+\frac{\partial \mathbf{Q}A}{\partial \xi}+\frac{\partial \mathbf{Q}B}{\partial \eta}+\frac{\partial \mathbf{Q}|\mathcal{J}|}{\partial \tau}\right\}=0. \tag{8}$$

Rearrange Eq. (8), a final conservative form in the computational domain is obtained as,

$$\frac{\partial \tilde{\mathbf{F}}}{\partial \xi}+\frac{\partial \tilde{\mathbf{G}}}{\partial \eta}+\frac{\partial \tilde{\mathbf{Q}}}{\partial \tau}=0, \tag{9}$$

where $\tilde{\mathbf{F}}=\mathbf{F}y_\eta-\mathbf{G}x_\eta+\mathbf{Q}A$, $\tilde{\mathbf{G}}=\mathbf{G}x_\xi-\mathbf{F}y_\xi+\mathbf{Q}B$ and $\tilde{\mathbf{Q}}=|\mathcal{J}|\mathbf{Q}$. Written in matrix form, it will be

$$\begin{pmatrix}\tilde{\mathbf{F}}\\ \tilde{\mathbf{G}}\\ \tilde{\mathbf{Q}}\end{pmatrix}=|\mathcal{J}|\mathcal{J}^{-1}\begin{pmatrix}\mathbf{F}\\ \mathbf{G}\\ \mathbf{Q}\end{pmatrix}. \tag{10}$$

The formulation of Eq. (9) creates ease in code implementation because the well-known relation (Eq. (10)) for fixed grids holds now for moving and deforming grids carrying with a property of space–time method.

For problems involving moving and deforming physical grid, the last term in Eq. (9) can be expanded as,

$$\frac{\partial \tilde{\mathbf{Q}}}{\partial \tau}=\frac{\partial(|\mathcal{J}|\mathbf{Q})}{\partial \tau}=|\mathcal{J}|\frac{\partial \mathbf{Q}}{\partial \tau}+\mathbf{Q}\frac{\partial|\mathcal{J}|}{\partial \tau}, \tag{11}$$

where the term $\partial|\mathcal{J}|/\partial\tau$ is generally non-zero and is not readily available. By substituting the last GCL into the above equation, we can obtain,

$$\frac{\partial \tilde{\mathbf{Q}}}{\partial \tau}=|\mathcal{J}|\frac{\partial \mathbf{Q}}{\partial \tau}-\mathbf{Q}\left[\frac{\partial(|\mathcal{J}|\xi_t)}{\partial \xi}+\frac{\partial(|\mathcal{J}|\eta_t)}{\partial \eta}\right]. \tag{12}$$

Substitute the above equation into Eq. (9), by rearranging we get the final form,

$$\frac{\partial \mathbf{Q}}{\partial \tau}=\frac{1}{|\mathcal{J}|}\left\{\mathbf{Q}\left[\frac{\partial(|\mathcal{J}|\xi_t)}{\partial \xi}+\frac{\partial(|\mathcal{J}|\eta_t)}{\partial \eta}\right]-\left[\frac{\partial \tilde{\mathbf{F}}}{\partial \xi}+\frac{\partial \tilde{\mathbf{G}}}{\partial \eta}\right]\right\}. \tag{13}$$

This classical form was implemented by Yu et al. [22] and Liang et al. [11] in their SD codes, and will be used to obtain some comparative results against the CPR method using Liang–Miyaji form.

2.2. New form on moving and deforming domains (Liang–Miyaji form)

Instead of applying the product rule of differentiation and the GCL to Eq. (7), we rearrange it as,

$$\frac{\partial \mathbf{Q}}{\partial \tau}=-\frac{1}{|\mathcal{J}|}\left\{\frac{\partial \mathbf{F}}{\partial \xi}\mathcal{S}_{1,1}+\frac{\partial \mathbf{F}}{\partial \eta}\mathcal{S}_{2,1}+\frac{\partial \mathbf{G}}{\partial \xi}\mathcal{S}_{1,2}+\frac{\partial \mathbf{G}}{\partial \eta}\mathcal{S}_{2,2}+\frac{\partial \mathbf{Q}}{\partial \xi}\mathcal{S}_{1,3}+\frac{\partial \mathbf{Q}}{\partial \eta}\mathcal{S}_{2,3}\right\}, \tag{14}$$

and directly take the right-hand side as the residual $\mathcal{R}$, which leads to a set of ordinary differential equations (ODEs),

$$\frac{\partial \mathbf{Q}}{\partial \tau}=\mathcal{R}, \tag{15}$$

where

$$\mathcal{R}=\frac{1}{|\mathcal{J}|}\{\mathcal{F}_\xi+\mathcal{G}_\eta\}, \tag{16}$$

$$\mathcal{F}_\xi=\mathcal{F}_\xi^{inv}+\mathcal{F}_\xi^{v}=\frac{\partial \mathbf{F}}{\partial \xi}\mathcal{S}_{1,1}+\frac{\partial \mathbf{G}}{\partial \xi}\mathcal{S}_{1,2}+\frac{\partial \mathbf{Q}}{\partial \xi}\mathcal{S}_{1,3}, \tag{17}$$

$$\mathcal{G}_\eta=\mathcal{G}_\eta^{inv}+\mathcal{G}_\eta^{v}=\frac{\partial \mathbf{F}}{\partial \eta}\mathcal{S}_{2,1}+\frac{\partial \mathbf{G}}{\partial \eta}\mathcal{S}_{2,2}+\frac{\partial \mathbf{Q}}{\partial \eta}\mathcal{S}_{2,3}. \tag{18}$$

One of the advantages of using Eq. (15) is that the explicit Geometric Conservation Law is not required as long as we evaluate $\mathcal{S}_{1,3}$, $\mathcal{S}_{2,3}$, and $\mathcal{S}_{3,3}$ at every stage of the Runge–Kutta method for solving ODEs. In what follows we discuss several other merits of the Liang–Miyaji form.

If we rewrite Eq. (7) (or Eq. (14)) in the following form,

$$|\mathcal{J}|\frac{\partial \mathbf{Q}}{\partial \tau}+|\mathcal{J}|\left(\xi_x\frac{\partial \mathbf{F}}{\partial \xi}+\xi_y\frac{\partial \mathbf{G}}{\partial \xi}+\xi_t\frac{\partial \mathbf{Q}}{\partial \xi}+\eta_x\frac{\partial \mathbf{F}}{\partial \eta}+\eta_y\frac{\partial \mathbf{G}}{\partial \eta}+\eta_t\frac{\partial \mathbf{Q}}{\partial \eta}\right)=0, \tag{19}$$

then integrate the left-hand side within a standard reference element cell and within one time step, we obtain

