AIAA SciTech Forum
8–12 January 2018, Kissimmee, Florida
2018 AIAA Aerospace Sciences Meeting

10.2514/6.2018-1094

A Flux Reconstruction Method with Nonuniform Sliding-mesh Interfaces for Simulating Rotating Flows

Bin Zhang[a,*], Zihua Qiu[a,b,†] and Chunlei Liang[a,‡]

[a] *The George Washington University, Washington, DC 20052, USA*

[b] *Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China*

We present a high-order nonuniform sliding-mesh flux reconstruction (FR) method for studying flows about rotating geometries. This new method is an extension of our previous methods that require uniform meshes on a sliding interface.[1–3] It completely eliminates the uniform-mesh restriction and thus significantly simplifies mesh generation. Meanwhile, the nonuniform nature of this method gives the maximum flexibility on mesh points distribution along a sliding interface, which helps to achieve the best mesh resolution across the interface. This extension is also crucial to successfully apply it to complex three-dimensional geometries where the meshes are usually very nonuniform. Numerical tests on both inviscid and viscous flows successfully demonstrate that the present method is able to retain the high-order accuracy of the FR method. Finally, we report our high-fidelity and high-order simulations of flows over rotating cylinders with different cross-section shapes. These simulation results can be used as benchmark test cases for future studies by the CFD community.

I. Introduction

Fluid flows around rotating geometries are very common in engineering applications. For example, flows around marine propellers, wind turbines, helicopter blades, to name just a few. These flows usually carry a lot of vortical structures that are known to have strong dynamic effects on the equipments. Accurate simulation of these flows has been a challenge for decades. There are at least two difficulties: one arises from numerical dissipation that is detrimental to flow structures; the other is how to incorporate complex rotating geometries into a flow solver both accurately and efficiently.

High-order (third and above) numerical methods are becoming more and more popular in recent years due to their substantially smaller numerical dissipation than low-order methods.[4] Some of the most popular high-order methods in computational fluid dynamics include: the discontinuous Galerkin (DG) method,[5,6] the compact finite difference method,[7,8] the spectral element method,[9,10] the spectral volume method[11,12] and the spectral difference (SD) method.[13–17] Among them, the SD method which solves the flow equations in differential form, is one of the most efficient high-order methods that work with unstructured meshes. The SD method has seen a fast development in the past decade,[18–23] as well as a wide range of applications.[24–32] The fact that the stability of the SD method only depends on the location of flux points,[18,19] together with the idea of using higher-degree polynomials to reconstruct fluxes[33] have revolutionized the SD method to an even more efficient method – the flux reconstruction (FR) method (a.k.a. the correction procedure via reconstruction (CPR)).[34–39] Numerical studies confirmed that the FR method is obviously more efficient than the SD method for simulations on both static and dynamic meshes.[40,41] The FR method has even more advantages: it not only recovers the SD method but also the DG method; it even produces new schemes that had never been reported before. For these reasons, the FR method is adopted to handle the first difficulty in the present work.

A couple of methods were developed to overcome the second difficulty. For example, the overset mesh approach,[42] the immersed boundary (IB) method[43] and the arbitrary Lagrangian-Eulerian (ALE) method.[44]

[*] Postdoctoral Scientist in the Department of Mechanical and Aerospace Engineering, bzh@gwu.edu

[†] Ph.D Candidate in the School of Astronautics, visiting student in GWU, qzhnumber1@gmail.com

[‡] Associate Professor in the Department of Mechanical and Aerospace Engineering, AIAA Associate Fellow, chliang@gwu.edu

In the overset mesh method, a foreground mesh enclosing a moving object is overlapped to a background Cartesian mesh. These two sets of meshes are coupled through an extensive number of searches and interpolations, which makes this method expensive and prone to conservation issues. In the IB method, physical boundaries actually do not present. Instead, artificial body forces are used to force wall boundary conditions at cell faces. This method thus can not represent physical boundaries accurately, especially for geometries with curved boundaries. The ALE method is a natural treatment of dynamic meshes under moving frames of reference. It is usually conservative and is very efficient. However, traditional conforming ALE method is limited to small or moderate mesh movements. To deal with freely rotating geometries, nonconforming ALE method, such as sliding-mesh methods are ideal candidates. We previously developed high-order sliding-mesh approaches for the FR and the SD methods.[1–3] In these methods, flow variables and fluxes are projected back and forth between interface mesh elements and curved dynamic mortar elements to ensure conservation and to retain high orders of accuracy. Our methods were shown to be not only accurate but also simple and highly efficient. However, they require mesh elements on both sides of a sliding interface to be of equal size, which has restricted mesh generation especially mesh points distribution.

In this work, we extend our previous methods to be able to deal with sliding interfaces with arbitrarily nonuniform mesh. In this method, a cell face could have arbitrary number of mortar elements instead of only two in our previous methods, and this number could also change with time as the the mesh slides. This extension has greatly simplified mesh generation, as mesh elements on the two sides of a sliding interface do not need to be equal any more. The present method also allows improvement of mesh distribution on sliding interfaces and thus could improve flow filed resolution. We will show that this new method remains simple and high-order accurate. This method is also readily extensible to three-dimensional simulations.

This paper is organized as follows: Section II gives the governing equations. Section III describes the numerical methods, including the FR method and the new sliding-mesh method. Accuracy tests and applications are reported in Section IV. Finally, Section V concludes this paper.

II. Governing Equations

II.A. The Navier-Stokes Equations

The flows of interests are governed by the two-dimensional Navier-Stokes equations that take the following conservative form,

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

where $\mathbf{Q}$ is the vector of conservative variables; $\mathbf{F}$ and $\mathbf{G}$ are the x and the y flux vectors, respectively. Their expressions are

$$\mathbf{Q} = [\rho \ \rho u \ \rho v \ E]^{\mathsf{T}}, \tag{2}$$

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

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

where ρ denotes density; u and v are the x and the y velocity components, respectively; E is the total energy per volume defined as

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2), \tag{5}$$

and p is pressure, γ is the ratio of specific heats and is set to 1.4 (i.e., the value for ideal gas).

As can be seen from Eq. (3) and Eq. (4), the flux vectors have been decomposed into inviscid and viscous parts. This allows independent treatment of the two parts. The inviscid fluxes are only functions of the conservative variables, whose expressions are

$$\mathbf{F}_{inv} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{bmatrix}, \ \mathbf{G}_{inv} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{bmatrix}. \tag{6}$$

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

$$\mathbf{F}_{vis} = -\begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ u\tau_{xx} + v\tau_{yx} + \mathrm{k}T_x \end{bmatrix}, \ \mathbf{G}_{vis} = -\begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + \mathrm{k}T_y \end{bmatrix}, \tag{7}$$

where τ_{ij} is the shear stress tensor which is related to velocity gradients through the following constitutive equation for Newtonian fluids,

$$\tau_{ij} = \mu(u_{i,j} + u_{j,i}) + \lambda\delta_{ij}u_{k,k}, \tag{8}$$

and μ is the dynamic viscosity; $\lambda = -2/3\mu$ based on the Stokes' hypothesis; δ_{ij} is the Kronecker delta; k is the thermal conductivity; T represents temperature which is related to density and pressure through the ideal gas law (a.k.a., equation of state),

$$p = \rho RT, \tag{9}$$

where R is the gas constant. It is worth noting that, in a wide range of thermal dynamic conditions, the nondimensional Prandtl number (defined as $Pr = \mu c_p/\mathrm{k}$, where $c_p = \gamma/(\gamma-1)R$ is the specific heat at constant pressure) is almost a constant for air. In this work, we have set $Pr = 0.72$, from which the thermal conductivity k is obtained.

II.B. The ALE Transformation

We are going to solve the physical equations in an ALE framework. It means that the equations need to be transformed to a computational space that is fixed to a moving frame of reference. This process is depicted in Fig. 1, where a moving physical domain (denoted by $V(t)$) is mapped to a fixed computational domain (denoted by $\widetilde{V}$).

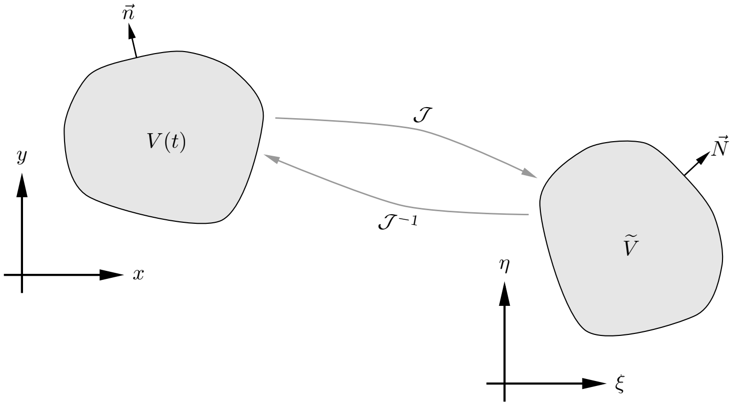


Figure 1. Mapping between a moving physical domain and a fixed computational domain.

Let (t, x, y) represent the physical time and coordinates, and (τ, ξ, η) the computational ones. Assume the mapping can be expressed as $t = \tau$, $x = x(\tau, \xi, \eta)$ and $y = y(\tau, \xi, \eta)$. Then, by using the chain rule of differentiation, it can be shown that the governing equations will take the following conservative form in the computational space,

$$\frac{\partial \widetilde{\mathbf{Q}}}{\partial t} + \frac{\partial \widetilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \widetilde{\mathbf{G}}}{\partial \eta} = 0, \tag{10}$$

and the computational variable and fluxes are related to the physical ones as

$$\begin{pmatrix} \widetilde{\mathbf{Q}} \\ \widetilde{\mathbf{F}} \\ \widetilde{\mathbf{G}} \end{pmatrix} = |\mathcal{J}|\mathcal{J}^{-1} \begin{pmatrix} \mathbf{Q} \\ \mathbf{F} \\ \mathbf{G} \end{pmatrix}, \tag{11}$$

where $|\mathcal{J}|$ is the determinant of the Jacobian matrix for mapping, and $\mathcal{J}^{-1}$ is the inverse Jacobian matrix. Their expressions are

$$|\mathcal{J}| = \left|\frac{\partial(t,x,y)}{\partial(\tau,\xi,\eta)}\right| = \begin{vmatrix} 1 & 0 & 0 \\ x_\tau & x_\xi & x_\eta \\ y_\tau & y_\xi & y_\eta \end{vmatrix} = x_\xi y_\eta - x_\eta y_\xi, \tag{12}$$

$$\mathcal{J}^{-1} = \frac{\partial(\tau,\xi,\eta)}{\partial(t,x,y)} = \begin{bmatrix} 1 & 0 & 0 \\ \xi_t & \xi_x & \xi_y \\ \eta_t & \eta_x & \eta_y \end{bmatrix} = \frac{1}{|\mathcal{J}|}\begin{bmatrix} |\mathcal{J}| & 0 & 0 \\ -x_t y_\eta + y_t x_\eta & y_\eta & -x_\eta \\ x_t y_\xi - y_t x_\xi & -y_\xi & x_\xi \end{bmatrix}, \tag{13}$$

where we have used the adjugate matrix of $\mathcal{J}$ to derive the expression for $\mathcal{J}^{-1}$; (x_t, y_t) represent the motion of the domain, which is prescribed in this study.

It is worth mentioning that in Eq. (11) we have used bold vector symbols to represent the actual scalar components in order to make the expression simpler. Hereinafter, we follow this convention, and a bold symbol could either be a vector or one of its scalar components, whichever makes the operation permissible.

II.C. The Geometric Conservation Law

For moving meshes, to ensure free-stream preservation (i.e., a constant free-stream flow should always stay constant), the following geometric conservation law (GCL)[45] also needs to be considered,

$$\begin{cases} \dfrac{\partial(|\mathcal{J}|\xi_x)}{\partial\xi} + \dfrac{\partial(|\mathcal{J}|\eta_x)}{\partial\eta} = 0, & (14) \\[2ex] \dfrac{\partial(|\mathcal{J}|\xi_y)}{\partial\xi} + \dfrac{\partial(|\mathcal{J}|\eta_y)}{\partial\eta} = 0, & (15) \\[2ex] \dfrac{\partial|\mathcal{J}|}{\partial t} + \dfrac{\partial(|\mathcal{J}|\xi_t)}{\partial\xi} + \dfrac{\partial(|\mathcal{J}|\eta_t)}{\partial\eta} = 0. & (16) \end{cases}$$

The GCL is achieved by substituting a constant flow solution into the governing equations. It is obvious that the GCL is only related to geometrics, and is independent of flow field. The physical meaning of the GCL is more evident if we integrate these equations over a control volume. The first two equations actually means that a closed element must stay closed all the time, and the last equation says that the volume change rate must be equal to the expanding speed of the boundaries.

III. Numerical Methods

This section describes the numerical methods for solving the governing equations, which include the mapping methods for mesh elements, the FR method with a new sliding-mesh method for spatial discretization, and an explicit Runge-Kutta scheme for temporal discretization.

