



# A Simple, Efficient, High-Order Accurate Sliding-mesh Interface Approach to FR/CPR Method on Coupled Rotating and Stationary Domains

Bin Zhang\*, Chunlei Liang†

*The George Washington University, Washington, DC 20052*

In this paper, we report a simple, efficient, high-order accurate sliding-mesh interface approach to the flux reconstruction/correction procedure via reconstruction (FR/CPR) method. We demonstrate the usefulness of this approach by solving 2D compressible Navier-Stokes equations on unstructured quadrilateral grids. This approach is an extension of the straight mortar method on stationary grid<sup>1,2</sup> to sliding-mesh interfaces with curved, dynamic mortars. On the sliding-mesh interfaces, common values of conservative variables as well as inviscid and viscous fluxes are computed on mortars. They are then projected back to the attached cell faces to ensure conservation. To demonstrate the spatial order of accuracy of the sliding-mesh FR/CPR method, both inviscid and viscous benchmark flows are simulated. It is shown that the sliding-mesh FR/CPR method preserves the high order accuracy of FR/CPR method, and is very efficient in terms of computational cost. This novel curved sliding-mesh interface method can have a wide range of applications, such as rotorcraft aerodynamics, wind turbine wake dynamics, marine propellers, and oscillating wind power generators.

## I. Introduction

During the last two decades, high-order (third and above) numerical methods have attracted wider range of interests in research communities, especially in the computational fluid dynamics (CFD) community. High-order methods are capable of producing more accurate solutions on relatively coarse grids at lower computational cost than low-order methods.<sup>3</sup> Researchers also began to use high-order methods to deal with very challenging fluid flow problems with complex geometries. Summaries of recent development and applications of high-order methods can be found in several books<sup>4,5,6,7</sup> and review papers.<sup>3,8,9</sup>

Among the numerous high-order methods, the discontinuous Galerkin (DG) method gained enormous popularity in the past decade for solving conservation laws on unstructured grid. The idea of the DG method was first introduced by Reed and Hill<sup>10</sup> to solve neutron transport equations. Cockburn, Shu, Bassi, Rebay and others<sup>7,11,12,13,14</sup> developed the DG method extensively and applied it to fluid dynamics problems. However, the computational complexity of the DG method increases rapidly as the order of accuracy increases.<sup>15</sup>

Instead of solving the equations in integral form as in the DG method, Kopriva<sup>16</sup> proposed a staggered-grid Chebyshev multidomain method, which solves the differential form of the conservation laws. Liu et al.<sup>17</sup> and Wang et al.<sup>18</sup> extended this method to unstructured triangular and quadrilateral grids and named the more general method as spectral difference (SD) method. For SD method on quadrilateral grids, solutions and fluxes in each coordinate direction in the computational domain are represented by 1D Lagrange polynomials, whereas the multi-dimensional solutions within each cell are reconstructed by tensor products of Lagrange polynomials. The 1D representation reduces the computational cost of fluxes and their derivatives dramatically, and makes the SD method very efficient and high-order accurate.

Recently, Huynh<sup>19,20,21</sup> introduced the flux reconstruction (FR) framework, which can further simplify the SD method. While in SD method, flux points and solution points are in a staggered fashion, they are

\*Ph.D Candidate, Department of Mechanical and Aerospace Engineering

†Assistant Professor, Department of Mechanical and Aerospace Engineering, AIAA Associate Fellow

collocated under the FR framework. Meanwhile, fluxes are corrected via a correction function that is one degree higher than solution polynomials. Huynh showed that for linear cases, depending on the choices of correction functions, FR method can recover several existing high-order methods, such as DG, SD and SV.<sup>22</sup> In addition, FR method results in numerous new schemes with favorable properties<sup>21</sup>. Wang and Gao<sup>23</sup> extended FR method to 2D unstructured grids under the lifting collocation penalty (LCP) framework. Since FR and LCP share similarities, the involved authors then named the two methods to correction procedure via reconstruction (CPR) method. Liang et al.<sup>24</sup> compared the efficiency between SD method and FR/CPR method with  $g_2$  correction function. They found that FR/CPR method is generally faster than SD method, for some cases, FR/CPR method can be as much as 45% faster. Yu et al.<sup>25</sup> carried out a comprehensive study on the accuracy and efficiency of DG, SD and FR/CPR methods. They found that FR/CPR method is much more efficient than DG, while the accuracies are comparable.

Liang et al.<sup>26</sup> extended FR/CPR method to solve Navier-Stokes equations on moving and deforming grids. They showed that FR/CPR method is more efficient than SD method even on dynamic grids, and their accuracies are close. However, the deforming grid approach fails when the grid undergoes very large rotational or translational motion, e.g. grid around a propeller. Therefore, a high-order accurate re-meshing technique shall be developed.

The low-order method community face a similar problem, and a few solutions were proposed in the past decades. One of the most popular approaches is the Chimera or overset approach,<sup>27,28</sup> where multiple component grids and background grids are used, and the communications are done through interpolations between the two sets of grids. Rai<sup>29,30</sup> introduced “zonal” and “patched-grid” approaches to 2D Euler equations. In his approaches, fluxes on the non-conforming interfaces are constructed through a weighted summation. Behr and Tezduyar<sup>31</sup> designed a “shear-slip mesh” method. In their method, grid components are connected through local re-meshing in shear-slip layers, and no over-lapping grids are involved. Several research groups<sup>32,33,34</sup> use halo nodes or halo cells approaches for rotating grids, where halo nodes or cells are extruded to the opposite side of the sliding interface to extrapolate information.

However, when it comes to high-order methods, such as FR/CPR method, the low-order approaches do not work directly. One reason is that in low-order approaches, solutions are often approximated as a constant or a linear function within each cell, while in high-order methods they are usually represented by higher degree polynomials, which leads to compatibility problems. Another reason is that many of these approaches are not conservative, and will degrade the accuracy of high-order methods. These reasons have led us to the development of a new method that is high-order accurate and efficient for the FR/CPR method.

In this paper, we present a novel curved dynamic mortar approach to the FR/CPR method with sliding-mesh interfaces. The mortar method was originally proposed for incompressible flow by Mavriplis.<sup>1</sup> Kopriva<sup>2,35</sup> applied this method to compressible Euler and Navier-Stokes equations on fixed multi-domain structured grids. In our approach, the pairing between cell faces and mortars, as well as the sizes and pairing offsets of mortars are changing over time. The common values of conservative variables as well as inviscid and viscous fluxes are computed on each mortar, and then projected back to the attached cell faces to ensure conservation. We show that our sliding-mesh approach is as simple as those designed for lower-order methods and it preserves the high-order accuracy of FR/CPR method. This simple but novel sliding-mesh FR/CPR method can have a wide range of applications, such as rotorcraft aerodynamics, wind turbine wake dynamics, marine propellers, and oscillating wing wind power generators.

The paper is organized as follows: Section II gives the mathematical equations. Section III reviews the FR/CPR method and presents the sliding-mesh FR/CPR method in details. Accuracy tests and applications are reported in Section IV. Finally, Section V concludes the paper.

## II. Mathematical formulation

### II.A. Compressible Navier-Stokes equations on stationary domain

We consider 2D unsteady compressible Navier-Stokes equations in conservative form,

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

where  $\mathbf{Q}$  is the vector of conservative variables,  $\mathbf{F}$  and  $\mathbf{G}$  are  $x$  and  $y$  components of the flux vector. These terms have the following expressions,

$$\mathbf{Q} = [\rho \ \rho u \ \rho v \ E]^T, \quad (2)$$

$$\mathbf{F} = \mathbf{F}_{inv}(Q) + \mathbf{F}_{vis}(Q, \nabla Q), \quad (3)$$

$$\mathbf{G} = \mathbf{G}_{inv}(Q) + \mathbf{G}_{vis}(Q, \nabla Q), \quad (4)$$

where  $\rho$  is fluid density,  $u$  and  $v$  are  $x$  and  $y$  components of velocity,  $E$  is the total energy per volume defined as  $E = p/(\gamma - 1) + \frac{1}{2}\rho(u^2 + v^2)$ ,  $p$  is pressure,  $\gamma$  is the ratio of specific heats and is set to 1.4.

As shown in Equations (3) and (4), the fluxes have been divided into inviscid and viscous parts. The inviscid fluxes are only functions of conservative variables which are,

$$\mathbf{F}_{inv} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{bmatrix}, \quad \mathbf{G}_{inv} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{bmatrix}. \quad (5)$$

The viscous fluxes are functions of the conservative variables as well as their gradients. They have the following expressions,

$$\mathbf{F}_{vis} = - \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ u\tau_{xx} + v\tau_{yx} + kT_x \end{bmatrix}, \quad \mathbf{G}_{vis} = - \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + kT_y \end{bmatrix}, \quad (6)$$

where  $\tau_{ij}$  is shear stress tensor which is related to velocity gradients as  $\tau_{ij} = \mu(u_{i,j} + u_{j,i}) + \lambda\delta_{ij}u_{k,k}$ ,  $\mu$  is dynamic viscosity,  $\lambda = -2/3\mu$  based on Stokes' hypothesis,  $\delta_{ij}$  is the Kronecker delta,  $k$  is thermal conductivity,  $T$  is temperature that is related to density and pressure through the ideal gas law  $p = \rho RT$ , where  $R$  is the gas constant.

## II.B. Compressible Navier-Stokes equations on rotating domain

On rotating domains, we implement an equation that is equivalent to the arbitrary Lagrange-Eulerian (ALE)<sup>36</sup> form of Equation (1). Due to grid motion, the inviscid fluxes are modified to take the following forms,

$$\mathbf{F}_{inv} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{bmatrix} - u_g \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{G}_{inv} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{bmatrix} - v_g \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad (7)$$

where  $u_g$  and  $v_g$  are the  $x$  and  $y$  components of grid velocity respectively. The viscous fluxes and all other variables stay unaffected and take the same expressions as those in the previous section.

For a domain rotating at angular velocity  $\boldsymbol{\omega}$ , the grid velocities are  $(u_g, v_g) = \boldsymbol{\omega} \times \mathbf{r}$ , where  $\mathbf{r}$  is the position vector with respect to rotating center. For all test cases in the present study,  $\boldsymbol{\omega}$  is known a priori, thus grid velocities and coordinates are updated analytically on the rotating domains.