$$\begin{aligned}
&\int_{\tau_1}^{\tau_2}\int_{\eta_1}^{\eta_2}\int_{\xi_1}^{\xi_2}|\mathcal{J}|\left(\frac{\partial \mathbf{Q}}{\partial \tau}+\xi_x\frac{\partial \mathbf{F}}{\partial \xi}+\xi_y\frac{\partial \mathbf{G}}{\partial \xi}+\xi_t\frac{\partial \mathbf{Q}}{\partial \xi}+\eta_x\frac{\partial \mathbf{F}}{\partial \eta}+\eta_y\frac{\partial \mathbf{G}}{\partial \eta}+\eta_t\frac{\partial \mathbf{Q}}{\partial \eta}\right)\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\tau\\
&\quad=\int_{\eta_1}^{\eta_2}\int_{\xi_1}^{\xi_2}\big[|\mathcal{J}|\mathbf{Q}\big]_{\tau_1}^{\tau_2}\mathrm{d}\xi\,\mathrm{d}\eta+\int_{\tau_1}^{\tau_2}\int_{\eta_1}^{\eta_2}\big[|\mathcal{J}|(\xi_x\mathbf{F}+\xi_y\mathbf{G}+\xi_t\mathbf{Q})\big]_{\xi_1}^{\xi_2}\mathrm{d}\eta\,\mathrm{d}\tau\\
&\qquad+\int_{\tau_1}^{\tau_2}\int_{\xi_1}^{\xi_2}\big[|\mathcal{J}|(\eta_x\mathbf{F}+\eta_y\mathbf{G}+\eta_t\mathbf{Q})\big]_{\eta_1}^{\eta_2}\mathrm{d}\xi\,\mathrm{d}\tau-\int_{\tau_1}^{\tau_2}\int_{\eta_1}^{\eta_2}\int_{\xi_1}^{\xi_2}\Big[\big(|\mathcal{J}|\big)_\tau\mathbf{Q}+\big(|\mathcal{J}|\xi_x\big)_\xi\mathbf{F}+\big(|\mathcal{J}|\xi_y\big)_\xi\mathbf{G}\\
&\qquad+\big(|\mathcal{J}|\xi_t\big)_\xi\mathbf{Q}+\big(|\mathcal{J}|\eta_x\big)_\eta\mathbf{F}+\big(|\mathcal{J}|\eta_y\big)_\eta\mathbf{Q}+\big(|\mathcal{J}|\eta_t\big)_\eta\mathbf{Q}\Big]\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\tau.
\end{aligned} \tag{20}$$

If all the GCL equations in Eq. (5) are satisfied, the above integral will be zero, which leads to a strong conservation form. In fact, the first two GCL equations are automatically satisfied as discussed previously. The third equation is satisfied if the motions of the mesh points are defined analytically, which can be expressed with a shape function M_i as,

$$x(\xi,\eta,\tau)=\sum_{i=1}^{k}M_i(\xi,\eta)x_i(\tau). \tag{21}$$

Therefore, the time derivative or grid velocity can also be analytically calculated.

In summary, our newly proposed formulation of the CPR method is: (1) free-stream (constant Q) preserving, (2) carrying a conservation property which is equivalent to the classical strong conservation form for moving and deforming grid. Interested readers are referred to Miyaji [15] for the discussion of the newly proposed form in terms of numerical accuracy on stationary grid.

3. Numerical formulation of the CPR method

Although the same number of solution points are used for both CPR and SD methods, we use a much smaller number of flux points for CPR than the SD method. Fig. 1(a) and (b) present a computational element used for the fourth-order SD and CPR methods respectively. However, the flux and solution points are collocated on the Legendre–Lobatto points in the CPR method. Note that $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ are stored at an identical set of flux points in the CPR method. For the fourth-order case, 16 flux points are employed to store $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ respectively.

The reconstructed fluxes are only element-wise continuous for both CPR and SD methods, but discontinuous across cell interfaces. We employ the simple Rusanov solver suggested by Rusanov [17] to compute the interface inviscid fluxes. An averaging approach reported in Kopriva [10] is employed for computing viscous fluxes at these interfaces in the SD method. However, in this paper, the BR2 scheme reported in Bassi et al. [2] is employed for computing viscous fluxes in the CPR method. *Although this method is different from that used for the SD code, the effect on the efficiency of the CPR method is small because an explicit time marching method is used throughout this paper.*

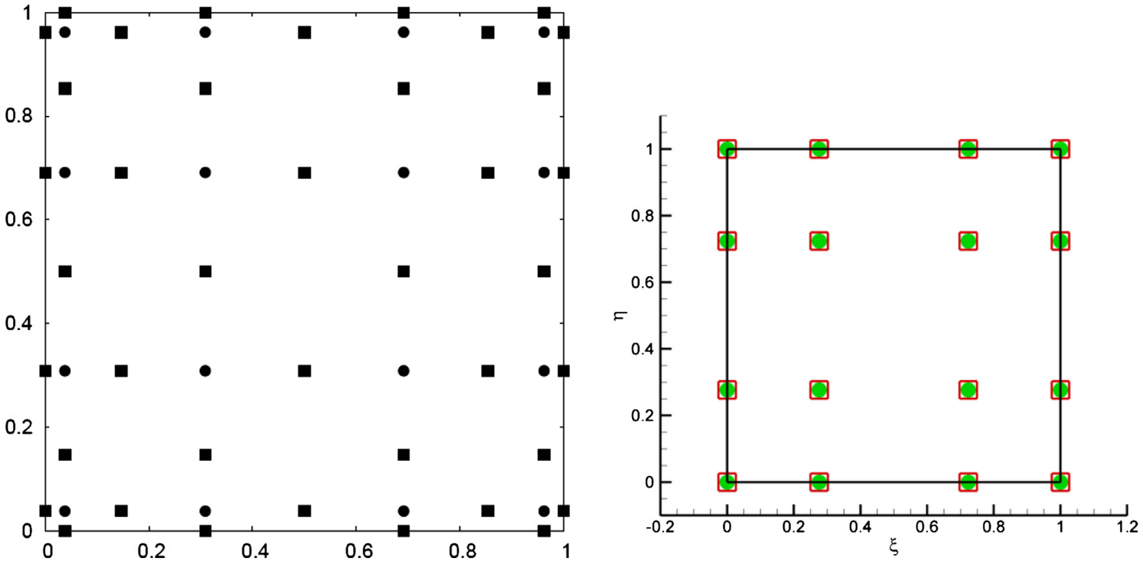
We employ a correction function g_2 as introduced in Huynh [6]. Among N Lobatto points, g_2' vanishes at $N-1$ of them. The g_2 scheme in a particular cell lumps corrections to its respective left (LB) and right (RB) boundaries.

In summary, the algorithm to compute the inviscid flux derivatives in CPR consists of the following steps:

(1) Given the conservative variables at the solution points, $\mathbf{F}(\xi)$ and $\mathbf{G}(\eta)$ are directly computed at the flux points. These values at solution points can form degree $N-1$ polynomials. Their derivatives can be computed using as degree $N-2$ polynomials in the corresponding cell to obtain $\mathbf{f}_\xi^{inv}=\frac{\partial \mathbf{F}}{\partial \xi}\mathcal{S}_{1,1}+\frac{\partial \mathbf{G}}{\partial \xi}\mathcal{S}_{1,2}+\frac{\partial \mathbf{Q}}{\partial \xi}\mathcal{S}_{1,3}$, and $\mathbf{g}_\eta^{inv}=\frac{\partial \mathbf{F}}{\partial \eta}\mathcal{S}_{2,1}+\frac{\partial \mathbf{G}}{\partial \eta}\mathcal{S}_{2,2}+\frac{\partial \mathbf{Q}}{\partial \eta}\mathcal{S}_{2,3}$, respectively without having any flux correction.

(2) The derivatives $\mathcal{F}_\xi^{inv}$ and $\mathcal{G}_\eta^{inv}$ on cell j interfaces can then be computed at the solution points (k) on cell boundaries via flux reconstruction. For instance,

$$\mathcal{F}_\xi^{inv}(j,k)=\mathbf{f}_\xi^{inv}(j,k)+\big[f_{j-\frac{1}{2}}^{com}-f_j(0)\big]g_{LB}'(\xi_k)+\big[f_{j+\frac{1}{2}}^{com}-f_j(1)\big]g_{RB}'(\xi_k). \tag{22}$$



(a) fourth-order SD method

(b) fourth-order CPR method

Fig. 1. Distribution of flux and solution points.