III.A. Iso-parametric and Transfinite Mappings

We map each physical mesh element to a standard computational element (i.e., $0 \le \xi,\ \eta \le 1$) using the following iso-parametric mapping,[46]

$$\begin{pmatrix} x(t,\xi,\eta) \\ y(t,\xi,\eta) \end{pmatrix} = \sum_{i=1}^{K} M_i(\xi,\eta) \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix}, \tag{17}$$

where K is the number of nodes to define a physical element, (x_i, y_i) are the physical coordinates of the i-th node, and M_i is the shape function associated with the i-th node.

Fig. 2 shows schematics of three iso-parametric representations (thin black lines) of a physical curved element (thick gray lines). It is obvious that as the number of nodes increases, an iso-parametric representation approximates the exact shape more closely. For this reason, we use cubic representation (i.e., K=12) along all curved boundaries (except circular boundaries) in this work.

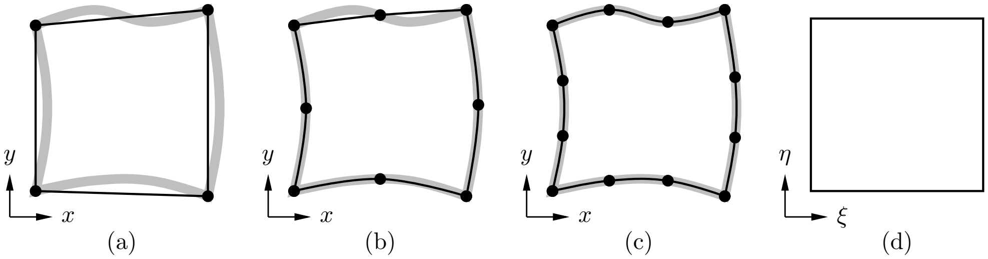


Figure 2. Iso-parametric representations (black lines) of a curved element (wide gray lines): (a) linear representation, $K = 4$, (b) quadratic representation, $K = 8$, (c) cubic representation, $K = 12$, (d) a standard computational element.

To minimize geometric errors along sliding interfaces (where each mesh element has a circular edge), we have adopted the transfinite mapping[47, 48] for cells along sliding boundaries. In transfinite mapping, an element is represented by a direct sum of its boundaries. For example, as shown in Fig. 3, if the faces and nodes of a quadrilateral element are $\vec{X}_{f_i}$ and $\vec{X}_i$, respectively, where $i = 1, 2, 3, 4$, then the transfinite representation of that element is

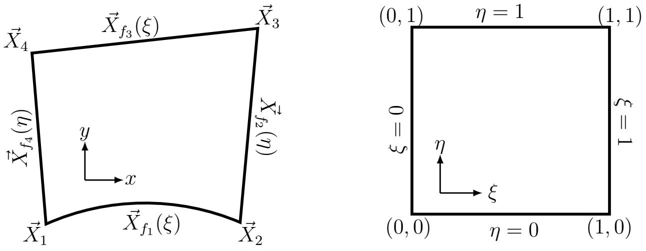


Figure 3. Transfinite mapping of a physical element to a standard square element.

$$\begin{aligned}\vec{X}(\xi,\eta) = (1-\eta)\vec{X}_{f_1}(\xi) + \xi\vec{X}_{f_2}(\eta) + \eta\vec{X}_{f_3}(\xi) + (1-\xi)\vec{X}_{f_4}(\eta) \\ - (1-\xi)(1-\eta)\vec{X}_1 - \xi(1-\eta)\vec{X}_2 - \xi\eta\vec{X}_3 - (1-\xi)\eta\vec{X}_4,\end{aligned} \tag{18}$$

where $\vec{X} = (x, y)$. If a cell face has an exact expression, then the above representation is exact along that face. In our case for a circular cell face (e.g., face $\vec{X}_{f_1}$ in Fig. 3), it can be analytically expressed as

$$\vec{X}_{f_1}(\xi) = \begin{bmatrix} \mathrm{R}\cdot\cos\big((1-\xi)\theta_1 + \xi\theta_2\big) + x_c \\ \mathrm{R}\cdot\sin\big((1-\xi)\theta_1 + \xi\theta_2\big) + y_c \end{bmatrix}, \tag{19}$$

where R and (x_c, y_c) are the radius and the center of that circular face, respectively; θ_1 and θ_2 are the angles of the first and the second nodes. For a cell face whose analytical expression is not available, then $1D$ iso-parametric representation can be used for that face. In fact, the $2D$ iso-parametric mapping in Eq. (17) is a special case of the transfinite mapping, where all cell faces have been approximated by $1D$ iso-parametric representation. The iso-parametric mapping is popular due to its universality and simplicity for implementation. However, when an analytical expression for a boundary is known, the transfinite mapping should be considered if one wants to minimize the geometric errors.

III.B. The FR Method

In contrast to the SD method where two sets of points (i.e., solution points (SPs) and flux points (FPs)) are defined within a standard element, these two sets of points collocate inside an element in the FR method. Therefore, for simplicity, we refer the interior points in the FR method as SPs and those on the boundaries as FPs. Fig. 4 shows a schematic of the distribution of SPs and FPs for a fourth-order FR scheme. Generally, for an N-th order FR method, N SPs are defined along each coordinate direction, and N FPs are required on each boundary face. The collocation of points has avoided the interpolation of solutions from SPs to FPs in the interior of an element, and thus improves computational efficiency.

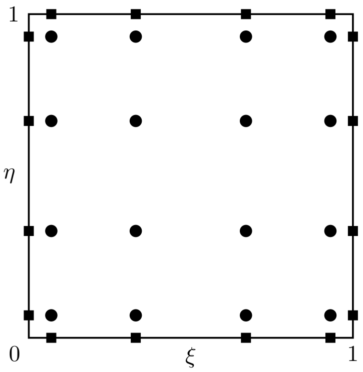


Figure 4. Schematic of the distribution of solution points (circular dots) and flux points (square dots) for a fourth-order FR scheme.

In the present implementation, the SPs and FPs for an N-th order scheme are chosen as the N Legendre points, i.e., the roots of the following N-th degree Legendre polynomial,

$$P_n(X) = \frac{2n-1}{n}(2X-1)P_{n-1}(X) - \frac{n-1}{n}P_{n-2}(X), \quad \text{with } P_{-1}=0,\ P_0=1, \tag{20}$$

where X denotes either ξ or η. If we use X_s to denote the SPs, then the following Lagrange interpolation basis is readily defined at the i-th SP,

$$h_i(X) = \prod_{s=1,s\neq i}^{N} \left(\frac{X-X_s}{X_i-X_s}\right), \tag{21}$$

where X_i is the coordinate of the i-th SP.

The solution and flux polynomials are constructed simply using tensor products of the above interpolation basis. This process can be formulated as

$$\widetilde{\mathbf{Q}}(\xi,\eta) = \sum_{j=1}^{N}\sum_{i=1}^{N} \widetilde{\mathbf{Q}}_{i,j} h_i(\xi) h_j(\eta), \tag{22}$$

$$\widetilde{\mathbf{F}}(\xi,\eta) = \sum_{j=1}^{N}\sum_{i=1}^{N} \widetilde{\mathbf{F}}_{i,j} h_i(\xi) h_j(\eta), \tag{23}$$

$$\widetilde{\mathbf{G}}(\xi,\eta) = \sum_{j=1}^{N}\sum_{i=1}^{N} \widetilde{\mathbf{G}}_{i,j} h_i(\xi) h_j(\eta), \tag{24}$$

where $\widetilde{\mathbf{Q}}_{i,j}$, $\widetilde{\mathbf{F}}_{i,j}$ and $\widetilde{\mathbf{G}}_{i,j}$ are the computational variable and fluxes at the (i,j)-th SP within a standard element, respectively.

These constructed polynomials are only continuous within each element, but discontinuous across element interfaces. To ensure continuity and conservation, common values need to be computed on the interfaces (i.e., element boundaries). At the same time, due to the existence of first-degree spatial derivatives on the fluxes in Eq. (10), the flux polynomials need to be reconstructed to be one-degree higher than the solution polynomial to ensure accuracy.

In the present implementation, the common solution on an interface is taken as the average of the left and the right solutions,

$$\mathbf{Q}^{com} = \frac{1}{2}(\mathbf{Q}^L + \mathbf{Q}^R), \tag{25}$$

where $\mathbf{Q}^L$ and $\mathbf{Q}^R$ are the left and the right solutions, respectively. The common inviscid flux is computed using a Riemann solver, for example, the Rusanov solver,[49]

$$\mathbf{F}_{inv}^{com} = \frac{1}{2}[(\mathbf{F}_{inv}^L + \mathbf{F}_{inv}^R)\mathbf{n} - \lambda(\mathbf{Q}^R - \mathbf{Q}^L)], \tag{26}$$

where $\lambda = |V_n| + c$ is the largest characteristic speed, V_n is the normal velocity, c is the local speed of sound, $\mathbf{F}_{inv}^L$ and $\mathbf{F}_{inv}^R$ are the left and the right discontinuous inviscid fluxes, respectively. The common gradient of solution is also taken as the averaged value,

$$(\nabla \mathbf{Q})^{com} = ((\nabla \mathbf{Q})^L + (\nabla \mathbf{Q})^R)/2. \tag{27}$$

The common viscous flux is then computed from the common solution and common gradient,

$$\mathbf{F}_{vis}^{com} = \mathbf{F}_{vis}(\mathbf{Q}^{com}, (\nabla \mathbf{Q})^{com}). \tag{28}$$

Common fluxes in the other direction, i.e., $\mathbf{G}_{inv}^{com}$ and $\mathbf{G}_{vis}^{com}$ are computed in the same way. These physical common solution and fluxes are converted to computational ones following Eq. (11).

The flux polynomials are reconstructed to be one-degree higher using correction functions. For example, the corrected polynomial $\widetilde{\mathbf{F}}_c$ for $\widetilde{\mathbf{F}}$ in the ξ direction is

$$\widetilde{\mathbf{F}}_c(\xi) = \widetilde{\mathbf{F}}(\xi) + (\widetilde{\mathbf{F}}_L^{com} - \widetilde{\mathbf{F}}_L) \cdot g_L(\xi) + (\widetilde{\mathbf{F}}_R^{com} - \widetilde{\mathbf{F}}_R) \cdot g_R(\xi), \tag{29}$$

where $\widetilde{\mathbf{F}}_L^{com}$ and $\widetilde{\mathbf{F}}_R^{com}$ are the common fluxes at $\xi = 0$ and 1, respectively; $\widetilde{\mathbf{F}}_L = \widetilde{\mathbf{F}}(0)$ and $\widetilde{\mathbf{F}}_R = \widetilde{\mathbf{F}}(1)$ are the original discontinuous fluxes at $\xi = 0$ and 1, respectively; $g_L(\xi)$ and $g_R(\xi)$ are the left and the right correction functions, respectively. These correction functions are polynomials one-degree higher than the original flux polynomials. Furthermore, they are required to satisfy

$$g_L(0) = 1, \quad g_L(1) = 0, \quad g_R(0) = 0, \quad g_R(1) = 1. \tag{30}$$

These conditions ensure that the flux is continuous and takes the common values on cell boundaries. A few correction functions are available,[33, 34] and we use the g_{DG} correction function (which recovers the DG scheme) in this work. Similarly, the same correction is applied to $\widetilde{\mathbf{G}}$ along the η direction, and to $\widetilde{\mathbf{Q}}$ along both directions (for computing the discontinuous gradients, e.g. those on the right hand side of Eq. (27)).

Finally, the residual is computed by substituting the corrected fluxes back into the governing equations,

$$\frac{\partial \widetilde{\mathbf{Q}}}{\partial t} = -\frac{\partial \widetilde{\mathbf{F}}_c}{\partial \xi} - \frac{\partial \widetilde{\mathbf{G}}_c}{\partial \eta}. \tag{31}$$

Details on the time-marching scheme will be discussed in a later section.

III.C. A Nonuniform Sliding-mesh Method

To deal with rotating geometries, we decompose a computational domain into non-overlapping subdomains using sliding interfaces. A subdomain enclosing a rotating geometry can rotate freely with respect to its neighboring subdomain, resulting in nonconforming meshes in between. Fig. 5 shows a schematic of the simplest situation where only one sliding interface presents. We use curved dynamic mortar elements for communication between the two subdomains. As can be seen, a mortar is formed between two successive mesh points from the two sides of an interface. A mortar is always connected to a left and a right cell face, while a cell face may have one or more mortars. These mortar and face connectivities need to be updated at every sub-time-step of a time marching scheme.