## II.C. The transformed equations

As will be discussed in the next section, we map each quadrilateral cell in the physical domain to a standard square element in a computational domain. This mapping facilitates the construction of solution and flux polynomials. As a result, we only need to solve a set of transformed equations within each standard element. Let us assume that the physical coordinates  $(x, y)$  are mapped to the computational ones  $(\xi, \eta)$  through a transformation:  $x = x(\xi, \eta)$ ,  $y = y(\xi, \eta)$ . It can be shown that Equation (1) will take the following conservative form after coordinates transformation,

$$\frac{\partial \tilde{\mathbf{Q}}}{\partial t} + \frac{\partial \tilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \tilde{\mathbf{G}}}{\partial \eta} = 0, \quad (8)$$

where  $\tilde{\mathbf{Q}} = |\mathcal{J}|\mathbf{Q}$ , and the transformed fluxes  $\tilde{\mathbf{F}}$ ,  $\tilde{\mathbf{G}}$  are related to the physical ones as,

$$\begin{pmatrix} \tilde{\mathbf{F}} \\ \tilde{\mathbf{G}} \end{pmatrix} = |\mathcal{J}|\mathcal{J}^{-1} \begin{pmatrix} \mathbf{F} \\ \mathbf{G} \end{pmatrix}, \quad (9)$$

where  $|\mathcal{J}|$  is determinant of the Jacobian matrix,  $\mathcal{J}^{-1}$  is the inverse Jacobian matrix:

$$|\mathcal{J}| = \left| \frac{\partial(x, y)}{\partial(\xi, \eta)} \right| = \begin{vmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{vmatrix} = x_\xi y_\eta - x_\eta y_\xi, \quad (10)$$

$$\mathcal{J}^{-1} = \frac{\partial(\xi, \eta)}{\partial(x, y)} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \frac{1}{|\mathcal{J}|} \begin{bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{bmatrix}. \quad (11)$$

### III. Numerical methods

In this section, we first give a brief review of FR/CPR method. Subsequently, we describe a newly formulated sliding-mesh interface technique that is built on the FR/CPR formulation. For temporal discretization, an explicit strong stability preserving Runge-Kutta method<sup>37</sup> is used for all computations throughout this paper.

#### III.A. The FR/CPR method

For FR/CPR method on quadrilateral grids, we first transform each cell from the physical domain to a standard square element ( $0 \leq \xi \leq 1, 0 \leq \eta \leq 1$ ) in the computational domain. The transformation can be done through iso-parametric mapping. As was reported by previous researches,<sup>38,39</sup> using linear cells defined by 4 nodes will generate instabilities for problems involving curved boundaries. Thus, high order cells with 12 nodes are used along the curved boundaries to ensure stability and accuracy in the present study.

After iso-parametric mapping, solution points (SPs) and flux points (FPs) are defined on the element. There are several ways to arrange SPs and FPs.<sup>25,26,40</sup> In this study, they are arranged in a way as shown in Figure 1. For an  $N$ -th order FR/CPR method,  $N \times N$  collocating SPs and FPs are used in the interior, and  $N$  auxiliary points which align with SPs (FPs) are defined on each boundary edge to facilitate computation of interface fluxes. The SPs (FPs) are chosen as  $N$  Lobatto points along each direction. This configuration allows us to construct solution and fluxes as degree  $(N - 1)$  polynomials. Due to the existence of first order spatial derivatives on flux terms (Equation (8)), a degree  $N$  correction function  $g$  is employed to correct the fluxes. Here, we take  $g_{DG}$  as our correction function, more information on  $g_{DG}$  can be found in Huynh's papers.<sup>19,20,21</sup>

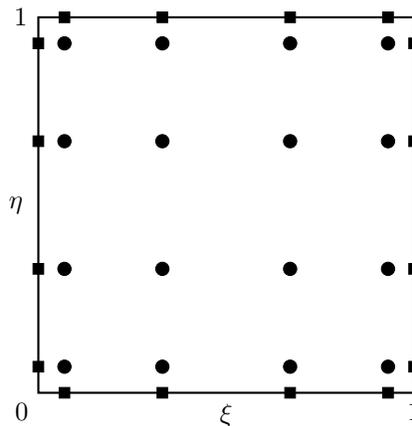


Figure 1. Schematic of arrangement of points for a fourth order FR/CPR method: circular dots, SPs (FPs); square dots, auxiliary boundary points for interface flux computation.

The basic steps of FR/CPR method are described as below:

- (a) Given  $\mathbf{Q}$  at SPs, compute  $\mathbf{F}_{inv}$  and  $\mathbf{G}_{inv}$  at SPs from Equations (5) and (7), and transform them to computational fluxes ( $\tilde{\mathbf{F}}_{inv}$  and  $\tilde{\mathbf{G}}_{inv}$ ) according to Equation (9). The solution and inviscid fluxes within each cell are reconstructed as,

$$\mathbf{Q}(\xi, \eta) = \sum_{j=1}^N \sum_{i=1}^N \frac{\tilde{\mathbf{Q}}_{i,j}}{|\mathcal{J}_{i,j}|} h_i(\xi) \cdot h_j(\eta), \quad (12)$$

$$\tilde{\mathbf{F}}_{inv}(\xi, \eta) = \sum_{j=1}^N \sum_{i=1}^N \tilde{\mathbf{F}}_{i,j}^{inv} h_i(\xi) \cdot h_j(\eta), \quad (13)$$

$$\tilde{\mathbf{G}}_{inv}(\xi, \eta) = \sum_{j=1}^N \sum_{i=1}^N \tilde{\mathbf{G}}_{i,j}^{inv} h_i(\xi) \cdot h_j(\eta), \quad (14)$$

where  $h$  represents the Lagrange basis,

$$h_\alpha(X) = \prod_{s=1, s \neq \alpha}^N \left( \frac{X - X_s}{X_\alpha - X_s} \right), \quad (15)$$

with  $X$  as coordinate,  $X_\alpha$  and  $X_s$  are SPs.

- (b) Compute  $\mathbf{Q}$ ,  $\tilde{\mathbf{F}}$  and  $\tilde{\mathbf{G}}$  on cell boundaries by Lagrange interpolation. Take the cell in Figure 1 for example, we will compute  $\mathbf{Q}_{0,j}$ ,  $\tilde{\mathbf{F}}_{0,j}^{inv}$ , and  $\tilde{\mathbf{G}}_{0,j}^{inv}$  for the left face,  $\mathbf{Q}_{N+1,j}$ ,  $\tilde{\mathbf{F}}_{N+1,j}^{inv}$ , and  $\tilde{\mathbf{G}}_{N+1,j}^{inv}$  for the right face,  $\mathbf{Q}_{i,0}$ ,  $\tilde{\mathbf{F}}_{i,0}^{inv}$ , and  $\tilde{\mathbf{G}}_{i,0}^{inv}$  for the bottom face, and  $\mathbf{Q}_{i,N+1}$ ,  $\tilde{\mathbf{F}}_{i,N+1}^{inv}$ , and  $\tilde{\mathbf{G}}_{i,N+1}^{inv}$  for the top face.
- (c) Compute common inviscid fluxes on cell interfaces using a Riemann solver. In the present work, the Rusanov solver<sup>41</sup> has been used for this purpose. Take the left cell face in Figure 1 as an example, the common inviscid flux is computed as,

$$\mathbf{F}_{0,j}^{inv,com} = \frac{1}{2} [(\mathbf{F}_{0,j}^{inv,-} + \mathbf{F}_{0,j}^{inv,+}) \mathbf{n} - \lambda (\mathbf{Q}_{0,j}^+ - \mathbf{Q}_{0,j}^-)], \quad (16)$$

where  $\lambda = |V_n| + c$  is the characteristic speed with the largest magnitude,  $V_n$  is the fluid velocity normal to the interface and  $c$  is the local sound speed. The superscripts ‘-’ and ‘+’ represent values on the left and right of the interface, respectively. The above common inviscid flux is transformed to computational one,  $\tilde{\mathbf{F}}_{0,j}^{inv,com}$ , through Equation (9).

- (d) Compute derivatives of corrected fluxes at SPs. For instance, the derivative of  $\tilde{\mathbf{F}}_{inv}$  is computed as,

$$(\tilde{\mathbf{F}}_{\xi}^{inv})_{i,j} = \sum_{\alpha=1}^N \tilde{\mathbf{F}}_{\alpha,j}^{inv} h'_{\alpha}(\xi_i) + [\tilde{\mathbf{F}}_{0,j}^{inv,com} - \mathbf{F}_{0,j}^{inv}] \cdot g'_L(\xi_i) + [\tilde{\mathbf{F}}_{N+1,j}^{inv,com} - \mathbf{F}_{N+1,j}^{inv}] \cdot g'_R(\xi_i), \quad (17)$$

where  $g_L$  and  $g_R$  are the left and right correction functions, respectively.

- (e) Compute common conservative variables on cell boundaries. Take the left cell face in Figure 1 for example,

$$\mathbf{Q}_{0,j}^{com} = \frac{1}{2} (\mathbf{Q}_{0,j}^- + \mathbf{Q}_{0,j}^+). \quad (18)$$

- (f) Compute gradients ( $\nabla \mathbf{Q}$ ) of corrected conservative variables. For example, the  $x$  component can be computed as,

$$(\mathbf{Q}_x)_{i,j} = \sum_{\alpha=1}^N \frac{\tilde{\mathbf{Q}}_{\alpha,j}}{|\mathcal{J}_{\alpha,j}|} h'_{\alpha}(\xi_i) \cdot (\xi_x)_{\alpha,j} + [(\mathbf{Q}_{0,j}^{com} - \mathbf{Q}_{0,j}) \cdot g'_L(\xi_i) + (\mathbf{Q}_{N+1,j}^{com} - \mathbf{Q}_{N+1,j}) \cdot g'_R(\xi_i)] \cdot (\xi_x)_{i,j}, \quad (19)$$

and  $(\mathbf{Q}_y)_{i,j}$  can be computed in the same way.

- (g) Compute  $\mathbf{F}_{vis}$  and  $\mathbf{G}_{vis}$  at SPs using Equation (6), and convert them to  $\tilde{\mathbf{F}}_{vis}$  and  $\tilde{\mathbf{G}}_{vis}$  through Equation (9). The reconstructed viscous fluxes in each cell have similar expressions as those for the inviscid fluxes in Equation (13) and (14).

- (h) Compute gradients and viscous fluxes on cell boundaries by Lagrange interpolation. Take the left cell face in Figure 1 for example, we will compute  $(\nabla \mathbf{Q})_{0,j}$ ,  $\tilde{\mathbf{F}}_{0,j}^{vis}$  and  $\tilde{\mathbf{G}}_{0,j}^{vis}$ . For the right boundary face, we will compute  $(\nabla \mathbf{Q})_{N+1,j}$ ,  $\tilde{\mathbf{F}}_{N+1,j}^{vis}$  and  $\tilde{\mathbf{G}}_{N+1,j}^{vis}$ , etc.
- (i) Compute common gradients and viscous fluxes on cell interfaces. Take the left cell face in Figure 1 for example,

$$(\nabla \mathbf{Q})_{0,j}^{com} = \frac{1}{2}((\nabla \mathbf{Q})_{0,j}^- + (\nabla \mathbf{Q})_{0,j}^+), \quad (20)$$

$$\mathbf{F}_{0,j}^{vis,com} = \mathbf{F}_{vis}(\mathbf{Q}_{0,j}^{com}, (\nabla \mathbf{Q})_{0,j}^{com}), \quad (21)$$

where the function  $\mathbf{F}_{vis}$  on the right hand side is defined in Equation (6), and the above physical flux is again transformed to computational one,  $\tilde{\mathbf{F}}_{0,j}^{vis,com}$ , through Equation (9).