(3) The inviscid common fluxes (e.g. $f^{com}_{j-\frac{1}{2}}$) at the element interfaces are computed using the Rusanov solver suggested by Rusanov [17].

The procedure of CPR to get viscous fluxes can be described as the following steps:

(1) Compute the common solution $\mathbf{Q}^{com}_f = \frac{1}{2}(\mathbf{Q}^-_f + \mathbf{Q}^+_f)$ at cell interfaces.
(2) Similar to the procedure of Eq. (22), we can compute corrected $\nabla\mathbf{Q}_f$ as $\mathbf{R}^- = \nabla\mathbf{Q}^-_f + \mathbf{r}^-_f$ for left cell and $\mathbf{R}^+ = \nabla\mathbf{Q}^+_f + \mathbf{r}^+_f$ for right cell. $\mathbf{r}^-_f$ and $\mathbf{r}^+_f$ are determined by using similar correction functions in Eq. (22).
(3) The common gradient is computed by $\nabla\mathbf{Q}^{com}_f = \frac{1}{2}(\nabla\mathbf{Q}^-_f + \mathbf{r}^-_f + \nabla\mathbf{Q}^+_f + \mathbf{r}^+_f)$.
(4) The derivatives $\mathcal{F}^v_\xi$ and $\mathcal{G}^v_\eta$ with respect to viscous terms can then be computed at the solution points on cell boundaries (k) via flux reconstruction. For instance,

$$\mathcal{F}^v_\xi(j,k) = \mathbf{f}^v_\xi(\mathbf{Q},\mathbf{R}) + \left[f^v_{j-\frac{1}{2},com} - f^v_j(0)\right]g'_{LB}(\xi_k) + \left[f^v_{j+\frac{1}{2},com} - f^v_j(1)\right]g'_{RB}(\xi_k). \tag{23}$$

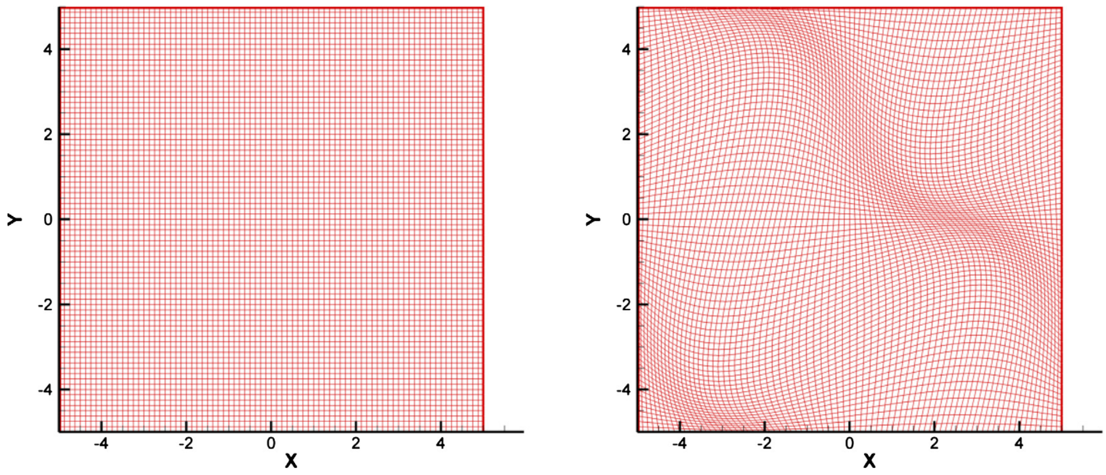
The above procedure of computing viscous fluxes is an analogy to the BR2 scheme proposed by Bassi et al. [2]. This discretization method, the I-continuous method proposed by Huynh [6], and the LDG2 method proposed by Kannan and Wang [8] are compact in the sense of involving only direct neighboring cells. The LD2 method is an extension of the original LDG method proposed by Cockburn and Shu [3].

In g_2 scheme, the evaluation of g'_{RB} and g'_{LB} is very simple according to Huynh [6]. For the computational element spans from 0 to 1, $g'_{RB} = N(N-1)$ and $g'_{LB} = N(1-N)$.

The CPR method employs less flux points. As a result, it reduces the number of operation in computing $\mathbf{F}(\mathbf{Q},\mathbf{R})$ and $\mathbf{G}(\mathbf{Q},\mathbf{R})$. The fourth-order CPR method computes $\mathbf{F}(\mathbf{Q},\mathbf{R})$ on 16 flux points in ξ direction and $\mathbf{G}(\mathbf{Q},\mathbf{R})$ on 16 flux points in η direction. The fourth-order SD method employs 20 flux points in each direction. Therefore, the SD method is 25% more expensive in computing $\mathbf{F}(\mathbf{Q},\mathbf{R})$ and $\mathbf{G}(\mathbf{Q},\mathbf{R})$. In addition, the g_2 method avoids extrapolating conservative variables $\mathbf{Q}$ of solution points onto flux points. Both CPR and SD methods require similar number of operation in order to compute common inviscid/viscous fluxes at cell interfaces where they have the same number of flux points. The primary cost in computing derivatives using the CPR method is dictated by a degree $(N-2)$ polynomial added with a small cost relevant to the correction (e.g. Eq. (22)). Therefore, it is more economical than the SD method which depends on a degree $N-1$ polynomial for computing derivatives.

3.1. Time marching scheme

All computations utilize a fourth-order accurate, strong-stability-preserving five-stage Runge–Kutta scheme introduced by Spiteri and Ruuth [18] for time marching.



(a) stationary grid

(b) deforming grid

Fig. 2. Grids for Euler vortex case.

4. Euler vortex with variable mapping

To verify our solver and show that we retain high-order spatial accuracy after the domain mapping, we solve an inviscid flow problem with a compressible vortex propagating in a square domain. The test case has similarity to the Euler vortex test case with variable mapping as employed in Persson et al. [16]. The initial conditions are given as

$$u = U_\infty \left\{ \cos\theta - \epsilon \frac{y}{r_c} \exp\left(-\frac{x^2+y^2}{2r_c^2} \right) \right\}, \tag{24}$$

$$u = U_\infty \left\{ \sin\theta + \epsilon \frac{x}{r_c} \exp\left(-\frac{x^2+y^2}{2r_c^2} \right) \right\}, \tag{25}$$

$$\rho = \rho_\infty \left\{ 1 - \frac{\gamma-1}{2} (\epsilon M_\infty)^2 \exp\left(-\frac{x^2+y^2}{r_c^2} \right) \right\}^{\frac{1}{\gamma-1}}, \tag{26}$$

$$p = p_\infty \left\{ 1 - \frac{\gamma-1}{2} (\epsilon M_\infty)^2 \exp\left(-\frac{x^2+y^2}{r_c^2} \right) \right\}^{\frac{\gamma}{\gamma-1}} \tag{27}$$

where ρ_∞, $U_\infty \cos\theta$, $U_\infty \sin\theta$, and p_∞ define the background flow. The superimposed vortex has a strength of $\epsilon = 1$. The concentration of this vortex has a radius of $r_c = 1$. The background Mach number is $M_\infty = 0.3$ and the computational domain size is $-5 \leqslant x, y \leqslant 5$. The initial vortex center locates at $(0, 0)$ and the free-stream velocity has an angle of $\theta = \arctan 1/2$. We use periodic boundary conditions. The final simulation time is $t_0 = 10\sqrt{5}$ when the vortex returns the origin $(0, 0)$. For all computations of this test case, we employ the time step size as $\frac{U_\infty \Delta t}{\Delta x} = 5 \times 10^{-5}$ in order to produce negligible numerical error from time integration.