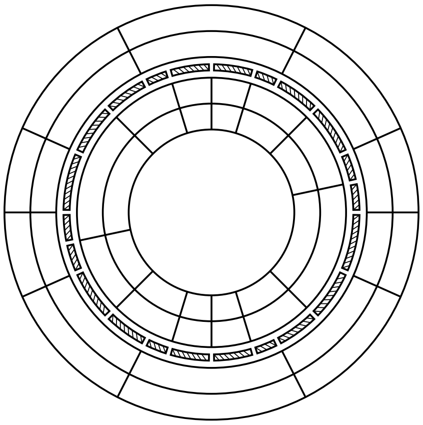


Figure 5. Schematic of the distribution of mortars (hatched) between a rotating mesh and a stationary mesh (inner domain has been scaled to show mortars in the middle).

Before updating connectivities, cell faces on both sides of a sliding interface are reordered counterclockwise and put into a list. This reordering is done during preprocessing, and needs to be done only once. We refer the inner side of an interface as left (denoted by 'l' or 'L'), and the outer side as right (denoted by 'r' or 'R'). If there are nfl cell faces on the left and nfr on the right, then the total number of cell faces is $nf = nfl + nfr$, and the total number of mortars is always $nm = nf$. We define four arrays: $\texttt{vof}(1{:}nf, 1{:}2)$, $\texttt{mof}(1{:}nf, 1{:}2)$, $\texttt{fom}(1{:}nm, 1{:}2)$ and $\texttt{vom}(1{:}nm, 1{:}2)$, to store the vertices of face, mortars of face, faces of mortar and vertices of mortar, respectively. To be more specific, $\texttt{vof}(1{:}nfl, 1{:}2)$ store vertices of faces on the left side, and $\texttt{vof}(nfl + 1{:}nf, 1{:}2)$ store those on the right side; $\texttt{mof}(ifa, 1{:}2)$ store the first mortar and the number of mortars, respectively, for face ifa; $\texttt{fom}(im, 1{:}2)$ store the left and the right face of mortar im, respectively; $\texttt{vom}(im, 1{:}2)$ store the two vertices of mortar im. The detailed procedures for updating these connectivities at every sub-time-step are listed in Algorithm 1.

Algorithm 1 Algorithm for updating mortar and face connectivities

```
mof=0; fom=0; vom=0                                                    ▷ initiate with zeros

$ifl = 1$
for $ifr = (nfl + 1)$ to $nf$ do
    if vof$(ifl, 1)$ lies between vof$(ifr, 1)$ and vof$(ifr, 2)$ then        ▷ the first mortar is located
        Exit
    end if
end for

$im \leftarrow 1$
mof$(ifl, 1) \leftarrow im$                                              ▷ connectivities of the first mortar
mof$(ifl, 2) \leftarrow$ mof$(ifl, 2) + 1$
mof$(ifr, 2) \leftarrow$ mof$(ifr, 2) + 1$
fom$(im, 1) \leftarrow ifl$
fom$(im, 2) \leftarrow ifr$
vom$(im, 1) \leftarrow$ vof$(ifl, 1)$

for $im = 2$ to $nm$ do                                                ▷ connectivities of remaining mortars
    if vof$(ifl, 2)$ lies between vof$(ifr, 1)$ and vof$(ifr, 2)$ then
        $ifl \leftarrow ifl + 1$
        $ifa \leftarrow ifl$
    else
        $ifr \leftarrow ifr + 1$
        if $(ifr > nfl + nfr)$ then
            $ifr \leftarrow ifr - nfr$
        end if
        $ifa \leftarrow ifr$
    end if

    mof$(ifa, 1) \leftarrow im$
    mof$(ifl, 2) \leftarrow$ mof$(ifl, 2) + 1$
    mof$(ifr, 2) \leftarrow$ mof$(ifr, 2) + 1$
    fom$(im, 1) \leftarrow ifl$
    fom$(im, 2) \leftarrow ifr$
    vom$(im, 1) \leftarrow$ vof$(ifa, 1)$
    vom$(im - 1, 2) \leftarrow$ vof$(ifa, 1)$
end for

vom$(nm, 2) \leftarrow$ vom$(1, 1)$                                        ▷ mortars always form a closed loop
```

Since a cell face is mapped to a straight edge (e.g. $0 \le \xi \le 1$) when the underlying cell is mapped to a standard square element, we also map each curved mortar element to a straight one (i.e. $0 \le z \le 1$) in the mortar space using the transfinite mapping. Fig. 6 demonstrates this process, where Ω stands for cell face

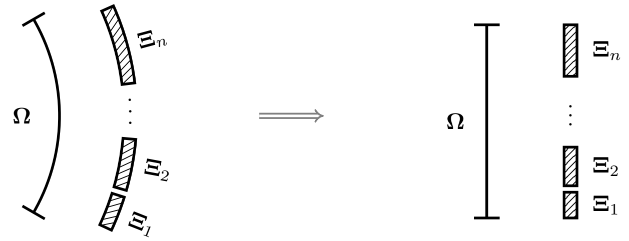


Figure 6. Mapping of curved cell face and mortars to straight ones: left, curved face and mortars in physical domain; right, straight face and mortars in computational domain.

and Ξ for mortar. The mortar space and the computational space are related as

$$\xi = o(t) + s(t)z, \tag{32}$$

where o is the offset of a mortar and s is the scaling, both with respect to the cell face. For example, for the i-th mortar in Fig. 6, we have

$$s_i = L^{\Xi_i}/L^{\Omega},\ o_i = \sum\nolimits_{k=1}^{i-1} s_k, \tag{33}$$

where L^{Ξ_i} is the physical length of the i-th mortar, L^{Ω} is the length of the cell face. It is worth noting that, the offset, scaling and lengths are all time dependent and need to be updated every sub-time-step of a time marching scheme.

From the constructed 2D solution polynomial in Eq. (22), the solution on a cell face is represented by the following one-dimensional polynomial,

$$\mathbf{Q}^{\Omega} = \sum_{i=1}^{N} \mathbf{Q}_i^{\Omega} h_i(\xi), \tag{34}$$

where $\mathbf{Q}_i^{\Omega}$ is the solution at the i-th SP on Ω. Similarly, the solution polynomial on the left side of a mortar is

$$\mathbf{Q}^{\Xi,L} = \sum_{i=1}^{N} \mathbf{Q}_i^{\Xi,L} h_i(z), \tag{35}$$

where $\mathbf{Q}_i^{\Xi,L}$ is the unknown solution at the i-th SP on the left side of mortar Ξ. To compute for $\mathbf{Q}_i^{\Xi,L}$, we require

$$\int_0^1 \left(\mathbf{Q}^{\Xi,L}(z) - \mathbf{Q}^{\Omega}(\xi)\right) h_j(z)dz = 0, \quad j = 1, 2, ..., N. \tag{36}$$

Solution of this equation system when written in matrix form is

$$\mathbf{Q}_{1:N}^{\Xi,L} = \mathbf{P}^{\Omega\to\Xi}\mathbf{Q}_{1:N}^{\Omega} = \mathbf{M}^{-1}\mathbf{S}^{\Omega\to\Xi}\mathbf{Q}_{1:N}^{\Omega}, \tag{37}$$

where $\mathbf{P}^{\Omega\to\Xi}$ is the projection matrix from Ω to Ξ, and the elements of the $\mathbf{M}$ and $\mathbf{S}^{\Omega\to\Xi}$ matrices are

$$M_{i,j} = \int_0^1 h_i(z)h_j(z)dz, \quad i, j = 1, 2, ..., N, \tag{38}$$

$$S_{i,j}^{\Omega\to\Xi} = \int_0^1 h_i(o_i + s_i z)h_j(z)dz, \quad i, j = 1, 2, ..., N. \tag{39}$$

The integrals on the right hand side can be evaluated numerically, for example, using the Clenshaw-Curtis quadrature method. This projection process is demonstrated schematically in Fig. 7(a). In the same way, the right solution $\mathbf{Q}^{\Xi,R}$ can be achieved. The common solution and common inviscid fluxes are computed in the same way as discussed in the previous section for cell interfaces, and they are

$$\mathbf{Q}_{1:N}^{\Xi} = \frac{1}{2}(\mathbf{Q}_{1:N}^{\Xi,L} + \mathbf{Q}_{1:N}^{\Xi,R}), \tag{40}$$

$$(\mathbf{F}^{\Xi}_{inv})_{1:N} = \frac{1}{2}\left[\left((\mathbf{F}^{\Xi,L}_{inv})_{1:N} + (\mathbf{F}^{\Xi,R}_{inv})_{1:N}\right)\mathbf{n} - \lambda(\mathbf{Q}^{\Xi,R}_{1:N} - \mathbf{Q}^{\Xi,L}_{1:N})\right], \tag{41}$$

where $\mathbf{n}$ and λ have the same meanings as those in Eq. (26). The flux polynomials on a cell face and on a mortar (i.e. $\mathbf{F}^{\Omega}_{inv}(\xi)$ and $\mathbf{F}^{\Xi}_{inv}(\xi)$) are constructed in the same way as for the solutions.

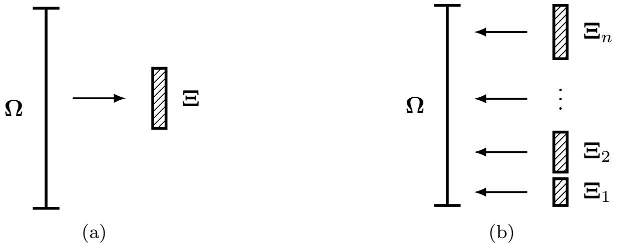


Figure 7. Projection between cell face and mortar: (a) from a left face to the left side of a mortar, (b) from mortars back to the left face.

As shown in Fig. 7(b), to project the common inviscid flux back to the cell face, we require

$$\sum_{i=1}^{n}\int_{o_i}^{o_i+s_i}\left(\mathbf{F}^{\Omega}_{inv}(\xi) - \mathbf{F}^{\Xi_i}_{inv}(z)\right)h_j(\xi)d\xi = 0, \quad j = 1, 2, ..., N. \tag{42}$$

The solutions of this equation system have the following matrix form,

$$(\mathbf{F}^{\Omega}_{inv})_{1:N} = \sum_{i=1}^{n}\mathbf{P}^{\Xi_i\to\Omega}(\mathbf{F}^{\Xi_i}_{inv})_{1:N} = \sum_{i=1}^{n}s_i\mathbf{M}^{-1}\mathbf{S}^{\Xi_i\to\Omega}(\mathbf{F}^{\Xi_i}_{inv})_{1:N}, \tag{43}$$

where the $\mathbf{M}$ matrix is identical to that in Eq. (37), the $\mathbf{S}^{\Xi_i\to\Omega}$ matrix is simply the transposes of $\mathbf{S}^{\Omega\to\Xi_i}$. The common solutions are projected back in the same way as Eq. (42) and are employed to compute local viscous fluxes. These local viscous fluxes are then projected to mortars following Eq. (36). The common viscous fluxes on a mortar are then computed as the average,

$$(\mathbf{F}^{\Xi}_{vis})_{1:N} = \frac{1}{2}\big((\mathbf{F}^{\Xi,L}_{vis})_{1:N} + (\mathbf{F}^{\Xi,R}_{vis})_{1:N}\big), \tag{44}$$

and they are projected back to cell faces following Eq. (42).

III.D. Time Marching and Treatment of GCL

The governing equations can now be written in the following residual form

$$\frac{\partial\widetilde{\mathbf{Q}}}{\partial t} = \mathcal{R}(\mathbf{Q}), \tag{45}$$

where the residual is

$$\mathcal{R} = -\frac{\partial\widetilde{\mathbf{F}}_c}{\partial\xi} - \frac{\partial\widetilde{\mathbf{G}}_c}{\partial\eta}, \tag{46}$$

and $\widetilde{\mathbf{F}}_c$ and $\widetilde{\mathbf{G}}_c$ are the corrected (reconstructed) fluxes. The following explicit Runge-Kutta method is adopted for the time marching,

$$\widetilde{\mathbf{Q}}^{n+1} = \widetilde{\mathbf{Q}}^{n} + \Delta t\sum_{m=1}^{s}b_m k_m, \tag{47}$$

where the intermediate residuals are

$$\begin{aligned} k_1 &= \mathcal{R}\big(t_n, \mathbf{Q}^n\big), \\ k_2 &= \mathcal{R}\big(t_n + c_2\Delta t, \mathbf{Q}^n + \Delta t(a_{21}k_1)\big), \\ &\vdots \\ k_s &= \mathcal{R}\big(t_n + c_s\Delta t, \mathbf{Q}^n + \Delta t(a_{s1}k_1 + a_{s2}k_2 + \cdots + a_{s,s-1}k_{s-1})\big). \end{aligned} \tag{48}$$

In this work, we have used an explicit five-stage forth-order strong stability preserving Runge-Kutta scheme, and the coefficients (i.e. a's and b's) can be found in previous papers.[50, 51]

To numerically satisfy the free-stream preservation condition on dynamic meshes, the GCL equations (14)-(16) need to be discretized using the same temporal and spatial schemes as those for the governing equations. Since the spatial discretization in the FR method is direct differentiation and the geometric variables are computed analytically from the iso-parametric or transfinite mapping, the first two GCL equations are satisfied automatically. However, the geometric variables generally do not satisfy the third GCL equation automatically due to the temporal discretization which is not analytical. To numerically satisfy the third GCL equation, we treat $|\mathcal{J}|$ as an unknown, and use the Runge-Kutta time marching scheme to update it. This numerically obtained $|\mathcal{J}|$ is then used to compute the physical solution (i.e. $\mathbf{Q} = \widetilde{\mathbf{Q}}/|\mathcal{J}|$ according to Eqs. (11) and (13)). In this way, the GCL is numerically satisfied, and free-stream preservation is ensured.