- (j) Compute derivatives of corrected viscous fluxes as SPs. For instance, the derivative of  $\tilde{\mathbf{F}}_{vis}$  is computed as,

$$(\tilde{\mathbf{F}}_{\xi}^{vis})_{i,j} = \sum_{\alpha=1}^N \tilde{\mathbf{F}}_{\alpha,j}^{vis} h'_{\alpha}(\xi_i) + [\tilde{\mathbf{F}}_{0,j}^{vis,com} - \mathbf{F}_{0,j}^{vis}] \cdot g'_L(\xi_i) + [\tilde{\mathbf{F}}_{N+1,j}^{vis,com} - \mathbf{F}_{N+1,j}^{vis}] \cdot g'_R(\xi_i). \quad (22)$$

- (k) Now we have the flux derivatives from step (d) and step (j), residual can be computed and solution can be updated at the SPs.

Readers are referred to previous publications<sup>21,24,26</sup> for more details on FR/CPR method.

### III.B. The sliding interface treatment

Sliding-mesh interfaces are formed between rotating and stationary meshes. The simplest situation involves only one rotating mesh and one stationary mesh as shown in Figure 2. The inner mesh can rotate while the outer is fixed, or vice versa. Communication between the stationary and rotating meshes are realized through ‘‘mortars’’. To make the explanation intuitive, we have scaled the inner mesh in order to place mortars in between two coupled meshes.

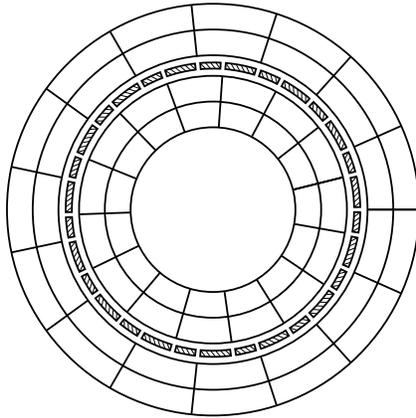


Figure 2. Schematic of distribution of mortars (hatched) between a rotating mesh and a stationary mesh.

The mortars are arranged in a counterclockwise order. We refer the inner mesh as left (L) and the outer mesh as right (R) with respect to mortars. To facilitate code implementation and reduce computational cost, cell faces on both sides of the sliding interface have been meshed uniformly. A closer look at Figure 2 reveals how mortars and cell faces on the sliding interface are connected: at each time instant, a cell face is connected to two mortars, and each mortar is associated with one left cell face and one right cell face. This face and mortar connectivity needs to be updated at every stage of the Runge-Kutta time stepping method.

Figure 3 shows a cell face  $\Omega$  and the attached two mortars  $\Xi_1$  and  $\Xi_2$ . Each curved mortar is mapped to a straight edge  $0 \leq z \leq 1$  through 1D iso-parametric mapping. Face  $\Omega$  is mapped to a straight edge  $0 \leq \xi \leq 1$

when the associated cell is mapped to a standard square element, thus no extra mapping is required.  $\xi$  and  $z$  are related by

$$\xi = o(t) + s(t)z, \quad (23)$$

where  $o(t)$  is an offset of the mortar relative to the bottom node of  $\Omega$  at time  $t$ , and  $s(t)$  is the relative scaling. For the example shown in Figure 3, we have  $o_1 = 0$  and  $s_1 = L^{\Xi_1}/L^\Omega$  for  $\Xi_1$ ,  $o_2 = L^{\Xi_1}/L^\Omega$  and  $s_2 = L^{\Xi_2}/L^\Omega$  for  $\Xi_2$ , where  $L$  means physical length of face and mortars.



**Figure 3. Mapping of curved cell face and mortars to straight ones: left, curved face and mortars in physical domain; right, straight face and mortars in computational domain.**

According to Equation (12), solutions on  $\Omega$  can be represented as,

$$\mathbf{Q}^\Omega = \sum_{i=1}^N \mathbf{Q}_i^\Omega h_i(\xi), \quad (24)$$

where  $\mathbf{Q}_i^\Omega$  represents solution at the  $i$ -th SP on  $\Omega$ ,  $h_i$  is the Lagrange basis defined in Equation (15). If we define the same set of SPs on  $0 \leq z \leq 1$  for each mortar, then solutions on each mortar can be reconstructed similarly,

$$\mathbf{Q}^\Xi = \sum_{i=1}^N \mathbf{Q}_i^\Xi h_i(z), \quad (25)$$

where  $\mathbf{Q}_i^\Xi$  is the solution at the  $i$ -th SP on a mortar  $\Xi$ .

The procedure for computing  $\mathbf{Q}_i^\Xi$  is demonstrated in Figure 4(a). For simplicity, we only show the process on the left side of mortar  $\Xi$ . To get the solutions, we require that,

$$\int_0^1 (\mathbf{Q}^{\Xi,L}(z) - \mathbf{Q}^\Omega(\xi)) h_j(z) dz = 0, \quad j = 1, 2, \dots, N. \quad (26)$$

Substitute Equations (23)-(25) into the above equation and evaluate it at each SP on  $\Xi$  will give a system of linear equations. The solution of this system when written in matrix form is,

$$\mathbf{Q}^{\Xi,L} = \mathbf{P}^{\Omega \rightarrow \Xi} \mathbf{Q}^\Omega = \mathbf{M}^{-1} \mathbf{S}^{\Omega \rightarrow \Xi} \mathbf{Q}^\Omega, \quad (27)$$

where  $\mathbf{P}^{\Omega \rightarrow \Xi}$  is the projection matrix from  $\Omega$  to  $\Xi$ , the matrices  $\mathbf{M}$  and  $\mathbf{S}^{\Omega \rightarrow \Xi}$  have the following elements,

$$M_{i,j} = \int_0^1 h_i(z) h_j(z) dz, \quad i, j = 1, 2, \dots, N, \quad (28)$$

$$S_{i,j}^{\Omega \rightarrow \Xi} = \int_0^1 h_i(o + sz) h_j(z) dz, \quad i, j = 1, 2, \dots, N, \quad (29)$$

where  $o$  and  $s$  are the offset and scaling of  $\Xi$  with respect to  $\Omega$ . It is important to note that  $o$  and  $s$  are time-dependent for the sliding-mesh interface method.

The right solution vector  $\mathbf{Q}^{\Xi,R}$  can be computed in the same way. Having both left and right solutions on a mortar, the Rusanov solver<sup>41</sup> is employed to compute common inviscid flux  $\mathbf{F}_{inv}^\Xi$ . This flux is then transformed to the computational flux as  $\tilde{\mathbf{F}}_{inv}^\Xi$  according to Equation (9).

As shown in Figure 4(b), to project the common inviscid fluxes  $\tilde{\mathbf{F}}_{inv}^{\Xi_1}$  and  $\tilde{\mathbf{F}}_{inv}^{\Xi_2}$  back to face  $\Omega$ , we require that,

$$\int_0^{o_2} (\tilde{\mathbf{F}}_{inv}^\Omega(\xi) - \tilde{\mathbf{F}}_{inv}^{\Xi_1}(z)) h_j(\xi) d\xi + \int_{o_2}^1 (\tilde{\mathbf{F}}_{inv}^\Omega(\xi) - \tilde{\mathbf{F}}_{inv}^{\Xi_2}(z)) h_j(\xi) d\xi = 0, \quad j = 1, 2, \dots, N, \quad (30)$$



**Figure 4. Projection between face and mortar: (a) from left face to left side of mortar, (b) from two mortars back to the associated left face.**

where  $\tilde{\mathbf{F}}_{inv}^{\Omega}(\xi)$  is the inviscid flux polynomial on face  $\Omega$ . Solution of the above equation when written in matrix form is,

$$\tilde{\mathbf{F}}_{inv}^{\Omega} = \mathbf{P}^{\Xi_1 \rightarrow \Omega} \tilde{\mathbf{F}}_{inv}^{\Xi_1} + \mathbf{P}^{\Xi_2 \rightarrow \Omega} \tilde{\mathbf{F}}_{inv}^{\Xi_2} = s_1 \mathbf{M}^{-1} \mathbf{S}^{\Xi_1 \rightarrow \Omega} \tilde{\mathbf{F}}_{inv}^{\Xi_1} + s_2 \mathbf{M}^{-1} \mathbf{S}^{\Xi_2 \rightarrow \Omega} \tilde{\mathbf{F}}_{inv}^{\Xi_2}, \quad (31)$$

where the matrix  $\mathbf{M}$  is identical to that in Equation (27), and matrices  $\mathbf{S}^{\Xi_1 \rightarrow \Omega}$  and  $\mathbf{S}^{\Xi_2 \rightarrow \Omega}$  are simply transposes of  $\mathbf{S}^{\Omega \rightarrow \Xi_1}$  and  $\mathbf{S}^{\Omega \rightarrow \Xi_2}$ , respectively.

For the computation of common viscous fluxes, we first compute the common solution on each mortar as the average of the left and right solutions,

$$\mathbf{Q}^{\Xi} = \frac{1}{2}(\mathbf{Q}^{\Xi,L} + \mathbf{Q}^{\Xi,R}). \quad (32)$$

This common solution is then projected back to cell faces in the same procedure as for the inviscid flux in Equation (31). After that, solution gradients and viscous fluxes are updated on both side of the interfaces. The viscous fluxes  $\tilde{\mathbf{F}}_{vis}^{\Omega}$  on cell faces are projected to mortars in the same way as Equation (27). The common viscous flux  $\tilde{\mathbf{F}}_{vis}^{\Xi}$  on a mortar is the average of left and right viscous fluxes,

$$\tilde{\mathbf{F}}_{vis}^{\Xi} = \frac{1}{2}(\tilde{\mathbf{F}}_{vis}^{\Xi,L} + \tilde{\mathbf{F}}_{vis}^{\Xi,R}). \quad (33)$$

The final step is to project  $\tilde{\mathbf{F}}_{vis}^{\Xi}$  back to faces, which is identical to the process in Equation (31).

Since uniform mesh is used for cell faces on the sliding interface, the  $\mathbf{S}$  matrix only need to be computed for the first two mortars, and can be reused by other corresponding mortars. At the same time since the  $\mathbf{M}$  matrix is time independent, it can be precomputed before the actual calculation. To compute the integrals in Equations (28) and (29), one can use the Clenshaw-Curtis quadrature method<sup>42</sup> as was used by previous researchers. In this study, the integrand is casted into a general form as a product of  $2(N - 1)$  first degree polynomials, and we implement a recursive algorithm to compute the integrals analytically. This approach requires the least number of operations which is much more efficient than the Clenshaw-Curtis quadrature method.

## IV. Numerical tests

In this section we test the accuracy of the sliding-mesh FR/CPR method on both inviscid and viscous flows, and then apply this method to study flows around a rotating elliptic cylinder and a pitching airfoil. A five-stage fourth order Runge-Kutta method for time stepping<sup>37</sup> is used for all test cases. For accuracy tests, various time step sizes were used in each case to make sure that the errors are almost time step size independent and are dominated by spacial discretization errors.