The grid motion is given analytically by

$$x(X, Y, t) = X + \sin(\pi X/5)\sin(\pi Y/5)\sin(4\pi t/t_0), \tag{28}$$

and

$$y(X, Y, t) = Y + \sin(\pi X/5)\sin(\pi Y/5)\sin(8\pi t/t_0). \tag{29}$$

We studied the accuracy of spatial discretization of the CPR motion on stationary grids as well as deforming grids given by the above variable mapping. Fig. 2 shows a stationary grid (a) and a deforming grid (b) for simulating the propagating vortex.

Table 1 presents L2 errors predicted the fourth-order CPR method on four stationary grids. Table 2 presents L2 errors predicted the fourth-order CPR method on four deforming grids. The L2 errors on deforming grids are generally larger than the ones on stationary grids of same number of cells. Nevertheless, we are able to demonstrate the ability of the CPR method to obtain almost optimal order of accuracy on both stationary and deforming grids.

Fig. 3 presents a typical density contour plot obtained by the fourth-order CPR method on the finest grid with 80 × 80 cells. At the time instant, the vortex is approaching to the center from left and bottom side.

Table 1
L2 errors on stationary grids predicted by the fourth-order CPR method.

Cell number	L2 error	Order
10 × 10	3.08e−4	–
20 × 20	2.92e−5	3.4
40 × 40	1.33e−6	4.46
80 × 80	9.39e−8	3.79

Table 2
L2 errors on deforming grids predicted by the fourth-order CPR method.

Cell number	L2 error	Order
10 × 10	5.15e−4	–
20 × 20	7.96e−5	2.7
40 × 40	5.99e−6	3.73
80 × 80	3.92e−7	3.93

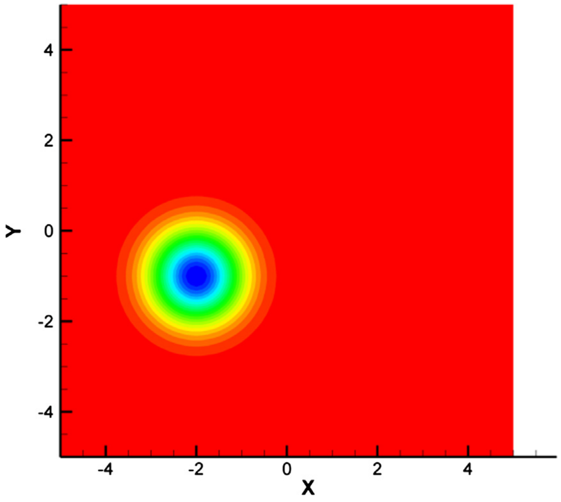


Fig. 3. Density contour when the vortex is approaching to the center.

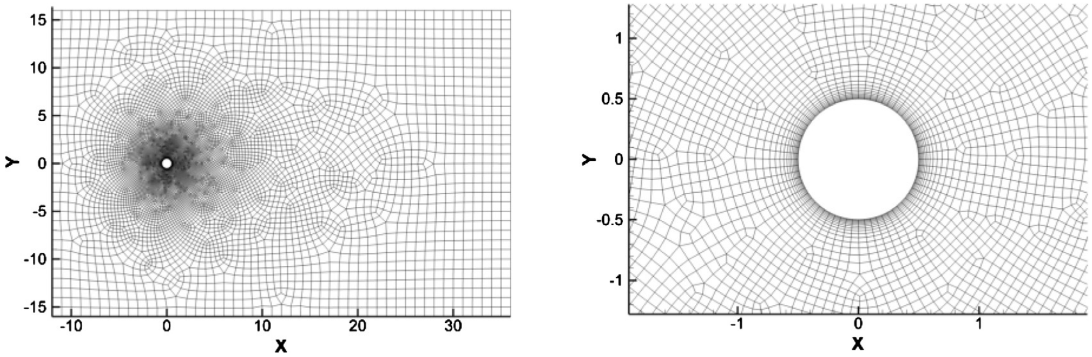
Without super-imposing a vortex, we can check the background flow alone using the same series of deforming grids for verifying the property of free-stream preservation. The Geometric Conservation Law was therefore considered in Persson et al. [16]. In the present formulation shown in Eq. (7), the free-stream preservation is strictly satisfied since the spatial derivatives of **F**, **G**, and **Q** are zero for uniform flow and will be maintained as zero in our CPR operations.

Recently, there is a surge of studies of unsteady flow aerodynamics around oscillating wings for harnessing wind and hydropower from renewable resources [7,1]. Understanding of vortex dominated flow generated by oscillating wings is very important for optimizing designs of oscillating-wing wind and hydropower generators. In the following sections, we study flows over oscillating cylinder, plunging airfoils as well as plunging and pitching airfoils.

5. Study of flow over an oscillating cylinder

In this section we study flow over a cylinder undergoing cross-flow oscillation. The parameters of the simulation are identical to those in a previous paper by Guilmineau and Queutey [4]. The Reynolds number based on the free-stream velocity and cylinder diameter is $Re = 185$, and the motion of the cylinder is given by $y(t) = -A_e \sin(2\pi f_e t)$, where $A_e = 0.2D$ is the magnitude of oscillation, $f_e = 1.1 f_0$ is the excitation frequency where f_0 is the natural shedding frequency from the same but stationary cylinder. To diminish the effect of compressibility, the free-stream Mach number is set to be 0.1.

Fig. 4 shows a global and local view of the computational mesh. The size of the domain is $48D$ in the x direction and $32D$ in the y direction, the cylinder is located at $12D$ downstream from the inlet. The first cell thickness in wall normal direction is $0.004D$. Totally 64 grid points are used around the periphery of the cylinder. This fine grid has 10 729 cells. In terms of degrees of freedom, this resolution is greater than that of the finest grid used by Guilmineau and Queutey [4] which was shown to give mesh independent solutions. In order to make sure the solutions we get are mesh independent, we ran the same cases on a coarser mesh with 3484 cells in all. The computational time step size for both coarse and fine grids is $\Delta t U_\infty / D = 1.0 \times 10^{-4}$.



(a) global view; (b) local view.

Fig. 4. Computational mesh for an oscillating cylinder.

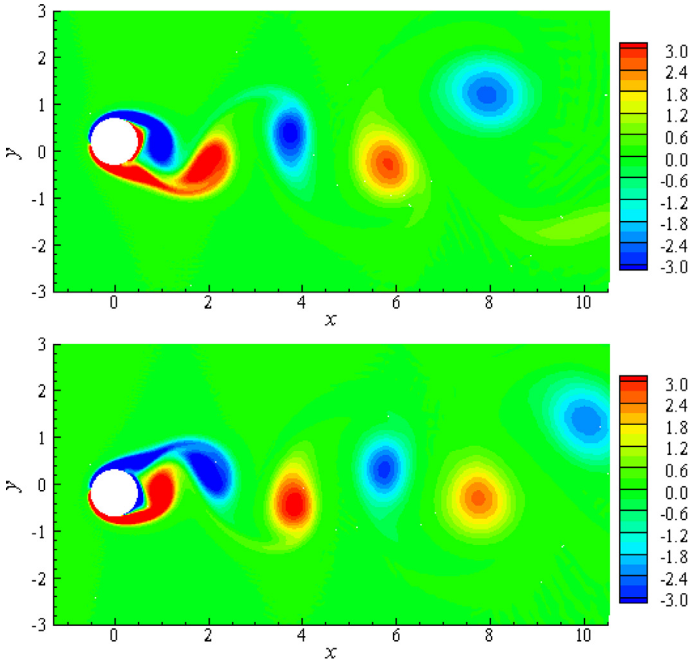


Fig. 5. Vorticity contour at two time instants for flow over an oscillating cylinder.