IV. Numerical Tests

We first verify the accuracy on an inviscid flow and a viscous flow. Following that, two comparison studies are carried out: one compares flow over a rotating circular cylinder on a sliding mesh to that on a static mesh with prescribed boundary velocities; the other compares flow over a rotating elliptic cylinder on a sliding mesh to that on a rigid-rotating o-type mesh. Finally, simulations of flows over a rotating triangular cylinder, a rotating square cylinder and two tandem rotating square cylinders are performed.

IV.A. Accuracy on Euler-vortex Flow

In Euler-vortex flow, an isentropic vortex is superimposed to a uniform mean flow and is convected by the mean flow. The flow field in an infinite domain at time t can be analytically described as,

$$u = U_\infty \left\{ \cos\theta - \frac{\epsilon y_r}{r_c} \exp\left(\frac{1 - x_r^2 - y_r^2}{2r_c^2} \right) \right\}, \tag{49}$$

$$v = U_\infty \left\{ \sin\theta + \frac{\epsilon x_r}{r_c} \exp\left(\frac{1 - x_r^2 - y_r^2}{2r_c^2} \right) \right\}, \tag{50}$$

$$\rho = \rho_\infty \left\{ 1 - \frac{(\gamma - 1)(\epsilon M_\infty)^2}{2} \exp\left(\frac{1 - x_r^2 - y_r^2}{r_c^2} \right) \right\}^{\frac{1}{\gamma - 1}}, \tag{51}$$

$$p = p_\infty \left\{ 1 - \frac{(\gamma - 1)(\epsilon M_\infty)^2}{2} \exp\left(\frac{1 - x_r^2 - y_r^2}{r_c^2} \right) \right\}^{\frac{\gamma}{\gamma - 1}}, \tag{52}$$

where U_∞, ρ_∞, p_∞, M_∞ are the mean flow speed, density, pressure and Mach number, respectively; θ is the direction of the mean flow; ϵ and r_c are vortex strength and size, respectively; $(x_r, y_r) = (x - x_0 - \bar{u}t, y - y_0 - \bar{v}t)$ are the relative coordinates; (x_0, y_0) are coordinates of the initial position of the vortex; $(\bar{u}, \bar{v}) = (U_\infty \cos\theta, U_\infty \sin\theta)$ denote the mean velocity.

In the present test, the mean flow is $(U_\infty, \rho_\infty) = (1, 1)$ with a Mach number $M_\infty = 0.3$. The background pressure, p_∞, is computed from ρ_∞ and M_∞. The flow direction is set to $\theta = \arctan(1/2)$. A vortex with $\epsilon = 1$, $r_c = 1$, is superimposed to the mean flow. The domain size is $0 \le x, y \le 10$. The vortex is initially placed at the domain center, i.e., at $(x_0, y_0) = (5, 5)$. Analytical Dirichlet boundary conditions are applied to all boundaries all the time.

Fig. 8 shows two density contours on top of the mesh. The overall domain has been discretized into 192 mesh elements and divided into two subdomains: an inner subdomain with a radius of 2 and 44 elements; an outer one that takes the rest of the domain with 148 elements. The inner subdomain rotates at an angular speed of $\omega = 1$, while the outer subdomain is fixed. Each side of the sliding interface is meshes randomly into 16 nonuniform segments. As can be seen from the contours, this nonconforming sliding interface does not cause any visible alteration to the shape of the vortex, which indicates qualitative correctness of the sliding-mesh method.

We have employed three types of errors to quantitatively measure the accuracy of the solver, and they have the following definitions (taking ρ as an example),

$$L_1 \text{ error} = \frac{\sum_{s=1}^{DOF} |\rho_s - \rho_s^{\text{exact}}|}{DOF}, \tag{53}$$

$$L_2 \text{ error} = \sqrt{\frac{\sum_{s=1}^{DOF}(\rho_s - \rho_s^{\text{exact}})^2}{DOF}}, \tag{54}$$

$$L_\infty \text{ error} = \max_{1\le s\le DOF} |\rho_s - \rho_s^{\text{exact}}| \tag{55}$$

where $DOF = N_{elem} \cdot N^2$ is the total number of degrees of freedom, N_{elem} is the total number of elements, N is the scheme order (i.e., number of SPs in each coordinate direction), ρ_s and ρ_s^{exact} are the numerical and the exact solutions at the SPs, respectively.

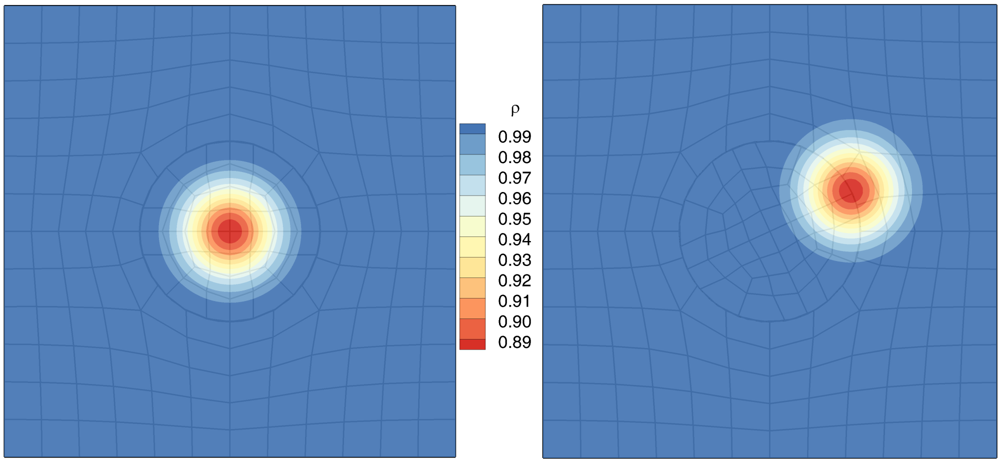


Figure 8. Density contours of the Euler-vortex flow at $t = 0$ (left) and $t = 2$ (right) from the 4^{th}-order scheme.

In Fig. 9, we have plotted the errors (computed from density) against the scheme order. It is seen that, from 2-nd to 10-th orders, all errors decrease exponentially as the scheme order increases, which clearly demonstrates that the nonuniform sliding-mesh method has well retained the high-order accuracy of the FR method on this inviscid flow.

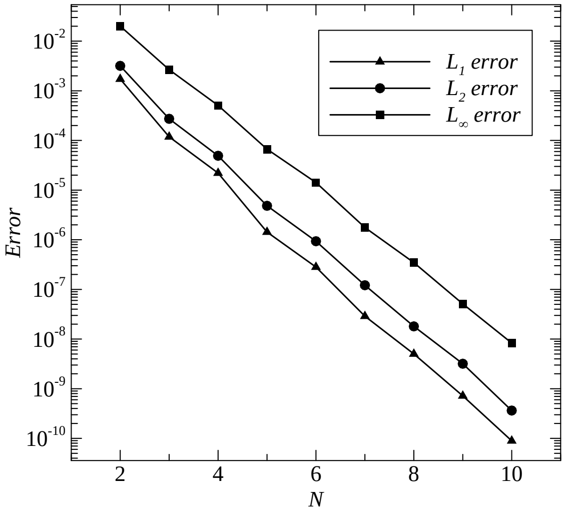


Figure 9. Errors against scheme order on the Euler-vortex flow.

IV.B. Accuracy on Taylor-Couette Flow

Taylor-Couette flow is formed between two concentric rotating circular cylinders. Due to viscous effects, this flow will finally reach a steady state if the Reynolds number is small. The steady-state azimuthal flow speed has the following expression,

$$v_\theta = \omega_i r_i \frac{r_o/r - r/r_o}{r_o/r_i - r_i/r_o} + \omega_o r_o \frac{r/r_i - r_i/r}{r_o/r_i - r_i/r_o}, \tag{56}$$

where r_i and r_o are the radii of the inner and outer boundaries; ω_i and ω_o are the inner and outer angular velocities. In the present setup, we have $r_i = 1$, $r_o = 2$, $\omega_i = 1$ and $\omega_o = 0$. No-slip isothermal wall boundary conditions are applied to both cylinders. The inner wall has fixed Mach number and pressure, and the Mach number is 0.1. The Reynolds number based on inner wall speed and radius is $Re = 10$.

Fig. 10 shows contours of the converged flow field on top of the mesh. The domain is decomposed into two subdomains by a sliding interface at $r = 1.5$. The inner and the outer subdomains are discretized into 36 and 40 cells, respectively. Mesh points are distributed nonuniformly along the sliding interface. The rotating speed of the inner subdomain is $\omega = \omega_i$. The sliding interface and the cylinders are represented by the transfinite mapping as described in Section III.A. As can be seen, the u-velocity contours shows an antisymmetric pattern, and the Mach contours are simply a series of concentric circles. These patterns agree with the analytical solution. The sliding interface again does not contaminate the flow field at all.

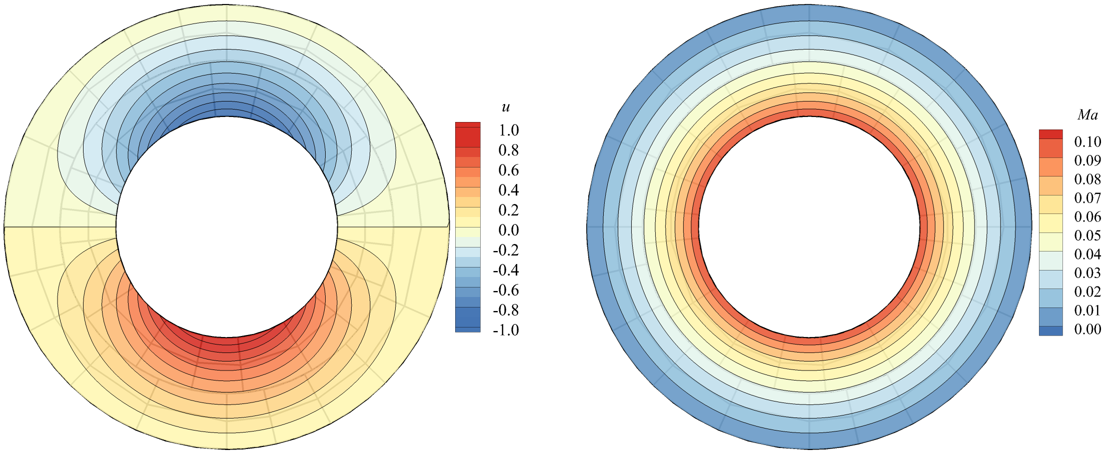


Figure 10. Steady state contours of the Taylor-Couette flow from the 4^{th}-order scheme.

To compute the spatial errors, we continued all simulations to $t = 35$ to make sure that they are well converged. Meanwhile, a time step size of $\Delta t = 1.0 \times 10^{-4}$ was used for all cases to minimize temporal error contributions. The numerical errors are computed based on the steady-state u-velocity, and are plotted against the scheme orders in Fig. 11. Again, the errors are seen to decrease exponentially as the scheme order varies from 2 to 10. Therefore, the sliding-mesh method retains the high-order accuracy of the FR method on this viscous flow as well.

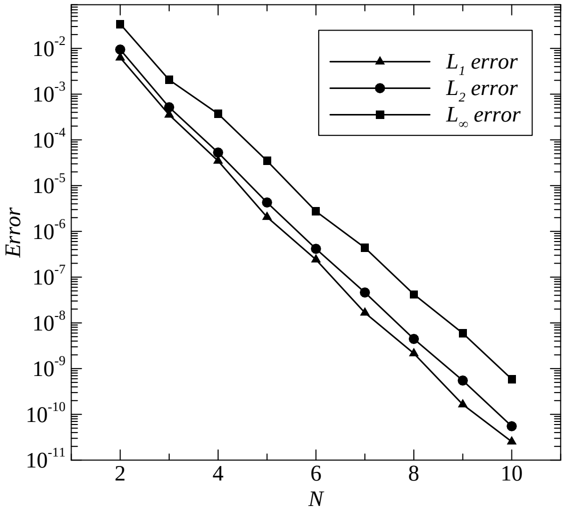


Figure 11. Errors against polynomial order on the Taylor-Couette flow.