### IV.A. Euler vortex flow

We first test the solver on an inviscid flow. For inviscid flow test, Euler vortex problem is an ideal choice and it has been used by many researchers, one of such an example can be found in the paper by Erlebacher et al.<sup>43</sup> In Euler vortex problem, an isentropic vortex is superimposed to an uniform mean flow and convected

by the mean flow. The flow field in an infinite domain at a time instant  $t$  can be analytically expressed as,

$$u = U_\infty \left\{ \cos \theta - \frac{\epsilon y_r}{r_c} \exp \left( \frac{1 - x_r^2 - y_r^2}{2r_c^2} \right) \right\} \quad (34)$$

$$v = U_\infty \left\{ \sin \theta + \frac{\epsilon x_r}{r_c} \exp \left( \frac{1 - x_r^2 - y_r^2}{2r_c^2} \right) \right\} \quad (35)$$

$$\rho = \rho_\infty \left\{ 1 - \frac{(\gamma - 1)(\epsilon M_\infty)^2}{2} \exp \left( \frac{1 - x_r^2 - y_r^2}{r_c^2} \right) \right\}^{\frac{1}{\gamma-1}} \quad (36)$$

$$p = p_\infty \left\{ 1 - \frac{(\gamma - 1)(\epsilon M_\infty)^2}{2} \exp \left( \frac{1 - x_r^2 - y_r^2}{r_c^2} \right) \right\}^{\frac{\gamma}{\gamma-1}} \quad (37)$$

where  $U_\infty$ ,  $\rho_\infty$ ,  $p_\infty$ ,  $M_\infty$  are the mean flow speed, density, pressure and Mach number respectively,  $\theta$  is the direction of the mean flow (i.e. the direction along which the vortex is convected),  $\epsilon$  and  $r_c$  can be interpreted as vortex strength and size. The relative coordinates  $(x_r, y_r)$  are defined as,

$$x_r = x - x_0 - \bar{u}t, \quad (38)$$

$$y_r = y - y_0 - \bar{v}t, \quad (39)$$

where  $\bar{u} = U_\infty \cos \theta$ ,  $\bar{v} = U_\infty \sin \theta$  are the  $x$  and  $y$  components of the mean velocity,  $(x_0, y_0)$  is the initial position of the vortex. The exact solution of Euler vortex within a square domain  $(0 \leq x, y \leq L)$  with periodic boundary conditions can be achieved by replacing the relative coordinates with the following expressions,

$$x_r = x_r - \left[ \frac{x_r + x_0}{L} \right] \cdot L, \quad (40)$$

$$y_r = y_r - \left[ \frac{y_r + y_0}{L} \right] \cdot L, \quad (41)$$

where the floor operator  $[x]$  gives the largest integer that is not greater than a real number  $x$ , the  $x_r$  and  $y_r$  on the right hand side are from Equation (38) and (39).

In this test, the uniform mean flow is chosen as  $(U_\infty, \rho_\infty, p_\infty) = (1, 1, 1)$  with a Mach number of  $M_\infty = 0.3$ . The flow direction is set to  $\theta = \arctan(1/2)$ . A vortex with parameters:  $\epsilon = 1$ ,  $r_c = 1$ , is superimposed to the mean flow. The domain size is  $0 \leq x, y \leq 10$  (i.e.  $L = 10$ ), and the vortex is initially located at the domain center  $(x_0, y_0) = (5, 5)$ . Periodic boundary conditions are applied in both  $x$  and  $y$  directions.

Figure (5) shows a computational mesh with 700 cells. The mesh is composed by two parts: an inner part with a radius of 2 which can rotate; a fixed outer part which takes the rest of the computational domain. Three meshes with 180, 700 and 2731 cells have been used for accuracy tests. For all three cases, the inner part is set to rotate at an angular speed of  $\omega = 1.0$ .

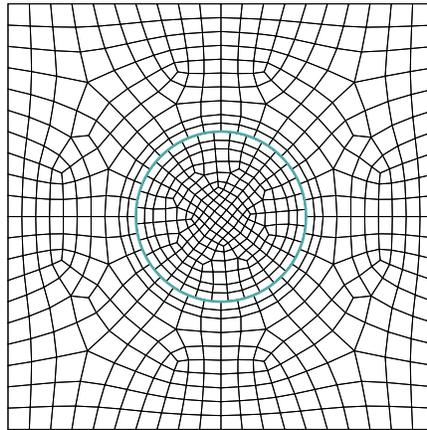


Figure 5. Mesh with 700 cells at a time instant for Euler vortex flow simulation (blue circle indicates sliding interface).

Figure (6) compares density contour of the exact solution and that from fourth order sliding-mesh FR/CPR method on the finest mesh at  $t = 2$ . At this time instant, the vortex is traveled to a position with its center

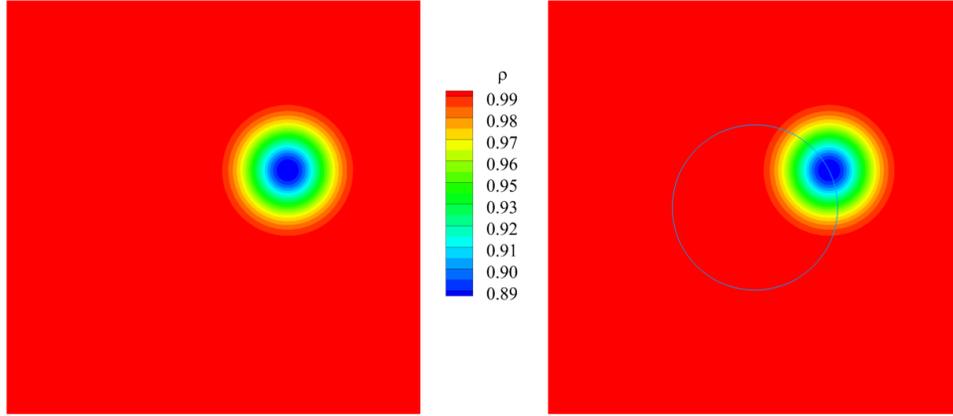


Figure 6. Contour of density at time instant  $t = 2$ . Left, exact solution; right, numerical solution from  $4^{th}$  order scheme (blue circle indicates location of sliding interface).

right on the sliding interface. As we can see, the solver resolves the vortex very well, and we see no visible difference between the exact solution and the numerical one.

Further more, Table 1 and Table 2 give the spatial accuracy of the scheme, where the  $L_1$  and  $L_2$  errors are computed from density at  $t = 2$  when vortex center is traveled right onto the sliding interface. From the two tables we see that the sliding-mesh FR/CPR method gives very reasonable order of accuracy.

cells	L1 error	order	L2 error	order
180	1.34E-4	-	3.16E-4	-
700	1.78E-5	2.97	4.32E-5	2.93
2731	2.60E-6	2.90	6.69E-6	2.84

Table 1. Error and order of accuracy of the  $3^{rd}$  order scheme on Euler vortex flow.

cells	L1 error	order	L2 error	order
180	2.73E-5	-	5.26E-5	-
700	1.74E-6	4.05	3.36E-6	4.05
2731	1.30E-7	3.93	2.93E-7	3.81

Table 2. Error and order of accuracy of the  $4^{th}$  order scheme on Euler vortex flow.

To see how efficient the sliding-mesh FR/CPR method is, we compare the total computational time and the communication time on the sliding interface in Table 3 and Table 4 for third and fourth order schemes, respectively. Times in both tables are collected for 100 computational steps and do not include any post-processing time. It is seen that for all test cases, communication on the sliding interface takes only a few percent of the total computational time, which clearly shows that the sliding-mesh FR/CPR method is efficient. What is interesting is that the relative communication time (represented by the percentage) decreases as either number of cells or order of schemes increases. This is due to the fact that cells are one dimension higher than faces: when perform a mesh refinement, the total number of faces in the domain grows faster than on the sliding interface; when increase scheme order, the total number of degrees of freedom in the domain also grows faster than on the sliding interface.

#### IV.B. Taylor-Couette flow

To test the order accuracy on viscous flow, we use Taylor-Couette flow as the test case. Previous researchers<sup>39,44</sup> used similar flows to test the accuracy of their solvers. In the present test, the inner cylinder has a radius of  $r_i = 1$ , the outer cylinder has radius of  $r_o = 2$ . Both boundaries are set to be isothermal

cells	total time	comm. time	percentage
180	0.249664	0.017105	6.85%
700	0.894914	0.033392	3.73%
2731	4.492501	0.085210	1.87%

**Table 3. Total computation time and interface communication time (both in seconds) for 100 computational steps using 3<sup>rd</sup> order scheme on Euler vortex flow.**

cells	total time	comm. time	percentage
180	0.391325	0.022380	5.72%
700	1.402065	0.044611	3.18%
2731	7.343970	0.110397	1.50%

**Table 4. Total computation time and interface communication time (both in seconds) for 100 computational steps using 4<sup>th</sup> order scheme on Euler vortex flow.**

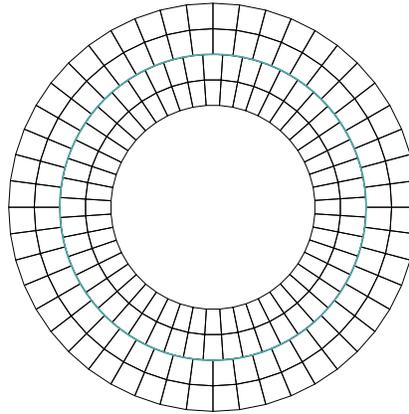
walls. The domain has been divided into two parts at  $r = 1.5$ . The inner part rotates at an angular speed of  $\omega_i = 1$  while the outer part stays stationary. Reynolds number based the inner cylinder radius and velocity is  $Re = 10$ . Mach number on the inner wall is set to be  $Ma_i = 0.1$ . Three meshes with 192, 768 and 3072 cells are used for the tests. Figure 7 shows the mesh with 192 cells.

Figure 8 shows the steady state contours of  $u$  velocity and Mach number from fourth order scheme on the finest mesh. We see that the steady state Mach contours are a series of concentric circles, and the  $u$  velocity is highly symmetric. These results are consistent with our expectations and the analytical solutions.

The exact solution for the circumferential velocity has the following relation to radius  $r$ ,

$$v_\theta = \omega_i r_i \frac{r_o/r - r/r_o}{r_o/r_i - r_i/r_o}, \quad (42)$$

The  $x$  component of this velocity (i.e.  $u$ ) is used to compute the  $L1$  and  $L2$  errors. From Table 5 and Table 6 we see that the sliding-mesh FR/CPR method preserves the high order accuracy for viscous flow as well.



**Figure 7. Mesh with 192 cells at a time instant for Taylor-Couette flow simulation (blue circle indicates sliding interface).**

cells	L1 error	order	L2 error	order
192	9.28E-5	-	1.35E-4	-
768	1.26E-5	2.88	1.78E-5	2.92
3072	1.83E-6	2.83	3.02E-6	2.74

**Table 5. Error and order of accuracy of the 3<sup>rd</sup> order scheme on Taylor-Couette flow.**

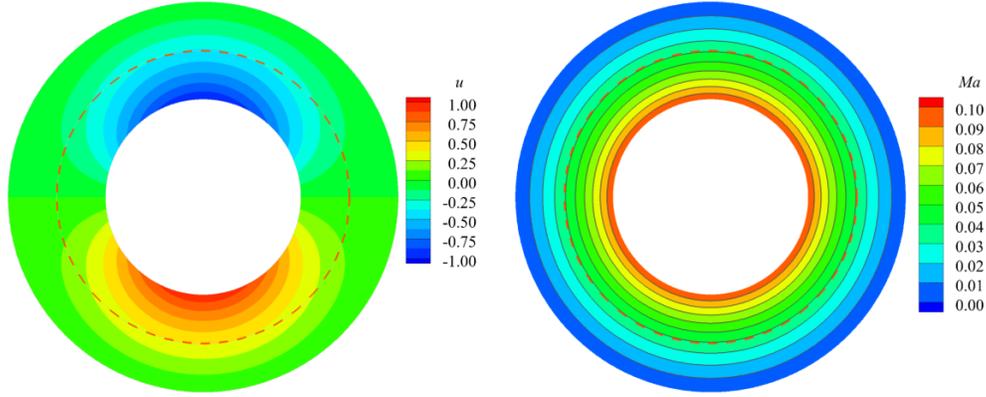


Figure 8. Contours of  $u$  velocity and Mach number (dashed circle indicates location of sliding interface).

cells	L1 error	order	L2 error	order
192	8.79E-6	-	1.30E-5	-
768	6.10E-7	3.85	9.39E-7	3.79
3072	4.81E-8	3.76	7.41E-8	3.73

Table 6. Error and order of accuracy of the 4<sup>th</sup> order scheme on Taylor-Couette flow.