The motion of the mesh is given in two ways, i.e. a rigid moving grid and a deforming grid. For the rigid moving grid, the entire mesh moves as a rigid body together with the cylinder. For the deforming grid, only a small region within $r < 2$ (measured from the center of the cylinder) moves rigidly. The outer region ($r > 6$) remains fixed through the simulation. In between these two regions ($2 \leqslant r \leqslant 6$), we adopted a blended function as in Persson et al. [16] to regularize the mesh analytically. For a rigid moving grid, since the Jacobian for the transformation from the physical domain to the computational domain does not change with time which means less error will be introduced than a deforming grid, and thus can be used to compare with solution from a deforming grid. Meanwhile, simulations using our previously developed spectral difference solver [14] are also carried out to verify the CPR method here.

Fig. 5 shows vorticity contours at two time instants using the 4th-order CPR method on deforming grid, from which we see that the CPR method captures the vortical structures well. Fig. 6 shows the time history of lift and drag coefficients (all simulations are 4th order in space and time). It is seen that results from CPR method and SD method are very close, and results from rigid moving grid and deforming grid also agree well. Table 3 presents the statistics of C_L and C_D. It is found that results from different solvers and grids agree quite well. These results also agree well with previous results reported by Guilmineau and Queutey [4]. Note that there is little difference between the coarse mesh solution and fine mesh solution which indicates that our solutions are mesh independent.

The speeds of the CPR method and SD method (both tested on a single processor) are compared on the deforming grid using different order of accuracies. It is seen from Table 4 that the CPR method is generally more efficient than the SD method, especially when we use higher order of polynomials. We also tested the scalability of our parallelization of the CPR

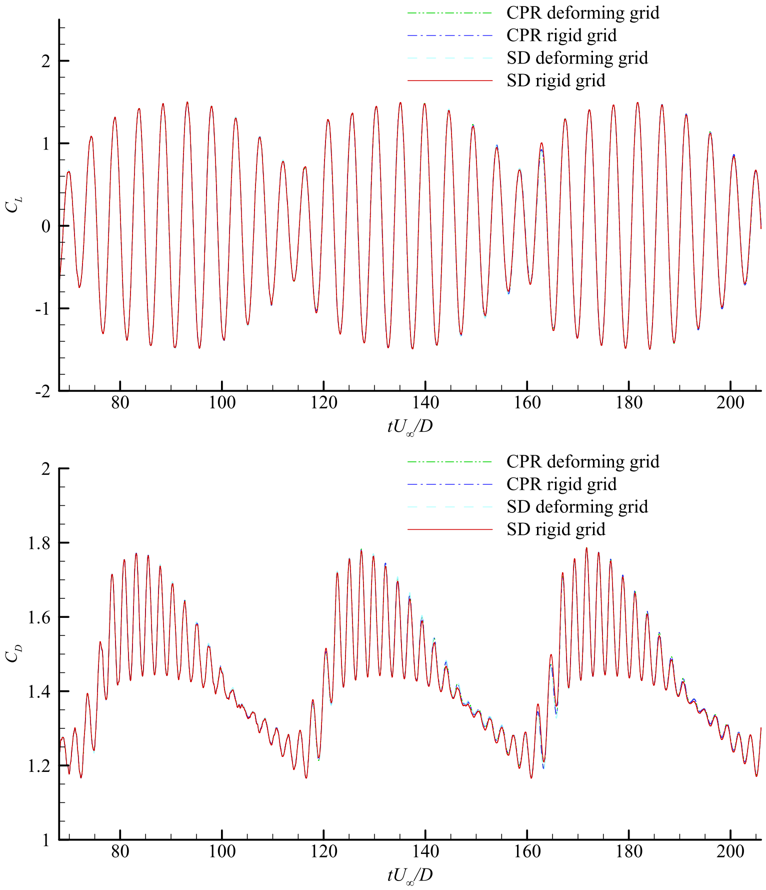


Fig. 6. Time history of lift and drag coefficients for oscillating cylinder.

Table 3
Statistics of lift and drag coefficients for oscillating cylinder.

	$\bar{C}_D$	$C_{D\ rms}$	$C_{L\ rms}$	St
CPR rigid moving grid (coarse)	1.424	0.148	0.893	0.214
CPR rigid moving grid	1.428	0.148	0.897	0.214
SD rigid moving grid	1.429	0.150	0.897	0.214
CPR deforming grid (coarse)	1.417	0.144	0.889	0.214
CPR deforming grid	1.419	0.144	0.895	0.214
SD deforming grid	1.419	0.147	0.885	0.214
Guilmineau et al. (180 × 200)	1.404	0.153	0.893	0.214
Guilmineau et al. (240 × 200)	1.420	0.149	0.897	0.214

Table 4
CPU time (in seconds) per 1000 time steps for CPR and SD method (tested on deforming grid of flow over an oscillating cylinder).

	2nd order	3rd order	4th order	5th order
CPR	228	420	712	1081
SD	230	498	907	1481
Speed improvement	0.9%	15.7%	21.5%	27.0%

code on the fine deforming grid, which is shown in Fig. 7. It can be seen that the speedup of the CPR method is almost linear with the number of processors.

6. Studies of plunging airfoils

In this section, we focus on laminar flow past plunging airfoils. The free-stream Mach number is set to 0.2 and the Reynolds number is 1850 based on free-stream velocity and airfoil chord length. The transverse motion of the airfoil is

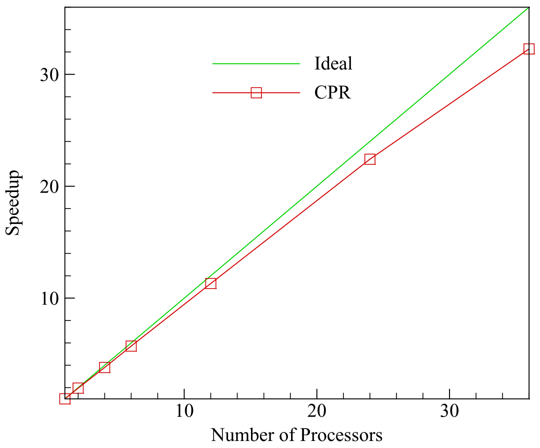
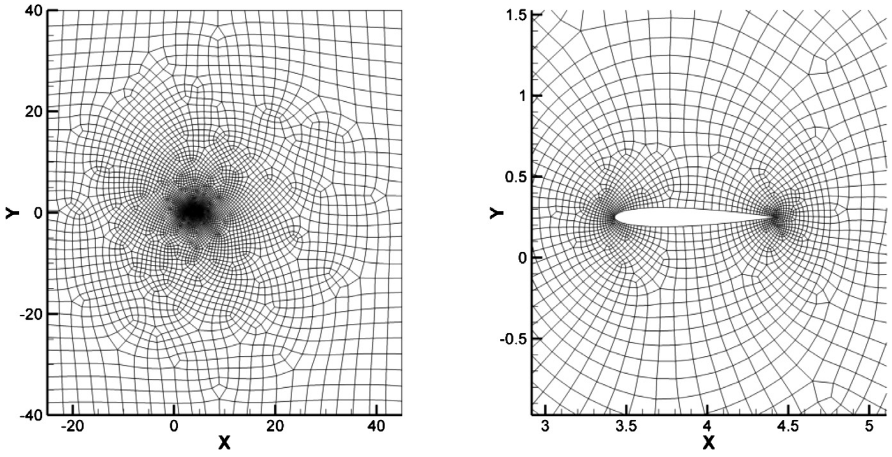


Fig. 7. Speedup of the 4th-order CPR method on deforming grid.