IV.C. Comparison study of Flow over a Rotating Circular Cylinder

Due to the isotropic shape, simulation of flow passing a rotating circular cylinder is feasible even on static mesh by simply applying the appropriate velocity boundary conditions. The aim of this test is to verify the solver by comparing flows passing a rotating circular cylinder on a sliding mesh and a static mesh. Ideally, the two configurations should give the same results if the method and solver are correct.

Fig. 12 shows global and local views of the two meshes. Both domains have an overall size of 100 × 100. The cylinder has a diameter of $D = 1$ and is located 30 units away from the inlet. It rotates at an angular speed $\omega_0 D/U_\infty = \pi/2$ (i.e., with a rotating frequency of $f_0 D/U_\infty = 0.25$ and period of $T_0 U_\infty/D = 4$). As can be seen, meshes are refined around the cylinders as well as in the wake region, but coarsened out very quickly towards outer boundaries. The total number of cells is 9,240 for the static mesh, and 8,569 for the sliding mesh, resulting in similar mesh resolution. The minimum mesh spacing is about 0.03, and is found on the first layer of mesh normal to the cylinder surface. The maximum mesh size is about 20, which is found on the outer boundaries. No-slip adiabatic wall boundary condition is applied on the cylinder surface, and characteristic farfield boundary conditions are applied to the outer boundaries. The Reynolds number based on free-stream flow properties and the cylinder diameter is $Re = 100$. The incoming free-stream flow has a Mach number of $Ma = 0.1$.

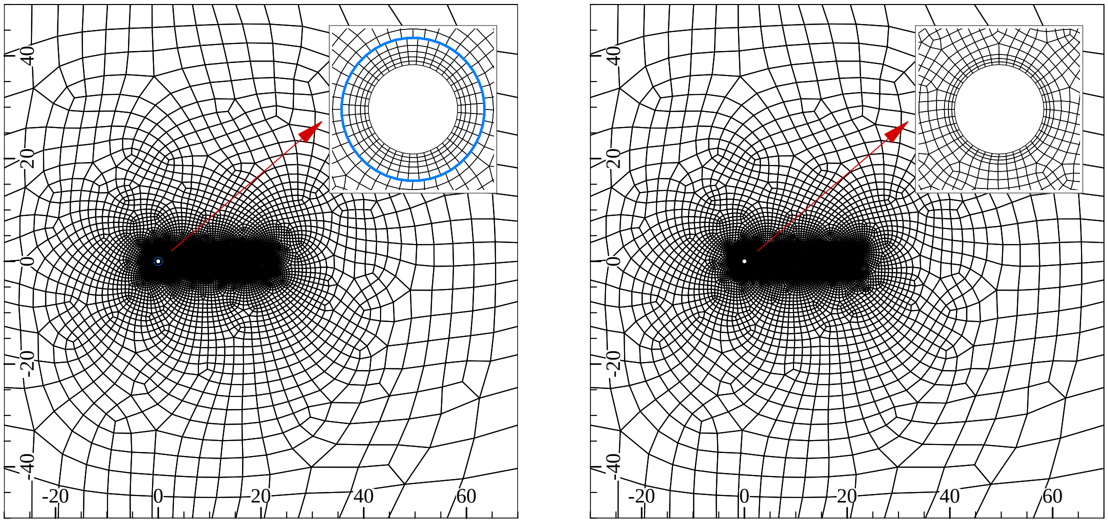


Figure 12. Global and local views of the sliding mesh (blue circle represents sliding interface) and the static mesh (right) for a rotating circular cylinder.

We have tested the 4-th to 6-th order schemes to ensure enough resolution. The strong-stability-preserving five-stage fourth-order Runge-Kutta scheme[50, 51] is used for time marching, and the time step size is set to $\Delta t U_\infty/D = 2.0 \times 10^{-4}$. All simulations are started from a uniform free-stream flow field. In Fig. 13, we compare the transient lift and drag coefficients from the two meshes and from the 4-th and the 6-th order schemes. We see no difference on the curves from the static mesh and the sliding mesh even when the flow still experiences very unsteady development in this process. This clearly demonstrates the correctness of the sliding-mesh method for dealing with rotating geometries. We also see no difference on the results from the 4-th and the 6-th order schemes, which indicates that mesh and scheme independent solutions are obtained. For this reason, we only report results from the 6-th order scheme hereinafter for this test.

The simulations were continued for 50 rotating periods, i.e., to $tU_\infty/D = 200$. The converged lift and drag coefficients from the 6-th order scheme on the sliding mesh are plotted in Fig. 14 from $tU_\infty/D = 160$ to 200. It is seen that the cylinder experiences a positive drag and negative lift all the time. Furthermore, it is very interesting to notice from these curves that the flow seems to be periodic and the period (which is approximately 6) differs from the rotating period of the cylinder (which is $T_0 U_\infty/D = 4$). This is due to the isotropic shape of the circular cylinder which exerts a continuous disturbance instead of a periodic one to the flow.

It is necessary to check if the fully developed flow is indeed periodic or not. This can be achieved by checking, for example, the Lissajous curve of C_L and C_D for multiple periods. If a Lissajous curve repeats itself with time, then the flow is periodic. We plot this curve in Fig. 15 for the sliding mesh case, where

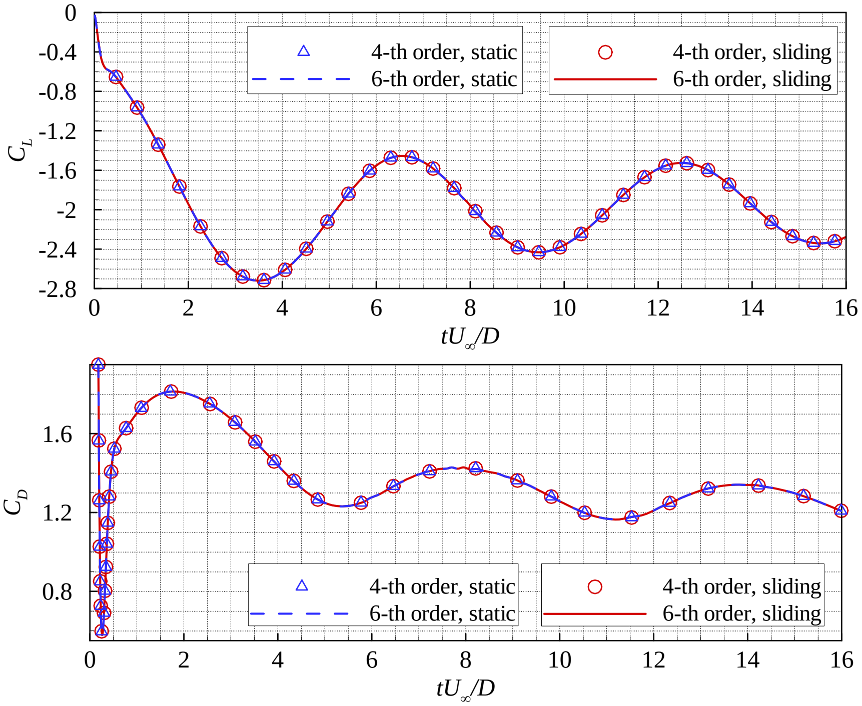


Figure 13. Comparison of transient lift and drag coefficients for a rotating circular cylinder from the 4-th and the 6-th order schemes on a static and a sliding meshes.

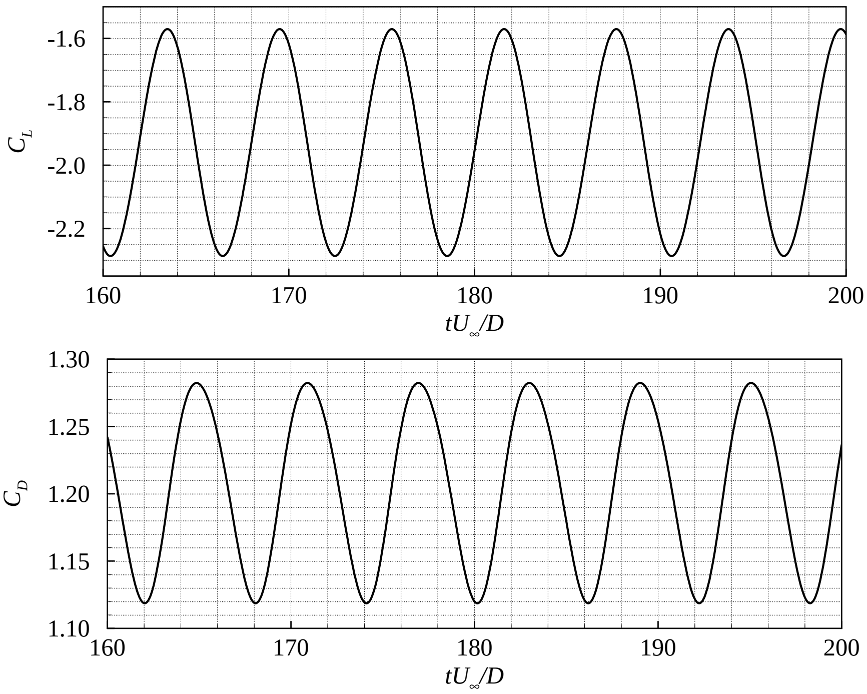


Figure 14. Converged lift and drag coefficients for flow over a rotating circular cylinder.

the curve represents the history from $tU_\infty/D = 120$ to 200, i.e. for 20 rotating periods. We see that the Lissajous curve very well repeats itself, which clearly demonstrates the periodic nature of the flow.

We further decompose the lift and drag coefficients into steady and unsteady components as: $C_L = \bar{C}_L + c_L$, $C_D = \bar{C}_D + c_D$, where $\bar{C}_L$ and $\bar{C}_D$ are the mean coefficients, and c_L and c_D are the time-variant parts. For this flow, the mean coefficients achieved from time averaging are found to be: $\bar{C}_L = -1.930$ and

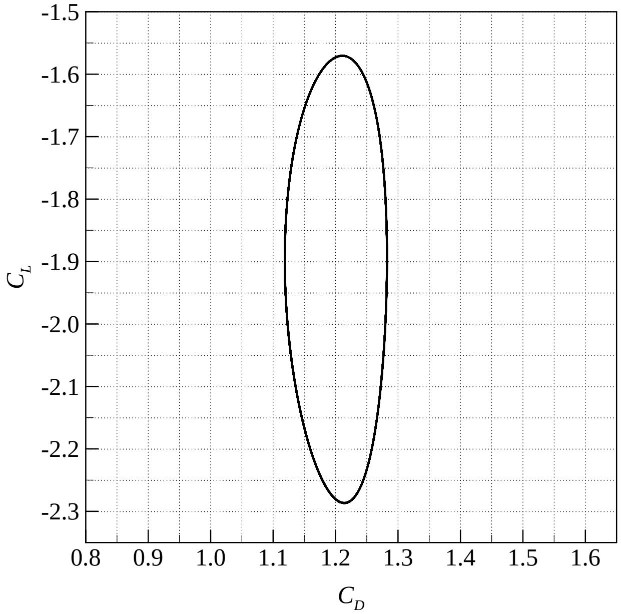


Figure 15. Lissajous curve of C_L and C_D in 20 rotating periods (from tU_∞/D = 120 to 200) for a rotating circular cylinder.

$\bar{C}_D = 1.205$. To calculate the flow period, we compute the power spectrum of c_L and c_D, and the results are plotted in Fig. 16. We see two peaks on each spectrum curve at $fD/U_\infty = 0.1655$ and 0.3310. It is interesting to notice that the latter frequency is twice of the former, but with a much smaller spectrum amplitude. These two frequencies correspond to two periods of $TU_\infty/D = 6.04$ and 3.02, respectively. Therefore, the characteristic period of the flow is $TU_\infty/D = 6.04$, which is consistent with what we observed from the lift and drag coefficient curves in Fig. 14.