The total computational time and the sliding interface communication time are shown in Table 7 and Table 8 for third and fourth order schemes. Again, data in both tables are collected for 100 computational steps and do not include any post-processing time. It is seen that for viscous flow simulation the sliding-mesh FR/CPR method remains very efficient.

cells	total time	comm. time	percentage
192	0.942448	0.093744	9.93%
768	3.193459	0.217843	6.82%
3072	11.394474	0.468880	4.11%

Table 7. Total computation time and interface communication time (both in seconds) for 100 computational steps using 3<sup>rd</sup> order scheme on Taylor-Couette flow.

cells	total time	comm. time	percentage
192	1.373590	0.123075	8.96%
768	4.611803	0.280074	6.07%
3072	17.745552	0.640689	3.61%

Table 8. Total computation time and interface communication time (both in seconds) for 100 computational steps using 4<sup>th</sup> order scheme on Taylor-Couette flow.

#### IV.C. Flow over a rotating elliptic cylinder

To further verify the solver, we simulate flow over a 2D elliptic cylinder in this section. Maruoka<sup>45</sup> and Zhang et al.<sup>46</sup> studied incompressible flow over a rotating elliptic cylinder using Finite Element and Finite Volume methods respectively. Both studies use Chimera grids for communication between foreground rotating mesh and background stationary mesh. To compare with the incompressible flow results, the freestream Mach number is set to  $Ma = 0.05$  to keep compressibility effects negligible for our sliding-mesh FR/CPR computations.

The major and minor axes are 1.0 and 0.5 for the elliptic cylinder. Initially, the major axis is parallel to freestream. The cylinder rotates counterclockwisely at an angular speed of  $\omega = 0.5\pi$ . The Reynolds number based on freestream velocity and major axis length is 200. Figure 9 shows a schematic of the computational domain. The top and bottom boundary conditions are slippery walls. Dirichlet boundary condition is used for the inlet and fixed pressure is used at the outlet boundary. Finally, isothermal and no-slip conditions are employed for the cylinder wall.

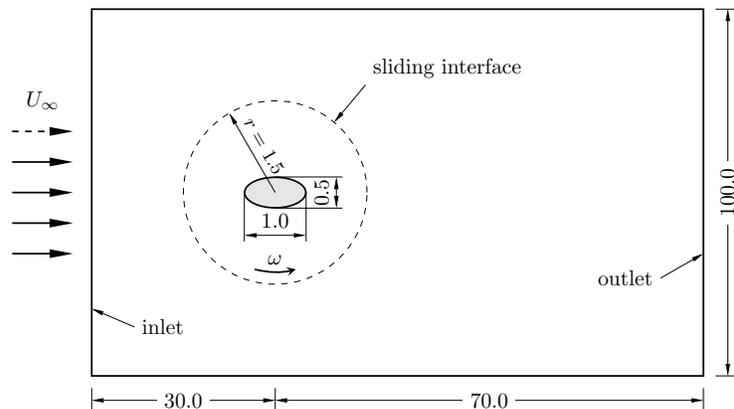


Figure 9. Schematic of the computational domain for flow over a rotating elliptic cylinder (not to scale).

The inner rotating domain has a radius of 1.5 and is meshed with 1280 cells. The rest of the domain is stationary and has 7391 cells in all. Mesh refinement are performed around the leading and trailing edges, and in the wake region. Figure 10 shows part of the mesh around the cylinder and in the wake region a part of the mesh on the left, and mesh close to the cylinder on the right. The first layer of the mesh around the airfoil has a thickness of about 0.005, and the maximum aspect ration is around 2. The non-dimensional time step size  $\Delta t U_\infty / L$  for the simulation is set to  $1.0 \times 10^{-4}$ , where L is the major axis and  $U_\infty$  is the freestream velocity.

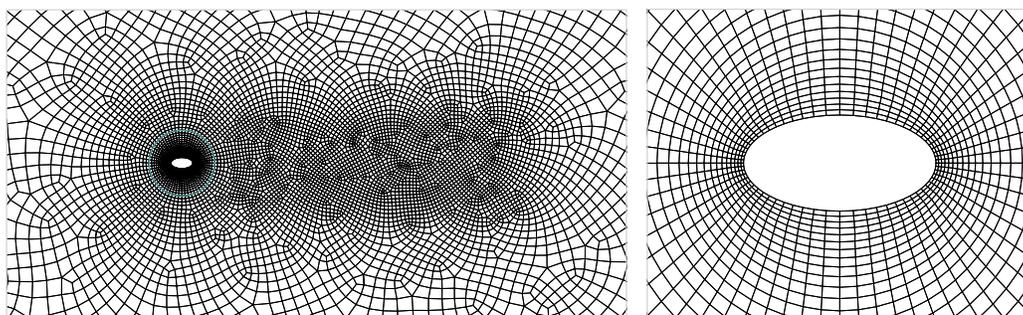


Figure 10. Two local views of mesh around the elliptic cylinder (blue circle indicates sliding interface).

Both third and fourth order schemes were tested for this flow and no visible difference was observed between two solutions. We only present results from the fourth-order scheme. As was noticed by Maruoka<sup>45</sup> and Zhang et al.,<sup>46</sup> the fully developed flow takes a periodic pattern as the cylinder rotates. The lift and drag coefficients in one period are shown in Figure 11. It is seen that the present result agrees very well with published results.

Figure 12 shows the streamlines superimposed on vorticity contours at a series of time in one rotating period. A clockwise vortex and a counterclockwise vortex appear alternatively around two ends of the cylinder. From (h) and (a), we see that a clockwise vortex is formed at the leading edge as the cylinder rotates, and this vortex then sheds off from the leading edge and hits the trailing edge. From time instant (b) to (d), the same clockwise vortex is again shed off from the trailing edge and then convected towards downstream. From time instant (e) to (g), a counterclockwise vortex slowly emerges around the other end of the cylinder and is then convected downstream without reattaching to the cylinder. This process repeats as the cylinder rotates, and a vortex street forms downstream of the cylinder.

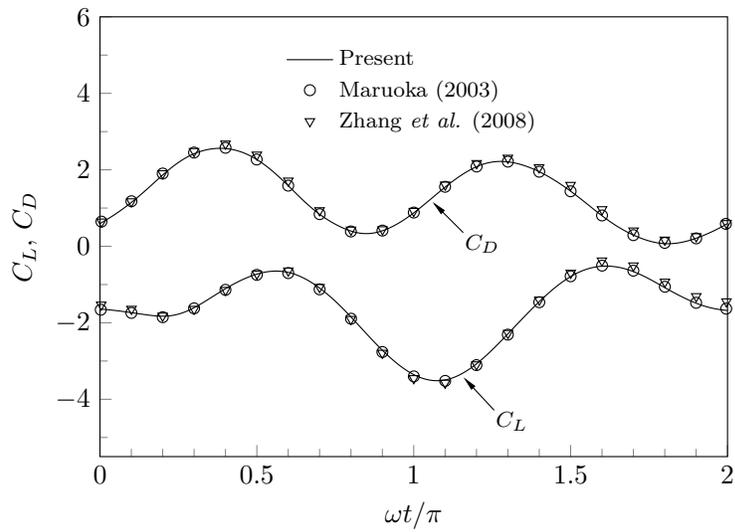


Figure 11. Lift and drag coefficients for flow over an elliptic cylinder.

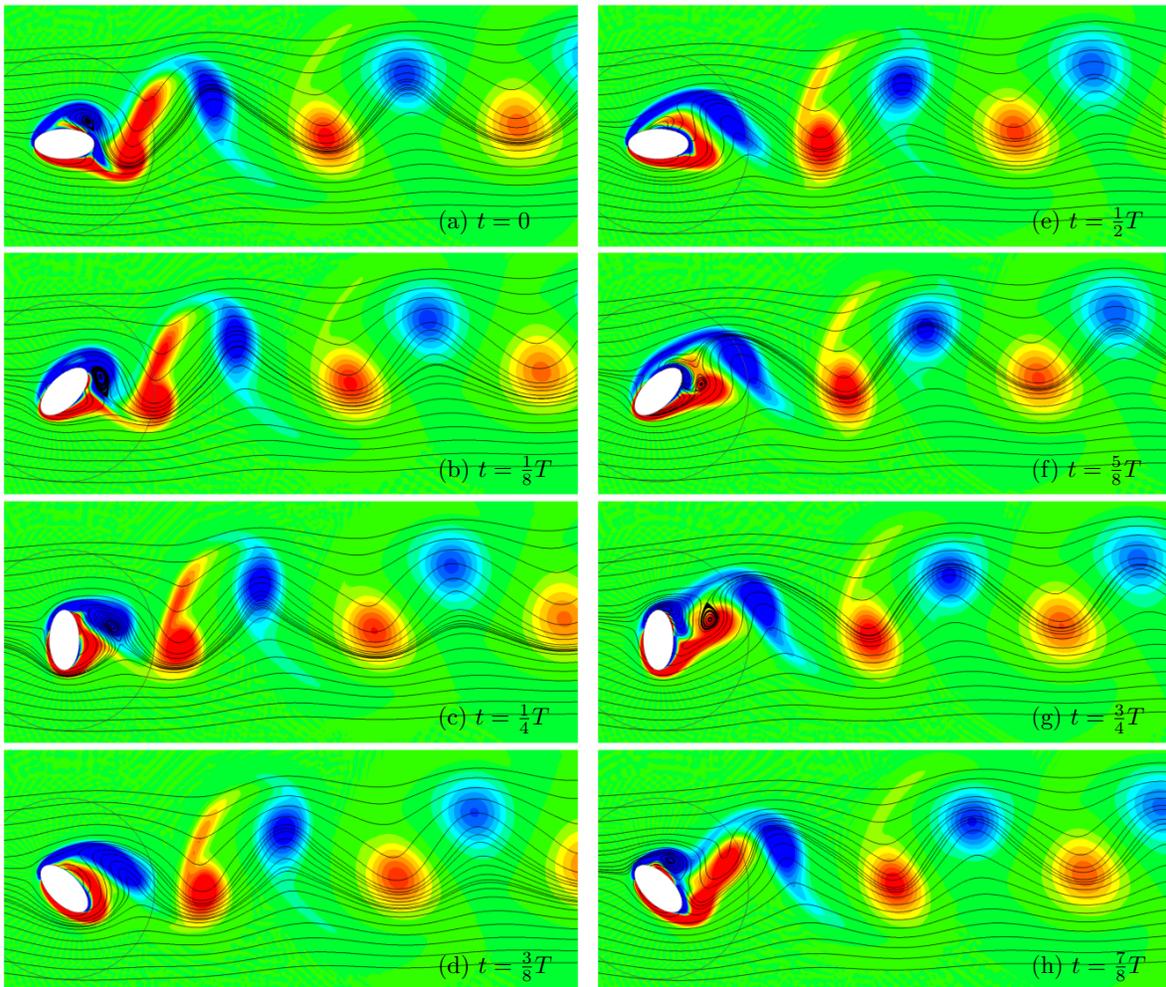


Figure 12. Streamlines and vorticity contours (blue means negative value, red means positive) for flow over a counter-clockwise rotating elliptic cylinder (big circle is sliding interface).