(a) global view;

(b) local view.

Fig. 8. A quadrilateral mesh for NACA0012 airfoil.

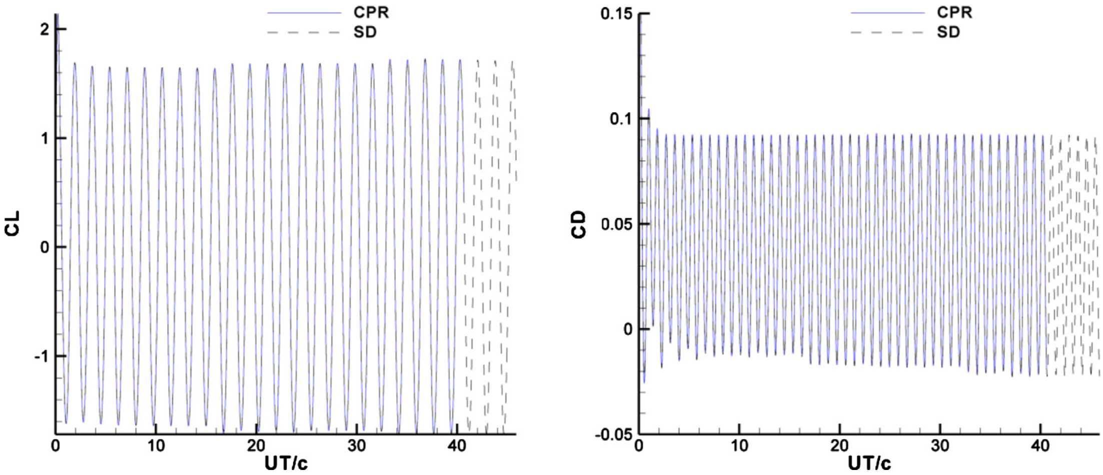
given by $h(t) = h \cdot \cos(\omega t)$. The amplitude of airfoil vibration is $h = 0.08$. The reduced frequency is $\tilde{k} = \omega c/U_\infty$ and the Strouhal number is $St = \tilde{k}h = 0.29$. The computational domain sizes for both cases are $70C$ in the x direction and $80C$ in the y direction, which are chosen to be large enough in order to eliminate the boundary effects, and the airfoils are located $25C$ away from the inlet, where C is the chord length.

6.1. NACA 0012

Fig. 8 presents a mesh with all quadrilateral elements for a NACA0012 airfoil. The grid has 6345 cells in all and 94 edges to represent the profile of this airfoil with closed trailing edge. The curvature of its profile is improved using a cubic spline fitting method. We checked the dependence of degrees of freedom for lift and drag coefficients using the fourth-order and fifth-order CPR methods on the above grid. Both methods produce almost identical lift and drag coefficients. In the following sections, we only present results obtained using the fifth-order methods for NACA 0012 airfoil and results of fourth-order methods for NACA 0014 airfoil.

In order to validate the accuracy of the CPR method on moving and deforming grids, we compare the results of lift and draft coefficients predicted by the fifth-order CPR method with the ones predicted by the fifth-order SD method. More details of the SD method on moving and deforming domains can be seen in Liang et al. [11] and Liang and DeJong [13]. As shown in Fig. 9, the newly developed CPR method obtains almost identical results to the lift and drag coefficients predicted by the SD method.

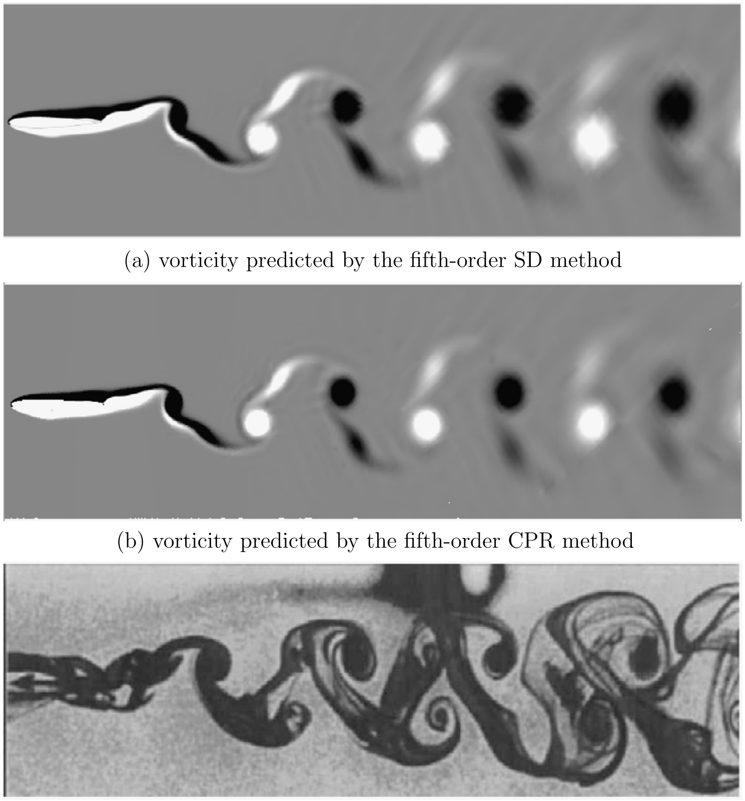
Fig. 10(a) shows vorticity contour predicted by the SD method. Fig. 10(b) presents our simulation results of vorticity contour using the CPR method. Both SD and CPR methods are able to predict vorticity contour plots which resemble the photograph shown in Fig. 10(c) obtained from experiments of Jones et al. [7].



(a) Lift coefficient;

(b) Drag coefficient.

Fig. 9. Lift and drag coefficients predicted by the fifth-order CPR method in comparison with the fifth SD method.



(a) vorticity predicted by the fifth-order SD method

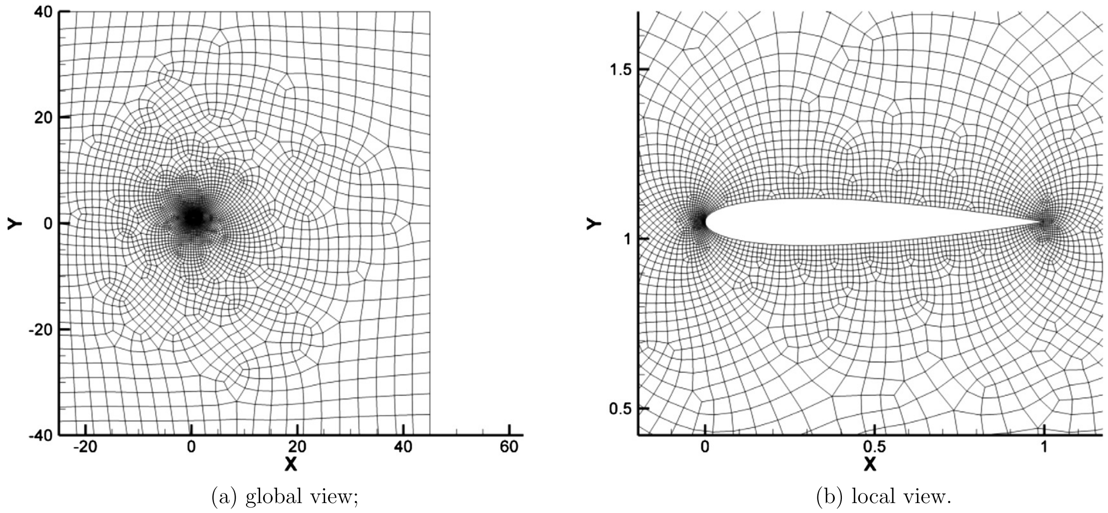
(b) vorticity predicted by the fifth-order CPR method

(c) experiment with dye injection

Fig. 10. Vorticity field predicted for flow around NACA0012 airfoil.