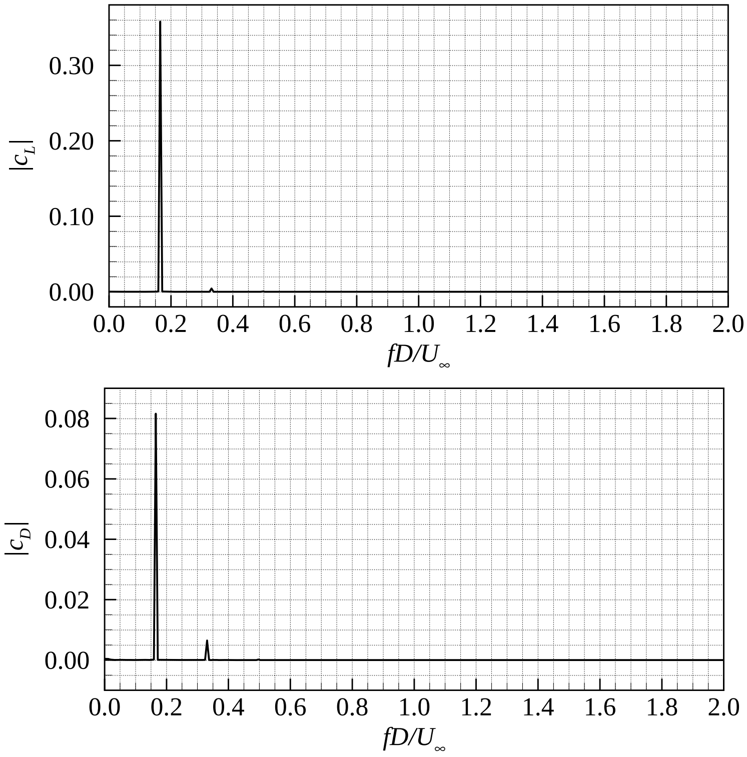
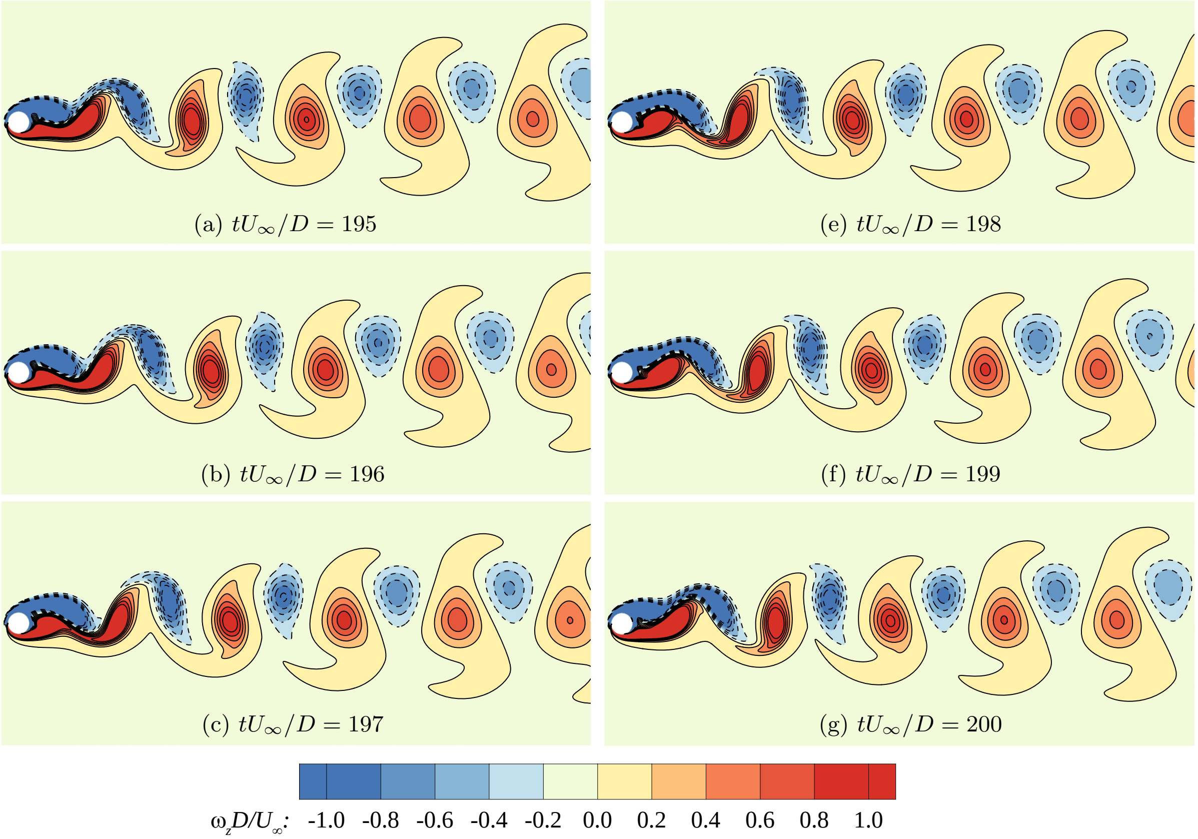


Figure 16. Spectrum of unsteady lift and drag coefficients for flow over a rotating circular cylinder.

In Fig. 17 we compare the flow fields from the two meshes by visualizing vorticity contours at several time instants. As can be seen, contours from the sliding mesh (represented by lines) are visually identical to those from the static mesh (represented by colors). A vortex street consisting of negative and positive vortex pairs is formed behind the cylinder. Due to viscous effects, it is seen that the strength of vortices decays as the flow travels downstream.


(a) $tU_\infty/D = 195$

(e) $tU_\infty/D = 198$

(b) $tU_\infty/D = 196$

(f) $tU_\infty/D = 199$

(c) $tU_\infty/D = 197$

(g) $tU_\infty/D = 200$

Figure 17. Comparison of vorticity contours from sliding mesh (lines) and static mesh (colors) for flow over a rotating circular cylinder.

IV.D. Comparison study of Flow over a Rotating Elliptic Cylinder

In this test, we simulate flow over a rotating elliptic cylinder using a sliding mesh and a rigid-rotating mesh (whole domain rotates). The cylinder has a major axis length of $A = 1.0$ and a minor axis length of $B = 0.5$. It rotates counterclockwise at an angular speed of $\omega_0 A/U_\infty = \pi/2$ (i.e., with a rotating frequency of $f_0 A/U_\infty = 0.25$ and period of $T_0 U_\infty/A = 4$). The freestream flow has a Mach number of $Ma = 0.1$, and a Reynolds number of $Re = 100$ (based on cylinder major axis length and freestream flow properties).

Fig. 18 shows global and local views of the two meshes. The sliding-mesh case has a domain size of 100×100, and the cylinder locates 30 units away from the inlet. The rigid-rotating case has a circular domain with a radius of 50, and the cylinder locates at the center of the domain. The sliding mesh has a total number of 8,769 cells, with 400 within the sliding subdomain whose radius is 0.8. The rigid-rotating mesh has a total number of 9,060 cells. For each case, the cylinder surface is treated as no-slip adiabatic wall, and all other boundaries are treated as characteristic farfield. Three schemes with orders of accuracy from 4 to 6 are tested. The same time step size of $\Delta t U_\infty/A = 2.0 \times 10^{-4}$ is used for all tests.

Fig. 19 compares the transient lift and drag coefficients from the 4-th and 6-th order schemes on the rigid-rotating and the sliding meshes. We see no difference between any two of them, which clearly indicates that nonconforming sliding mesh does not deteriorate the solution compared to a conforming mesh, and the present scheme order and mesh provide enough resolution for producing accurate solutions. For this reason, unless otherwise stated, all results reported hereinafter for this test are from the 6-th order scheme and the sliding mesh.

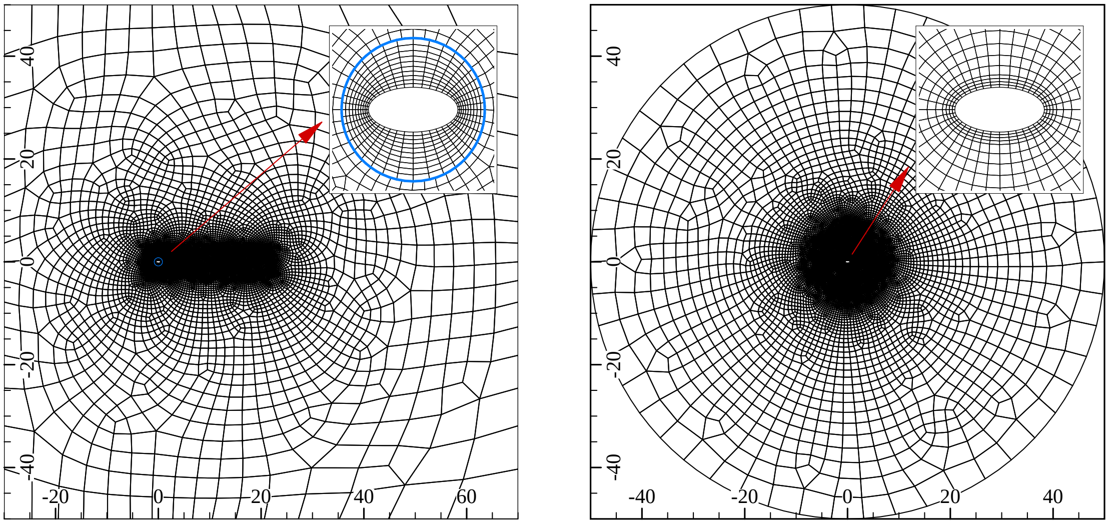


Figure 18. Global and local views of the sliding mesh (blue circle represents sliding interface) and the rigid-rotating mesh (right) for a rotating elliptic cylinder.

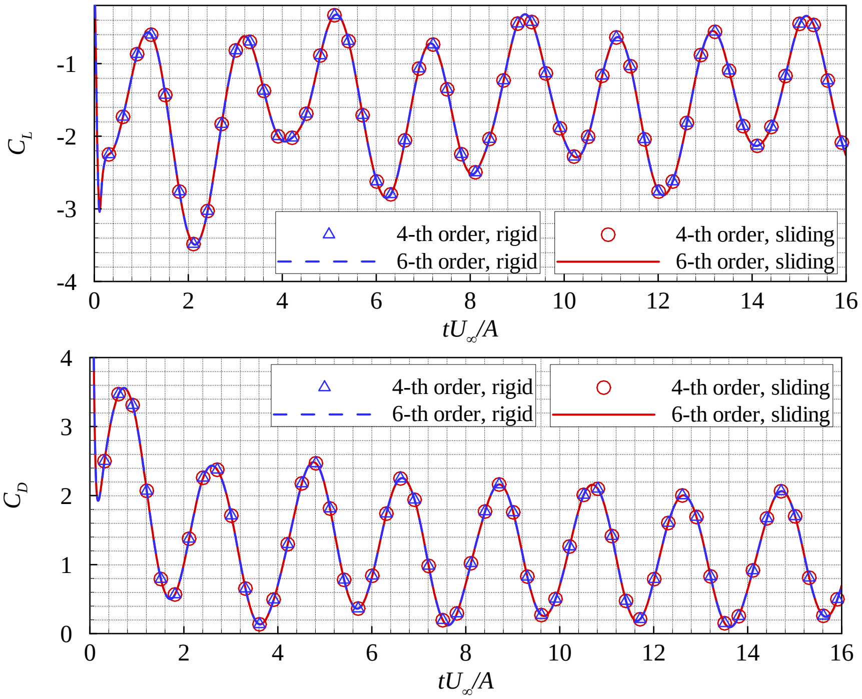


Figure 19. Comparison of transient lift and drag coefficients for a rotating elliptic cylinder from the 4-th and the 6-th order schemes on a rigid-rotating mesh and a sliding mesh.

The converged lift and drag coefficients are plotted in Fig. 20 from $tU_\infty/D = 160$ to 200. It is consistent with the circular cylinder that this elliptic one experiences a negative lift and positive drag all the time. The mean lift and drag coefficients (averaged over 160 time units) are found to be $\bar{C}_L = -1.488$ and $\bar{C}_D = 1.095$, both having smaller magnitudes than those of the circular cylinder. It is also obvious that these curves show a less periodic pattern than those of the circular one. The Lissajous curve of C_L and C_D are plotted in Fig. 21 to check the periodicity of the flow. As can be seen, the curve does not always repeat itself, which indicates a complex frequency distribution in the flow.

The power spectrum of the unsteady lift and drag coefficients (i.e., c_L and c_D) are plotted in Fig. 22. We see five obvious peaks at the frequencies of $fD/U_\infty = 0.1875$, 0.3125, 0.5, 0.6875 and 1.0, respectively. The largest peak is found at $fD/U_\infty = 0.5$, which is induced by the rotating motion that applies a periodic

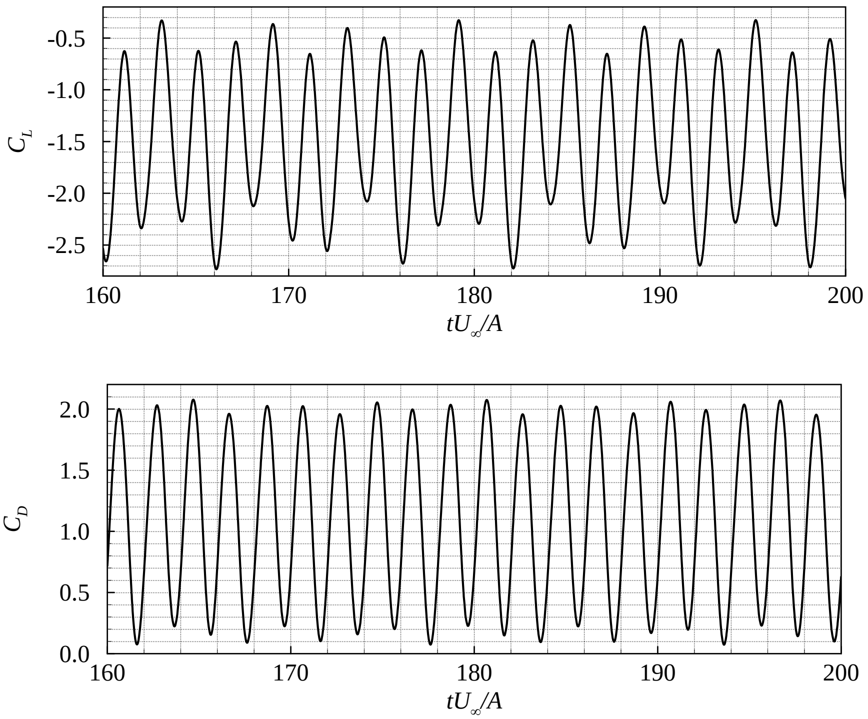


Figure 20. Converged lift and drag coefficients for flow over a rotating elliptic cylinder.