The efficiency of sliding-mesh FR/CPR method for this case is shown in Table 9. The results in the table confirm our previous conclusion that the relative communication time generally decreases as the number of cells or the order of scheme increases. In fact, the interface communication time in the table is almost negligible comparing to the total computation time.

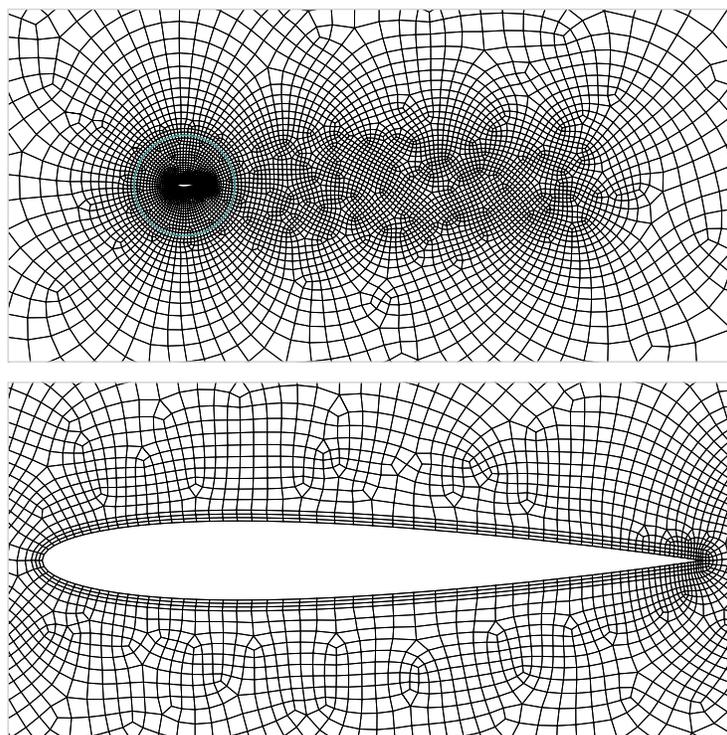
order	total time	comm. time	percentage
3	30.943320	0.155668	0.50%
4	50.437870	0.201519	0.40%

**Table 9.** Total computation time and interface communication time (both in seconds) for 100 computational steps for simulation of flow over an elliptic cylinder.

#### IV.D. Flow over a pitching airfoil

In this last test case, we apply the sliding-mesh FR/CPR method to study flow around a pitching airfoil at various pitching angles. Profile of the airfoil is chosen as NACA0012 with a sharp trailing edge. The airfoil is pitching around a point  $1/3$  chord away from the leading edge on the mean camber. The computational domain has a size of  $100c \times 100c$  with the pitching center located  $30c$  downstream to the inlet, where  $c$  is the chord length. A sliding interface which has a radius of  $2c$  divides the whole domain into two parts: a stationary outer part; a moving inner part which undergoes a rigid body rotation with the airfoil. The motion of the airfoil is given as  $\theta(t) = \alpha \cos(2\pi ft)$ , where  $\theta$  is the angle of attack at time  $t$ ,  $\alpha$  is the pitching angle,  $f$  is the frequency and is set to 0.25. We study three pitching angles,  $\alpha = 30^\circ$ ,  $60^\circ$  and  $90^\circ$ . The freestream has a Mach number of  $Ma_\infty = 0.2$ . Reynolds number based on freestream velocity and airfoil chord length is  $Re = 1000$ .

Figure 13 shows two local views of the unstructured quadrilateral mesh for this case. The inner mesh has a total number of 4600 cells, and the first layer of mesh on the airfoil has a thickness around  $0.004c$ , the maximum aspect ratio there is about 8. The outer domain is meshed into 5059 cells and the maximum cell size is about  $8c$ . Mesh is refined around the leading and trailing edges as well as in the wake region. Fourth-order scheme with a time step size  $\Delta t U_\infty / c = 2.0 \times 10^{-5}$  have been used for all tests.



**Figure 13.** Two local views of mesh around a NACA0012 airfoil (blue circle indicates sliding interface).

IV.D.1. Results for  $\alpha = 30^\circ$

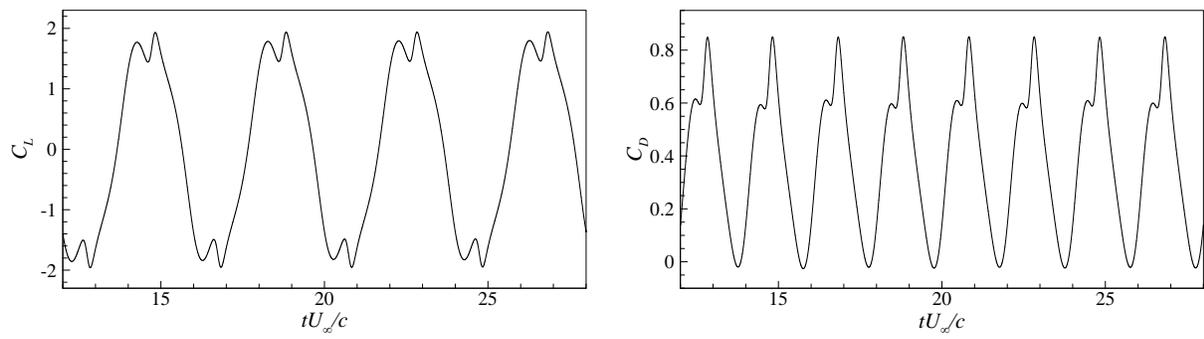


Figure 14. Lift and drag coefficients for NACA0012 airfoil with pitching angle of  $30^\circ$ .

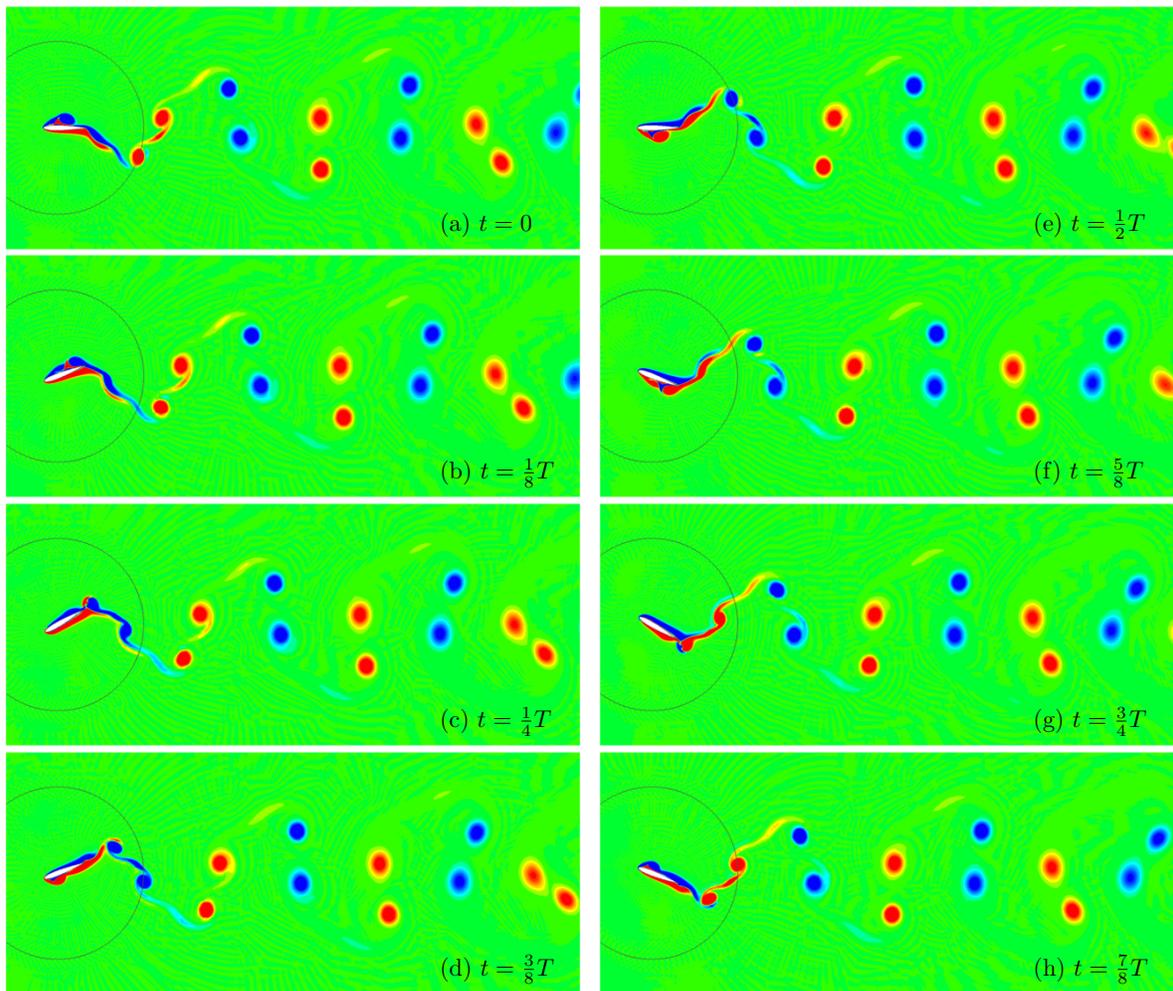


Figure 15. Vorticity contours at different phases for  $\alpha = 30^\circ$  (blue means negative value, red means positive) for flow over a pitching airfoil (big circle is sliding interface).

IV.D.2. Results for  $\alpha = 60^\circ$

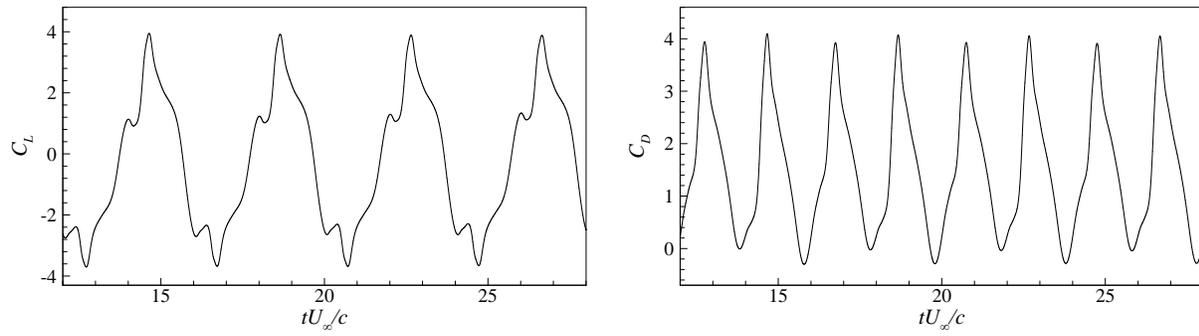


Figure 16. Lift and drag coefficients for NACA0012 airfoil with pitching angle of  $60^\circ$ .