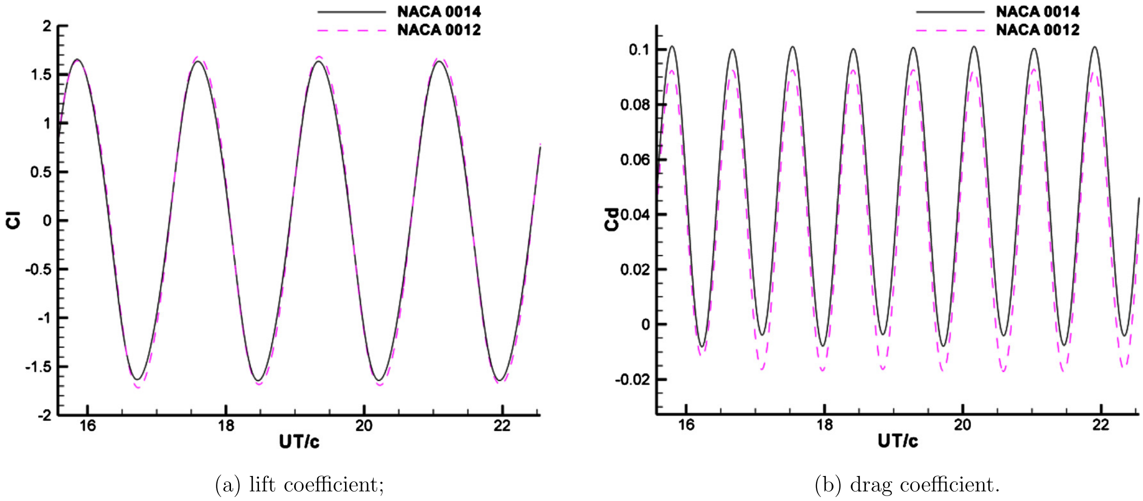
6.2. NACA 0014

In order to study the effect of airfoil thickness on the drag coefficient exerted by unsteady flow, we also study a plunging NACA 0014 airfoil undergoes the same transverse motion. Fig. 11 presents a mesh with all quadrilateral elements for a NACA0014 airfoil. The grid has 7770 cells in all and 200 edges to represent the profile of this airfoil. The curvature of its profile is improved again using a cubic spline fitting method.



(a) global view; (b) local view.

Fig. 11. A quadrilateral mesh for NACA0014 airfoil.



(a) lift coefficient; (b) drag coefficient.

Fig. 12. Lift and drag coefficients for plunging NACA0014 and NACA0012 airfoils.

Fig. 12 plots lift and drag coefficients for both NACA0012 and NACA0014 airfoils predicted by the fourth-order CPR method. It can be seen the lift coefficients are dominated by the effect of plunging motion and they experience little difference when varying the airfoil thickness from 0.12c to 0.14c. On the other hand, airfoil thickness has much higher impact on the drag coefficients. NACA0014 airfoil experiences much 25% increase of drag coefficient over that of NACA0012 airfoil. Specifically, the average drag coefficients for NACA0014 and NACA0012 airfoils are 4.75×10^{-2} and 3.79×10^{-2} respectively.

7. Studies of plunging and pitching airfoils

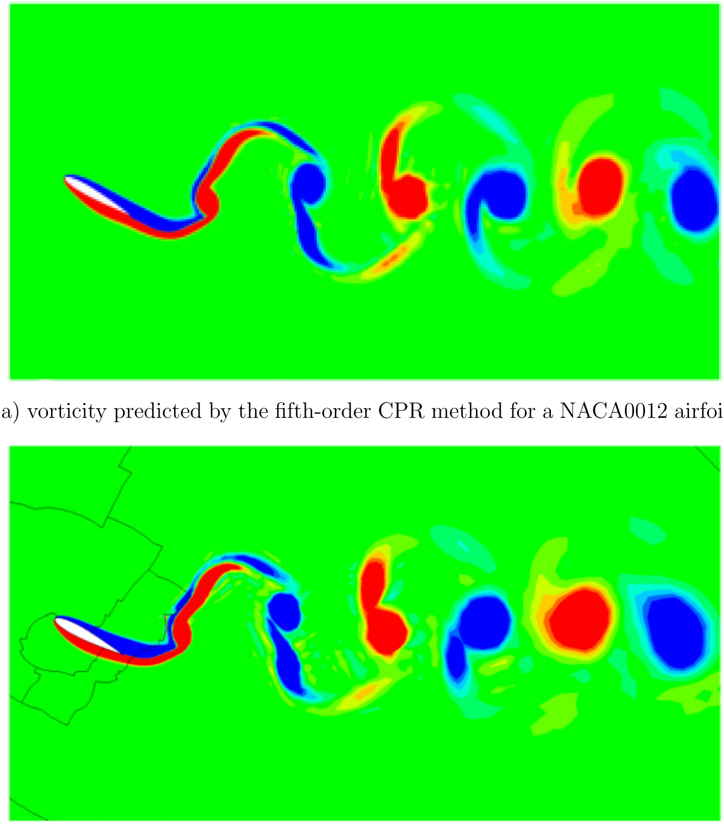
In this section, we again study both NACA0012 and NACA0014 airfoils plunging and pitching simultaneously. The airfoil's pitch center locates at the third chord, i.e. $c/3$ behind the leading edge. The time-dependent angle between the airfoil and the horizontal line is given by

$$\theta(t) = \alpha \sin(2\pi f t + \pi/2), \tag{30}$$

the plunging motion is governed by

$$h(t) = A \sin(2\pi f t). \tag{31}$$

We set the free-stream velocity to $U_\infty = 1$ giving a Reynolds number of 1000 and the reduced frequency $\frac{fc}{U_\infty} = 0.4$. The maximum pitching angle is $\alpha = \pi/6$, and the plunging amplitude A is set to 0.25. In addition, the free-stream Mach number



(a) vorticity predicted by the fifth-order CPR method for a NACA0012 airfoil

(b) vorticity predicted by the fourth-order CPR method for a NACA0014 airfoil

Fig. 13. Vorticity field predicted for flow around NACA0012 and NACA0014 airfoils with plunging and pitching motion (black lines mark processor interfaces).

is 0.2 and computational time step size is $\Delta t U_\infty / c = 1.25 \times 10^{-5}$. Following Persson et al. [16], we use a blended function to impose an analytical profile for the plunging and pitching of airfoils.