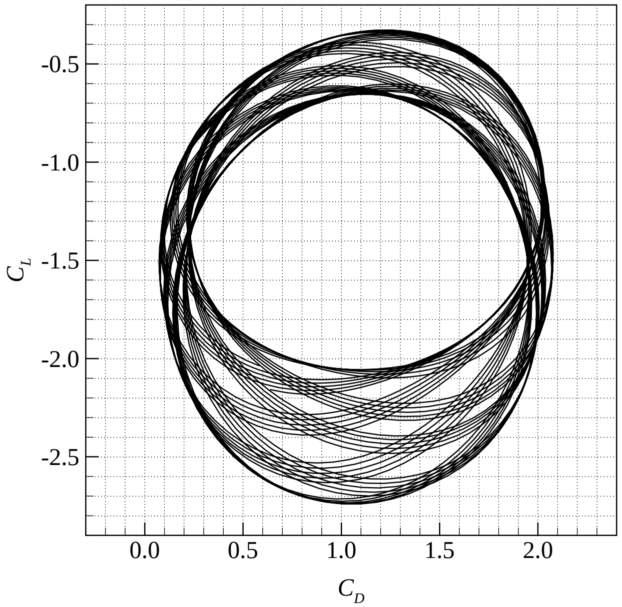


Figure 21. Lissajous curve of C_L and C_D in 20 rotating periods (from $tU_\infty/D = 120$ to 200) for a rotating elliptic cylinder.

disturbance to the flow filed (it's twice of the rotating frequency because of the geometric symmetry of the elliptic cylinder). Therefore, this flow is obviously dominated by the rotating motion.

We plot and compare the flow fields from the sliding mesh and the rigid-rotating mesh in Fig. 23 and Fig. 24. It is evident that the sliding mesh delivers cleaner results than the rigid-rotating mesh as there are fewer wiggles on the contours. This is in fact caused by grid resolution in the wake region. Although the rigid-rotating mesh has slightly more grid points than the sliding-mesh, but the mesh needs to be distributed

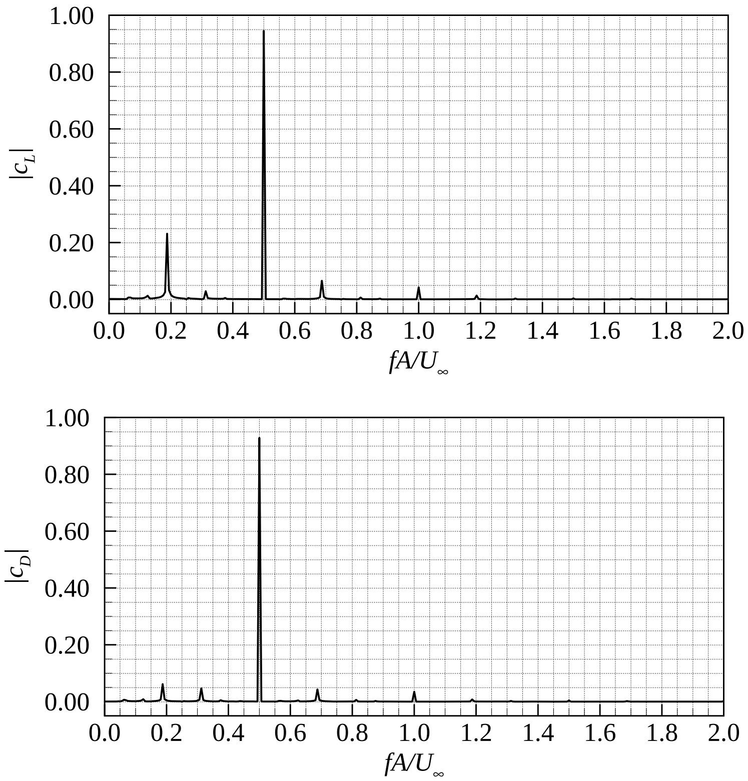
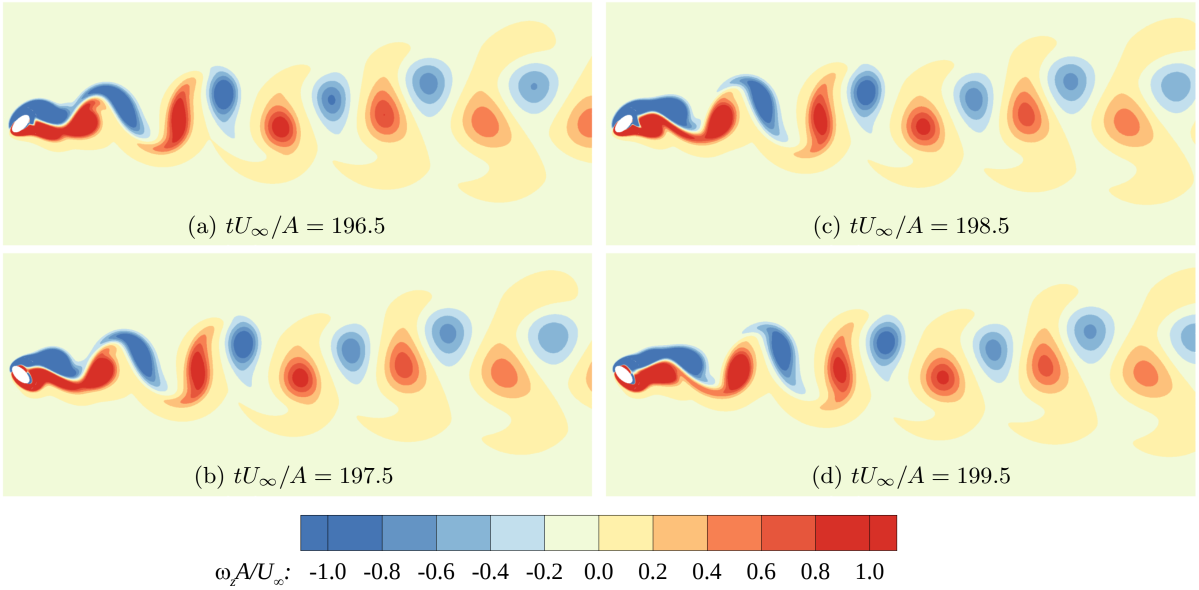


Figure 22. Spectrum of unsteady lift and drag coefficients for flow over a rotating elliptic cylinder.

as uniform as possible in the azimuthal direction around the cylinder to make sure that the wake region always has fair mesh resolution as the overall domain rotates. Therefore, at each time instant, a majority of the grids are wasted in non-wake regions for a rigid-rotating mesh. In contrast, for a sliding mesh, only a small part of the domain rotates, and the wake region is fixed and can be effectively refined. For this reason, a sliding mesh could save a lot of grids compared to a rigid-rotating mesh, and therefore it could potentially reduce the overall computational cost.

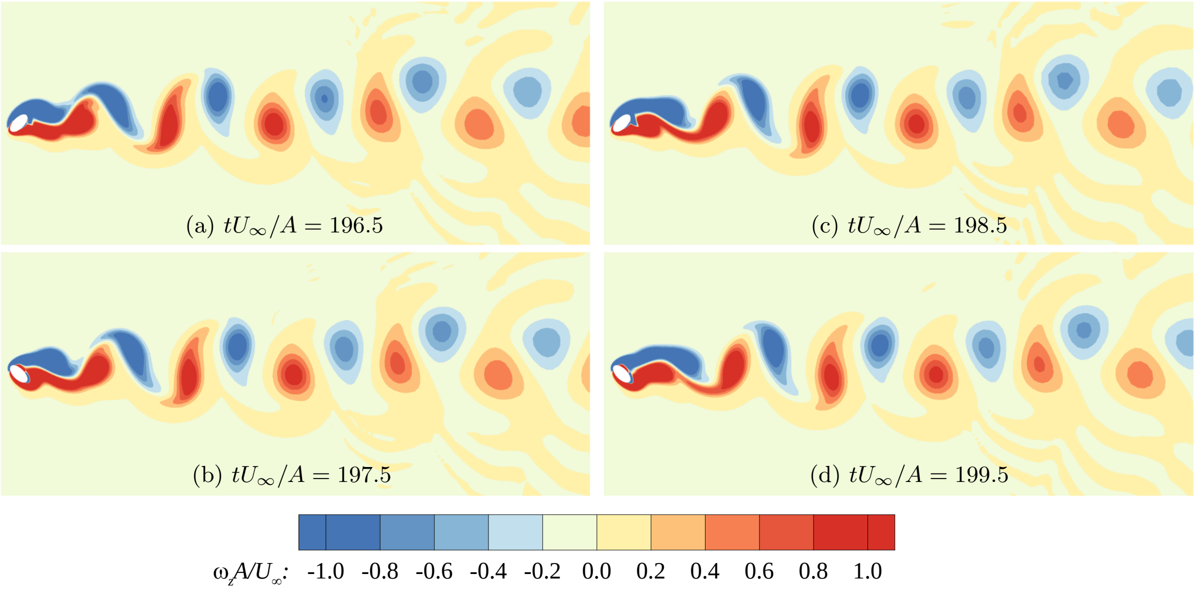

(a) $tU_\infty/A = 196.5$

(c) $tU_\infty/A = 198.5$

(b) $tU_\infty/A = 197.5$

(d) $tU_\infty/A = 199.5$

Figure 23. Vorticity contours for flow over a rotating elliptic cylinder using a sliding mesh.



(a) $tU_\infty/A = 196.5$

(b) $tU_\infty/A = 197.5$

(c) $tU_\infty/A = 198.5$

(d) $tU_\infty/A = 199.5$

Figure 24. Vorticity contours for flow over a rotating elliptic cylinder using a rigid-rotating mesh.

IV.E. Flow over a Rotating Triangular Cylinder

In this test, we study flow over a rotating triangular cylinder using the sliding-mesh solver. The cylinder is equilateral, with an edge length of $A = 1.0$. It rotates counterclockwise around its geometric center again at an angular speed of $\omega_0 A/U_\infty = \pi/2$. The freestream flow has a Mach number of $Ma = 0.1$, and a Reynolds number of $Re = 100$ (based on cylinder edge length and freestream flow properties).

Fig. 25 shows global and local views of the mesh. The overall domain size is 100 × 100, and the cylinder locates 30 units away from the inlet. The domain is discretized into 8,750 cells, with 381 within the rotating subdomain (with a radius of 0.8). The cylinder surface is treated as adiabatic wall, and all other boundaries are treated as characteristic farfield. The time step size for computation is $\Delta t U_\infty/A = 2.0 \times 10^{-4}$. Three schemes with orders of accuracy from 4 to 6 have been tested to ensure mesh- and scheme-independent solution, and all results reported below are from the 6-th order scheme.

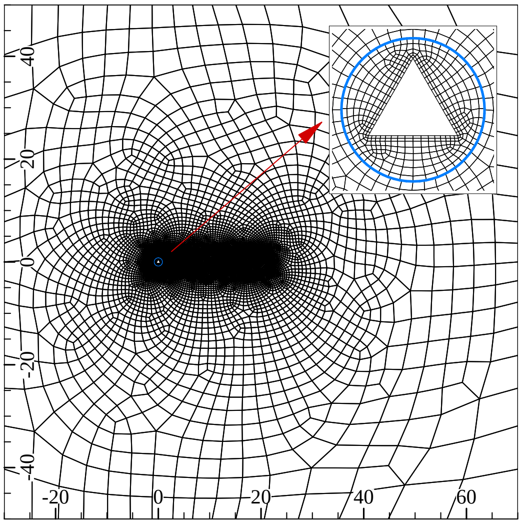


Figure 25. Mesh for a rotating triangular cylinder (blue circle indicates sliding interface).

Fig. 26 plots the time history for C_L and C_D from $tU_\infty/D = 160$ to 200. Similar to the circular cylinder case, this triangular cylinder also experiences an always-negative lift and always-negative drag. The mean values from time averaging over 170 time units are found to be $\bar{C}_L = -2.083$ and $\bar{C}_D = 1.240$. The Lissajous curve for C_L and C_D is plotted in Fig. 27, and it does not always repeat itself, which indicates a complex nature of the flow periodicity.