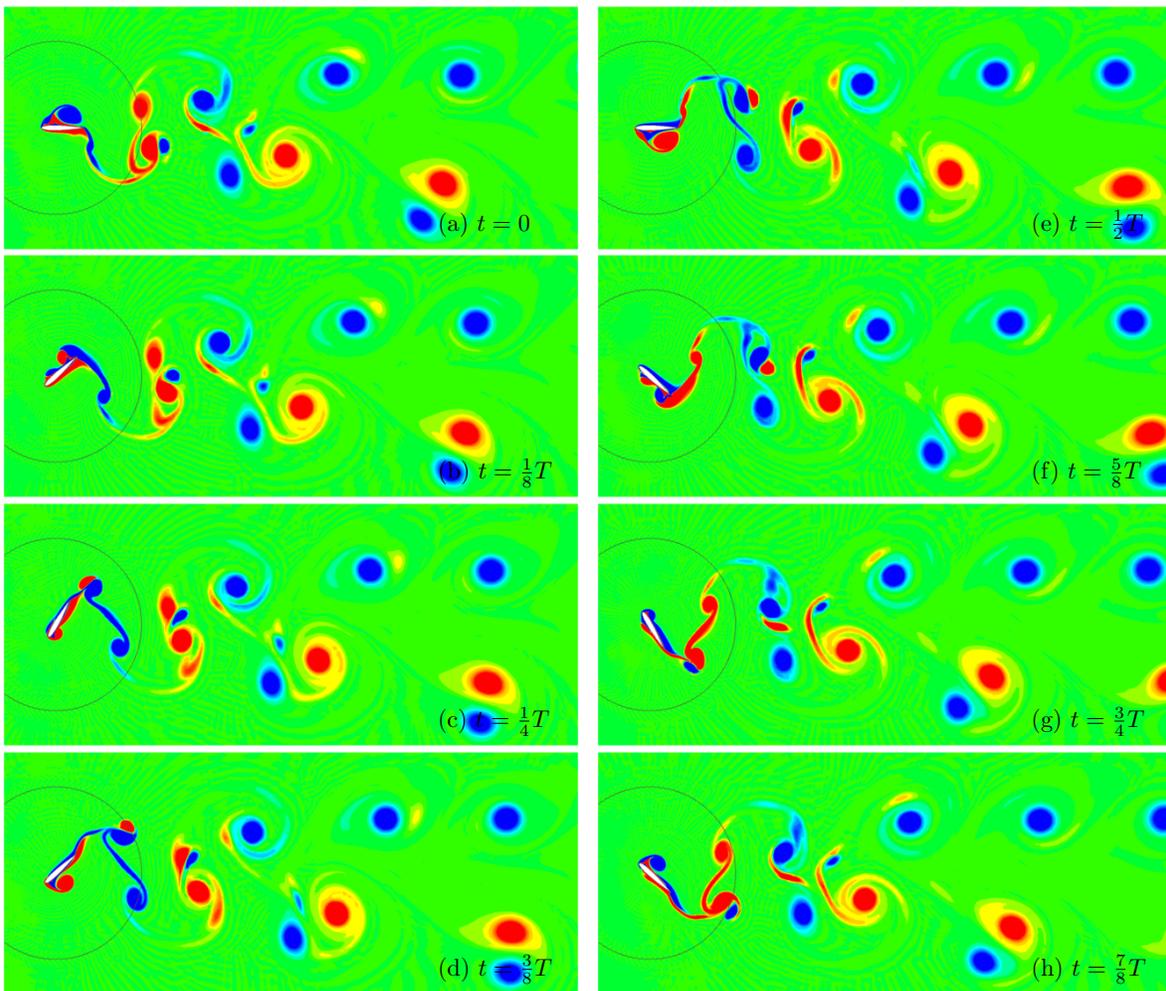


Figure 17. Vorticity contours at different phases for  $\alpha = 60^\circ$  (blue means negative value, red means positive) for flow over a pitching airfoil (big circle is sliding interface).

IV.D.3. Results for  $\alpha = 90^\circ$

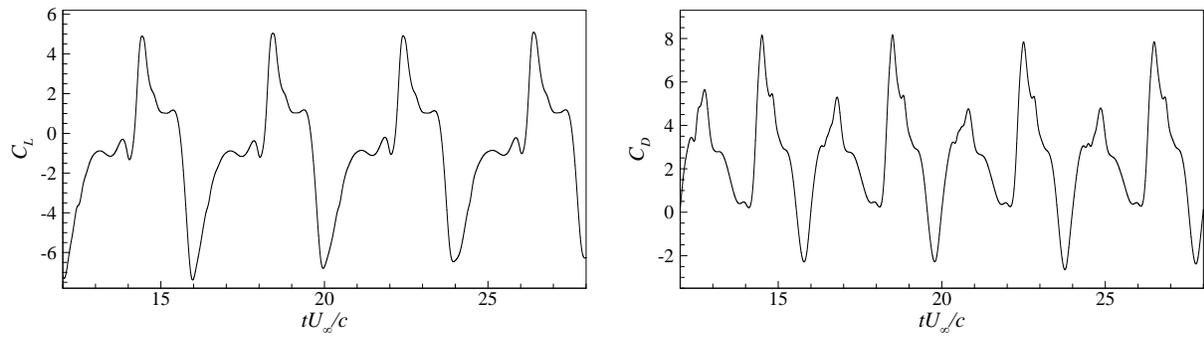


Figure 18. Lift and drag coefficients for NACA0012 airfoil with pitching angle of  $90^\circ$ .

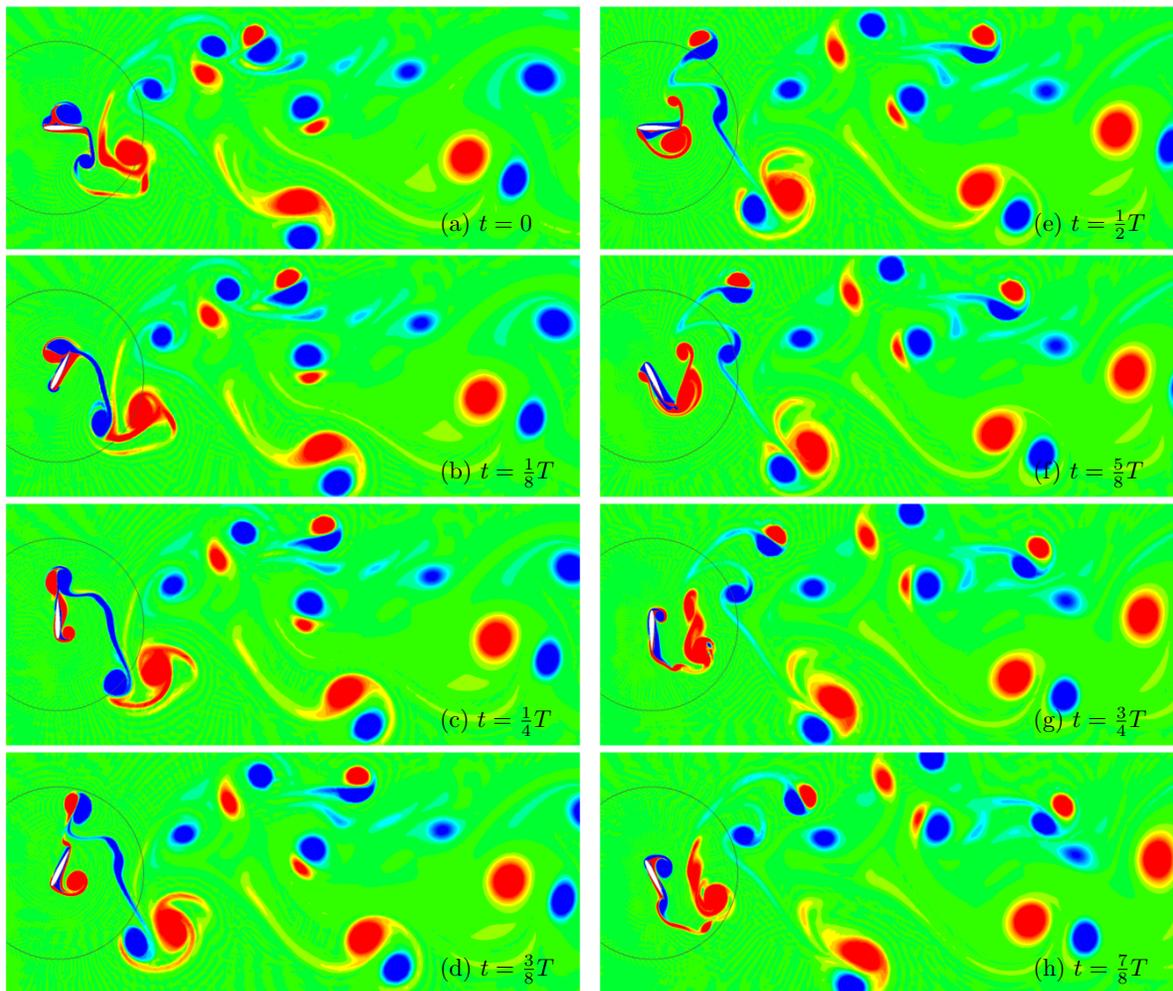


Figure 19. Vorticity contours at different phases for  $\alpha = 90^\circ$  (blue means negative value, red means positive) for flow over a pitching airfoil (big circle is sliding interface).

#### IV.D.4. Summary of results

From the above figures, it is seen that flows at different pitching angles take obviously different patterns, but they also share similarities. From lift and drag coefficients in Figure 14, 16 and 18, we see that all three flows are almost periodic, with their periods close to that of pitching. It's also interesting to notice that for each pitching angle,  $C_D$  changes almost twice as fast as  $C_L$ . Beside the large periodic patterns, the flows also carry structures varying at different frequencies, which are reflected on  $C_L$  and  $C_D$  curves as small peaks. For the amplitude of  $C_L$  and  $C_D$ , it's seen that they both increase as the pitching angle increases. Finally, at all three pitching angles, the airfoil experiences a positive mean drag. For  $\alpha = 30^\circ$  and  $60^\circ$ , the mean lift is close to 0, while at  $\alpha = 90^\circ$ , the airfoil experiences a negative mean lift.

From the vorticity contours in Figure 15, 17 and 19, we see that, when the pitching angle is small ( $\alpha = 30^\circ$ ), flow in the wake region are very organized: the airfoil sheds off a pair of clockwise or counterclockwise vortices alternatively. These vortex pairs are so coherent that the interactions between any two pairs are not obvious. At large pitching angles, the airfoil still sheds off a pair of vortices alternatively, one of them originate for the leading edge, the other from the trailing edge. However, the occurrences of two successive vortex pairs are so close that they interact with each other strongly. These strong interactions between vortex pairs in the wake region make the flow very complicated.