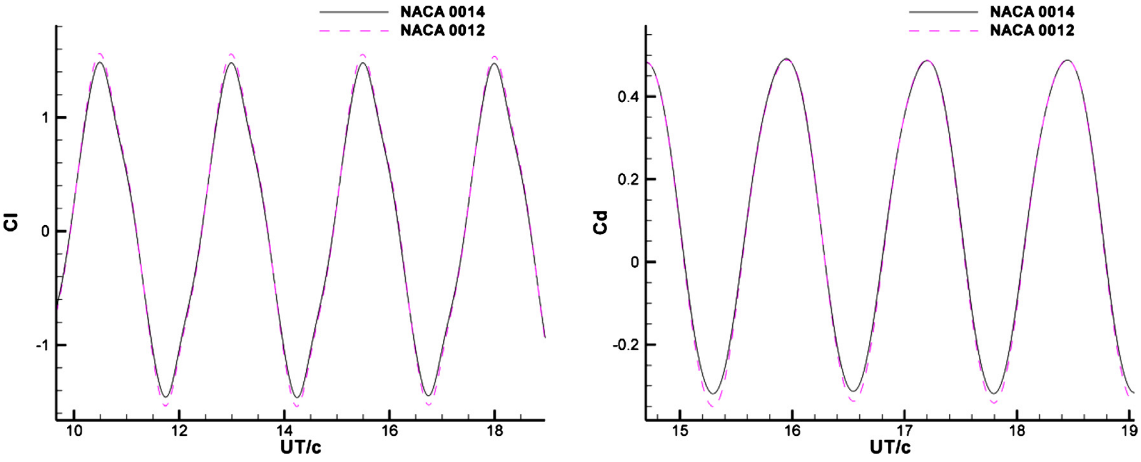
We present a typical contour plot for NACA0012 and NACA0014 airfoils respectively in Fig. 13 at a similar stroking phase. Note that the contour of vorticity is shown between -1 and 1 for $\omega c / U_\infty$ of 10 levels in Fig. 13. The downstream vortices behind the NACA0014 airfoil are larger as shown in Fig. 13.

Fig. 14 reports both lift and drag coefficients for NACA0014 and NACA0012 airfoils with plunging and pitching motion. It is shown that the average drag coefficient of NACA0014 is higher than its counterpart. By contrast, the lift coefficient of NACA0014 airfoil has smaller amplitude than that of NACA0012 airfoil. The time-average drag coefficients (excluding the initial development periods) for NACA0014 and NACA0012 are 0.106 and 0.0955 respectively. The increase of airfoil thickness enlarges the average drag coefficient for approximately 11%. It shall also be noted that the increase of average value of drag coefficient is much less than the pure plunging motion.

Although we obtained the optimal orders of accuracy for a moving inviscid vortex on moving and deforming domains, we do not have experimental results of viscous flow to quantitatively validate our calculations of viscous flows on moving and deforming domains. Nevertheless, interesting readers can look at a laminar flapping wing test case which was investigated in both the first and second international workshops on high-order CFD methods held in 2012 *and* 2013 *respectively. We simulated the laminar flapping wing problem using the SD and CPR methods respectively in* 2012 *and* 2013, *and we obtained quantitative agreements with the discontinuous Galerkin solutions from the groups of Drs. Krzysztof Fidkowski and Per-Olof Persson. The CPR method for dynamic grids developed in our present study is stable to obtain both fourth-order and fifth-order solutions for all viscous flow problems presented in this paper. We chose either the fourth-order or fifth-order method to deal with different problems so as to obtain grid-independent solutions.*

8. Concluding remark

We successfully developed a 2D solver using the CPR method on unstructured quadrilateral meshes for compressible Navier–Stokes equations in moving and deforming domains. The newly developed CPR method does not need the Geometric Conservation Law and is able to accurately predict both inviscid and viscous flows on moving and deforming grids. The method has been validated against other numerical methods as well as experimental results and is shown to be able to



(a) lift coefficient; (b) drag coefficient.

Fig. 14. Lift and drag coefficients for NACA0014 and NACA0012 airfoils with plunging and pitching motion.

achieve high fidelity solutions. Our CPR solver is also shown to be more efficient than the previously developed solver using the SD method. The scalability of the solver is found to be almost linear. We also found that the average drag coefficient for NACA0014 airfoil is 25% larger than that of NACA0012 airfoil for plunging cases and the increase reduces to 11% for plunging and pitching cases.

Acknowledgements

Chunlei Liang would like to acknowledge a research grant from ONR (program manager: Dr. Ki-Han Kim) with award no. N000141210500. He also acknowledges a JSPS invitation fellowship for his short-term visit of Yokohama National University in Japan (6/21-8/19/2012).

References

[1] M.A. Ashraf, J. Young, J.C. Lai, M.F. Platzer, Numerical analysis of an oscillating-wing wind and hydropower generator, AIAA J. 49 (2011) 1374–1386.
[2] F. Bassi, A. Crivellini, S. Rebay, M. Savini, Discontinuous Galerkin solution of the Reynolds-averaged Navier–Stokes and k–ω turbulence model equations, Comput. Fluids 34 (2005) 507?540.
[3] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws, J. Comput. Phys. 141 (1998).
[4] E. Guilmineau, P. Queutey, A numerical simulation of vortex shedding from an oscillating circular cylinder, J. Fluids Struct. 16 (2002) 773–794.
[5] H. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, AIAA paper AIAA-2007-4079, 2007.
[6] H. Huynh, A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion, AIAA paper AIAA-2009-403, 2009.
[7] K.D. Jones, T.C. Lund, M.F. Platzer, Experimental and computational investigation of flapping wing propulsion for micro air vehicles, Prog. Astronaut. Aeronaut. 195 (2001) 307–339.
[8] R. Kannan, Z.J. Wang, LDG2: A variant of the LDG viscous flux formulation for the spectral volume method, J. Sci. Comput. 46 (2011) 314–328.
[9] D.A. Kopriva, A conservative staggered-grid Chebyshev multidomain method for compressible flows. II A semi-structured method, J. Comput. Phys. 128 (1996) 475–488.
[10] D.A. Kopriva, A staggered-grid multidomain spectral method for the compressible Navier–Stokes equations, J. Comput. Phys. 143 (1998) 125–158.
[11] C. Liang, J. Chen, J.D. Lee, Spectral difference solution of unsteady 2d compressible micropolar equations on moving and deformable grids, in: AIAA Meeting, Nashville, TN, 2012 (AIAA-2012-294).
[12] C. Liang, C. Cox, M. Plesniak, A comparison of computational efficiencies of spectral difference method and correction procedure via reconstruction, J. Comput. Phys. 239 (2013) 138–146.
[13] C. Liang, A. DeJong, Massively parallel spectral difference solver for simulating vortex-induced vibrations of circular cylinders, in: International Mechanical Engineering Congress & Exposition IMECE (2012-93334), 2012.
[14] C. Liang, A. Jameson, Z.J. Wang, Spectral difference method for two-dimensional compressible flow on unstructured grids with mixed elements, J. Comput. Phys. 228 (2009) 2847–2858.
[15] K. Miyaji, Vortical flow simulations by a high-order accurate unstructured hexahedral grid method, Comput. Fluids 85 (2013) 78–84.
[16] P.O. Persson, J. Bonet, J. Peraire, Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains, Comput. Methods Appl. Mech. Eng. 198 (2009) 1585–1595.
[17] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267–279.
[18] R.J. Spiteri, S.J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, SIAM J. Numer. Anal. 40 (2002) 469–491.
[19] P.E. Vincent, P. Castonguay, A. Jameson, A new class of high-order energy stable flux reconstruction schemes, J. Sci. Comput. 47 (2011) 50–72.
[20] Z.J. Wang, H. Gao, A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids, J. Comput. Phys. 228 (2009) 8161–8186.
[21] Z.J. Wang, H. Gao, T. Haga, A Unifying Discontinuous Formulation for Hybrid Meshes, World Scientific Publishing, 2011.
[22] M.L. Yu, Z.J. Wang, H. Hu, A high-order spectral difference method for unstructured dynamic grids, Comput. Fluids 48 (2011) 84–97.