As for all previous cases, we monitored the efficiency of sliding-mesh FR/CPR method for this case. The results are shown in Table 10. Again, as can be seen from the table, the method is very efficient with negligible communication time.

order	total time	comm. time	percentage
3	33.404480	0.246506	0.73%
4	48.823478	0.309867	0.63%

**Table 10. Total computation time and interface communication time (both in seconds) for 100 computational steps for simulation of flow over a pitching airfoil.**

## V. Conclusions

In this paper, a novel, simple, efficient, and high-order accurate sliding-mesh interface method is reported for subsonic compressible flows. The baseline numerical method is the FR/CPR method for spatial discretization. We have successfully designed the sliding-mesh FR/CPR method and test it for several inviscid and viscous flow problems. The method maintains high-order accuracies of FR/CPR method in terms of spatial discretizations for both inviscid and viscous flows using unstructured grids with all quadrilateral elements. This new sliding-mesh interface treatment is very efficient because it introduces negligible extra computational cost to the FR/CPR method for realistic flow simulations. This high-order curved sliding-mesh interface method can also be extended to other discontinuous high-order methods for compressible flows. The curved sliding-mesh interface method will likely see a wide range of future applications, such as simulation of flow around propellers, flapping wing power generator, rotorcraft aerodynamics, etc.

## VI. Acknowledgment

The authors would like to express our acknowledgments for an Office of Naval Research (ONR) support with award number N000141210500 managed by Dr. Ki-Han Kim. Chunlei Liang would like to also acknowledge the support by an ONR Young Investigator Program award administrated by Dr. Ki-Han Kim.

## References

- <sup>1</sup>Mavriplis, C. A., *Nonconforming Discretizations and a Posteriori Error Estimates for Adaptive Spectral Element Techniques*, Ph.D. thesis, M.I.T., February 1989.
- <sup>2</sup>Kopriva, D. A., "A conservative staggered-grid Chebyshev multidomain method for compressible flows. II. A semi-structured method," *Journal of Computational Physics*, Vol. 128, 1996, pp. 475–488.
- <sup>3</sup>Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845.

- <sup>4</sup>Karniadakis, G. E. and Sherwin, S. J., *Spectral/hp Element Methods for Computational Fluid Dynamics*, Oxford University Press, Oxford, 2<sup>nd</sup> ed., 2005.
- <sup>5</sup>Hesthaven, J. S. and Warburton, T., *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Vol. 54 of *Texts in Applied Mathematics*, Springer, New York, 2008.
- <sup>6</sup>Wang, Z. J., editor, *Adaptive High-Order Methods in Computational Fluid Dynamics*, Vol. 2 of *Advances in Computational Fluid Dynamics*, World Scientific, 2011.
- <sup>7</sup>Cockburn, B., Karniadakis, G. E., and Shu, C.-W., editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, Vol. 11 of *Lecture Notes in Computational Science and Engineering*, Springer, New York, 2011.
- <sup>8</sup>Ekaterinaris, J. A., “High-order accurate, low numerical diffusion methods for aerodynamics,” *Progress in Aerospace Sciences*, Vol. 41, No. 3-4, April-May 2005, pp. 192–300.
- <sup>9</sup>Wang, Z. J., “High-order methods for the Euler and Navier Stokes equations on unstructured grids,” *Progress in Aerospace Sciences*, Vol. 43, No. 1-3, 2007, pp. 1–41.
- <sup>10</sup>Reed, W. H. and Hill, T. R., “Triangular mesh methods for the neutron transport equation,” Tech. Rep. LA-UR-73-479; CONF-730414-2, Los Alamos Scientific Laboratory, 1973.
- <sup>11</sup>Cockburn, B., Hou, S., and Shu, C. W., “The Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws IV: The Multidimensional Case,” *Mathematics of Computation*, Vol. 54, No. 190, April 1990, pp. 545–581.
- <sup>12</sup>Cockburn, B. and Shu, C. W., “Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems,” *Journal of Scientific Computing*, Vol. 16, No. 3, September 2001, pp. 173–261.
- <sup>13</sup>F. Bassi and Rebay, S., “High-order accurate discontinuous finite element solution of the 2D Euler equations,” *Journal of Computational Physics*, Vol. 138, No. 2, 1997, pp. 251–285.
- <sup>14</sup>Bassi, F. and Rebay, S., “A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations,” *Journal of Computational Physics*, Vol. 131, 1997, pp. 267–279.
- <sup>15</sup>Jameson, A. and Lodato, G., “A note on the numerical dissipation from high-order discontinuous finite element schemes,” *Computers & Fluids*, Vol. 98, No. 2, September 2014, pp. 186–195.
- <sup>16</sup>Kopriva, D. A. and Koliass, J. H., “A conservative staggered-grid Chebyshev multidomain method for compressible flows,” *Journal of Computational Physics*, Vol. 125, 1996, pp. 244–261.
- <sup>17</sup>Liu, Y., Vinokur, M., and Wang, Z. J., “Spectral difference method for unstructured grids I: Basic formulation,” *Journal of Computational Physics*, Vol. 216, 2006, pp. 780–801.
- <sup>18</sup>Wang, Z. J., Liu, Y., May, G., and Jameson, A., “Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations,” *Journal of Scientific Computing*, Vol. 32, 2007, pp. 45–71.
- <sup>19</sup>Huynh, H., “A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods,” *AIAA paper 2007-4079*, 2007.
- <sup>20</sup>Huynh, H., “A Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin for Diffusion,” *AIAA paper 2009-403*, 2009.
- <sup>21</sup>Huynh, H., Wang, Z., and Vincent, P., “High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids,” *Computers & Fluids*, Vol. 98, 2014, pp. 209–220.
- <sup>22</sup>Wang, Z. J., “Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids: Basic Formulation,” *Journal of Computational Physics*, Vol. 178, 2002, pp. 210–251.
- <sup>23</sup>Wang, Z. J. and Gao, H., “A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids,” *Journal of Computational Physics*, Vol. 228, 2009, pp. 8161–8186.
- <sup>24</sup>Liang, C., Cox, C., and Plesniak, M., “A comparison of computational efficiencies of spectral difference method and correction procedure via reconstruction,” *Journal of Computational Physics*, Vol. 239, 2013, pp. 138–146.
- <sup>25</sup>Yu, M., Wang, Z., and Liu, Y., “On the accuracy and efficiency of discontinuous Galerkin, spectral difference and correction procedure via reconstruction methods,” *Journal of Computational Physics*, Vol. 259, 2014, pp. 1–14.
- <sup>26</sup>Liang, C., Miyaji, K., and Zhang, B., “An efficient correction procedure via reconstruction for simulation of viscous flow on moving and deforming domains,” *Journal of Computational Physics*, Vol. 256, 2014, pp. 55–68.
- <sup>27</sup>Steger, J. L., Dougherty, F. C., and Benek, J. A., “A Chimera grid scheme,” *Advances in Grid Generation*, edited by K. N. Ghia and U. Ghia, Vol. 5 of *ASME FED*, 1983, pp. 59–69.
- <sup>28</sup>Steger, J. L., “The Chimera method of flow simulation,” *Workshop on applied CFD*, Univ. of Tennessee Space Institute, 1991.
- <sup>29</sup>Rai, M., “A conservative treatment of zonal boundaries for Euler equation calculations,” *Journal of Computational Physics*, Vol. 62, 1986, pp. 472–503.
- <sup>30</sup>Rai, M., “A relaxation approach to patched-grid calculations with the Euler equations,” *Journal of Computational Physics*, Vol. 66, 1986, pp. 99–131.
- <sup>31</sup>Behr, M. and Tezduyar, T. E., “Shear-slip mesh update method,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 174, 1999, pp. 261–274.
- <sup>32</sup>Blades, E. L. and Marcum, D. L., “A sliding interface method for unsteady unstructured flow simulations,” *International Journal for Numerical Methods in Fluids*, Vol. 53, No. 3, 2007, pp. 507–529.
- <sup>33</sup>Steijl, R. and Barakos, G., “Sliding mesh algorithm for CFD analysis of helicopter rotor-fuselage aerodynamics,” *International Journal for Numerical Methods in Fluids*, Vol. 58, No. 5, 2008, pp. 527–549.
- <sup>34</sup>McNaughton, J., Afgan, I., Apsley, D. D., Rolfo, S., Stallard, T., and Stansby, P. K., “A simple sliding-mesh interface procedure and its application to the CFD simulation of a tidal-stream turbine,” *International Journal for Numerical Methods in Fluids*, Vol. 74, No. 4, 2014, pp. 250–269.
- <sup>35</sup>Kopriva, D. A., “A Staggered-Grid Multidomain Spectral Method for the Compressible Navier-Stokes Equations,” *Journal of Computational Physics*, Vol. 143, 1998, pp. 125–158.

- <sup>36</sup>Hirt, C. W., Amsden, A. A., and Cook, J. L., “An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds,” *Journal of Computational Physics*, Vol. 14, 1974, pp. 227–253.
- <sup>37</sup>Spiteri, R. J. and Ruuth, S. J., “A new class of optimal high-order strong-stability-preserving time discretization methods,” *SIAM J. Numer. Anal.*, Vol. 40, 2002, pp. 469–491.
- <sup>38</sup>Liang, C., Jameson, A., and Wang, Z. J., “Spectral Difference method for two-dimensional compressible flow on unstructured grids with mixed elements,” *Journal of Computational Physics*, Vol. 228, 2009, pp. 2847–2858.
- <sup>39</sup>Liang, C., Premasuthan, S., and Jameson, A., “High-order accurate simulation of low-Mach laminar flow past two side-by-side cylinders using spectral difference method,” *Computers & Structures*, Vol. 87, 2009, pp. 812–817.
- <sup>40</sup>Miyaji, K., “Vortical flow simulations by a high-order accurate unstructured hexahedral grid method,” *Computers & Fluids*, Vol. 85, 2013, pp. 78–84.
- <sup>41</sup>Rusanov, V. V., “Calculation of interaction of non-steady shock waves with obstacles,” *Journal of Computational and Mathematical Physics USSR*, Vol. 1, 1961, pp. 267–279.
- <sup>42</sup>Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A., *Spectral Methods in Fluid Mechanics*, Springer-Verlag, New York, 1987.
- <sup>43</sup>Erlebacher, G., Hussaini, M. Y., and Shu, C.-W., “Interaction of a shock with a longitudinal vortex,” *Journal of Fluid Mechanics*, Vol. 337, 1997.
- <sup>44</sup>Michalak, C. and Ollivier-Gooch, C., “Unstructured high-order accurate finite-volume solutions of the Navier-Stokes equations,” *AIAA paper 2009-954*, 2009.
- <sup>45</sup>Maruoka, A., “Finite Element Analysis for Flow Around a Rotating Body using Chimera Method,” *International Journal of Computational Fluid Dynamics*, Vol. 17, No. 4, 2003, pp. 289–297.
- <sup>46</sup>Zhang, X., Ni, S., and He, G., “A pressure-correction method and its applications on an unstructured Chimera grid,” *Computers & Fluids*, Vol. 37, No. 2008, 2008, pp. 993–1010.