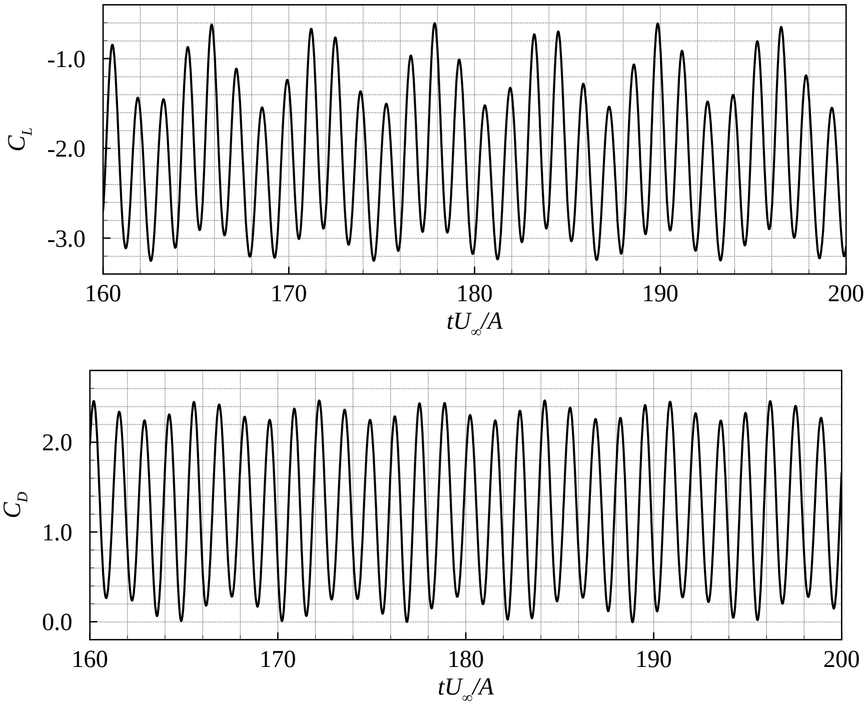


Figure 26. Converged lift and drag coefficients for flow over a rotating triangular cylinder.

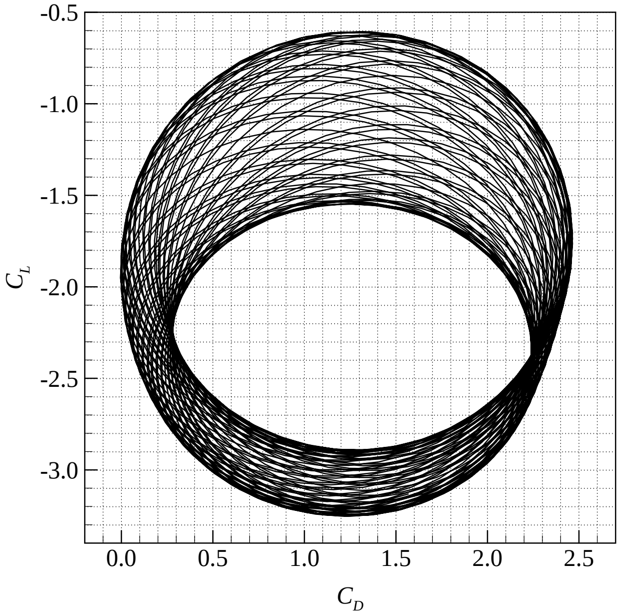


Figure 27. Lissajous curve of C_L and C_D in 20 rotating periods (from $tU_\infty/D = 120$ to 200) for a rotating triangular cylinder.

To see the dominant frequencies in the flow, we have computed the power spectrum of c_L and c_D, and the results are plotted in Fig. 28. The corresponding frequencies of the four peaks are $fD/U_\infty = 0.164$, 0.586, 0.75 and 0.914, respectively. The largest peak is found at the frequency $fD/U_\infty = 0.75$, which is three times the rotating frequency (i.e., $f_0D/U_\infty = 0.25$) due to the geometric symmetry of the three vertices. For this reason, this flow is still mostly dominated by the rotating motion.

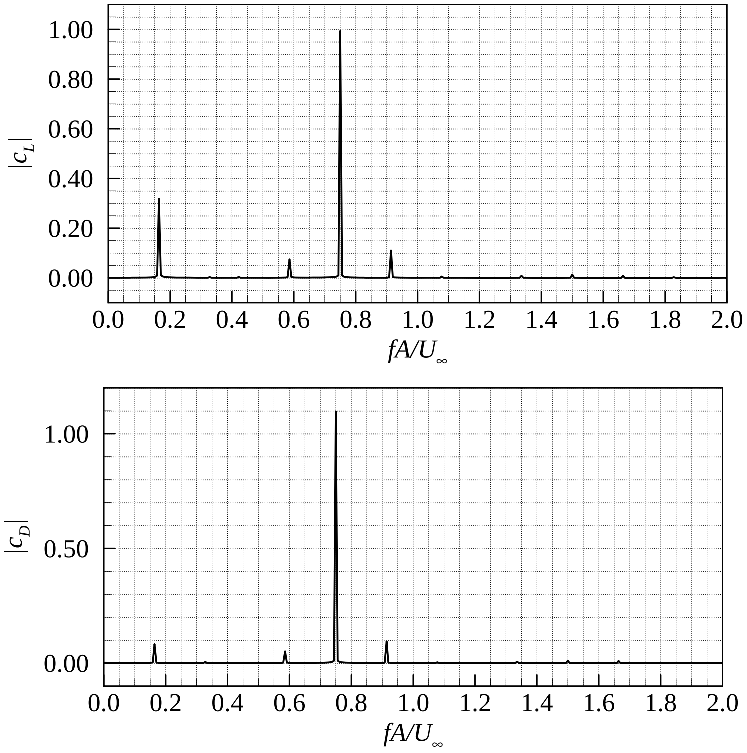
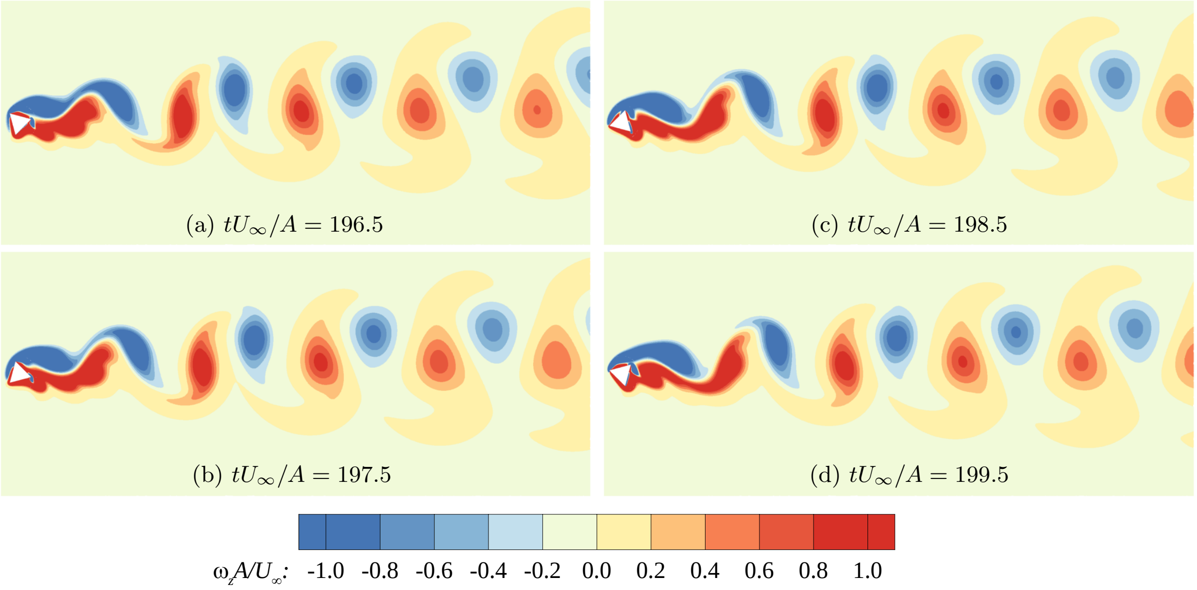


Figure 28. Spectrum of unsteady lift and drag coefficients for flow over a rotating triangular cylinder.

The flow fields at several time instants are plotted in Fig. 29. We see that a negative vortex is always shed off from the top and a positive one from the bottom of the cylinder. The vortex street consists of negative and positive vortex pairs, where the positive vortex has larger size than the negative one. This pattern is very similar to the previous two cylinders of different shapes.


(a) $tU_\infty/A = 196.5$

(c) $tU_\infty/A = 198.5$

(b) $tU_\infty/A = 197.5$

(d) $tU_\infty/A = 199.5$

Figure 29. Vorticity contours for flow over a rotating triangular cylinder.

IV.F. Flow over a Rotating Square Cylinder

In this test, we apply the solver to study flow over a rotating square cylinder. The cylinder has an edge length of $A = 1.0$. It once again rotates around its center at an angular speed of $\omega_0 A/U_\infty = \pi/2$ counterclockwise. The freestream flow again has a Mach number of $Ma = 0.1$, and a Reynolds number of $Re = 100$ (based on cylinder edge length and freestream flow properties).

This case has a similar mesh to the previous tests. Fig. 30 shows global and local views of the mesh. The overall domain size is 100×100, and the cylinder locates 30 units away from the inlet. The domain is discretized into 8,679 cells, with 533 within the rotating subdomain whose radius is 1.0. The cylinder surface is treated as adiabatic wall, and all other boundaries are treated as characteristic farfield. The computational time step size is $\Delta t U_\infty/A = 2.0 \times 10^{-4}$. Three schemes with orders of accuracy from 4 to 6 have been tested to ensure mesh- and scheme-independent solution, and all results reported below are from the 6-th order scheme which has been confirmed to have provided enough resolution.

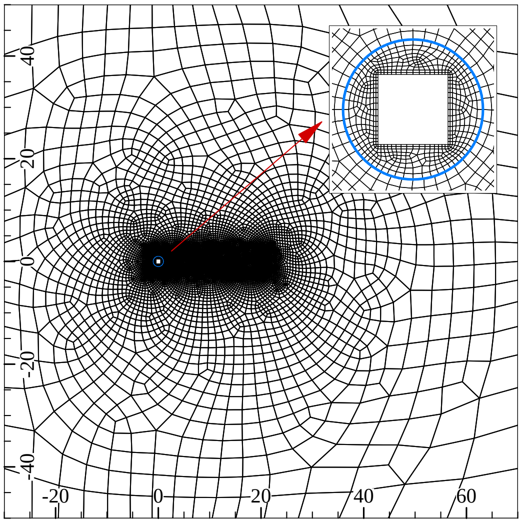


Figure 30. Mesh for a rotating square cylinder (blue circle indicates sliding interface).

Fig. 31 plots the time history for C_L and C_D from $tU_\infty/D = 160$ to 200. As can be seen, the envelopes of the curves seem rather periodic, and higher frequency components are reflected as wiggles on the envelopes. Once again, this square cylinder experiences an always-negative lift and always-positive drag. The mean values from time averaging over 178 time units are found to be $\bar{C}_L = -3.207$ and $\bar{C}_D = 1.420$. The Lissajous curve for C_L and C_D is plotted in Fig. 32, and the shape indicates the presence of multiple frequency components in the flow field.

To see the frequency distribution in the flow, we have computed the power spectrum of the unsteady lift coefficient c_L and the unsteady drag coefficient c_D, the results are plotted in Fig. 33. For c_L, we see four obvious peaks at $fD/U_\infty = 0.14$, 0.86, 1.0 and 1.14, respectively. The largest peak appears at $fD/U_\infty = 0.140$. The second largest peak is seen at $fD/U_\infty = 1.0$, which is induced by the rotating motion (it is four times the rotating frequency due to geometry symmetry of the square cylinder). For c_D, we see three more peaks at $fD/U_\infty = 0.28$, 0.72 and 1.28 on the spectrum, but with much smaller magnitudes than those at the other four frequencies. The larges peak for c_D is seen at $fD/U_\infty = 1.0$, and the second largest is found at $fD/U_\infty = 0.14$ (with a little smaller magnitude). The frequencies corresponding to the first two largest peaks are evidently reflected in the C_L and C_D curves in Fig. 31. The lower frequency (i.e., $fD/U_\infty = 0.14$) that corresponds to a period of $TU_\infty/D = 7.14$ shows up as the period of the envelopes for both C_L and C_D. The higher frequency (i.e., $fD/U_\infty = 1.0$) which is induced by the rotating motion and corresponds to a period of $TU_\infty/D = 1.0$ is approximately the time interval between two adjacent small peaks (or troughs) on the envelopes. An interesting conclusion can be drawn by comparing the spectra of all the cylinders tested so far: as the number of edges increases, the rotating frequency becomes less dominant in the flow field, and in the limit of infinite edges (i.e., the circular cylinder) the rotating frequency even does not show up in the spectrum at all (see Fig. 16).

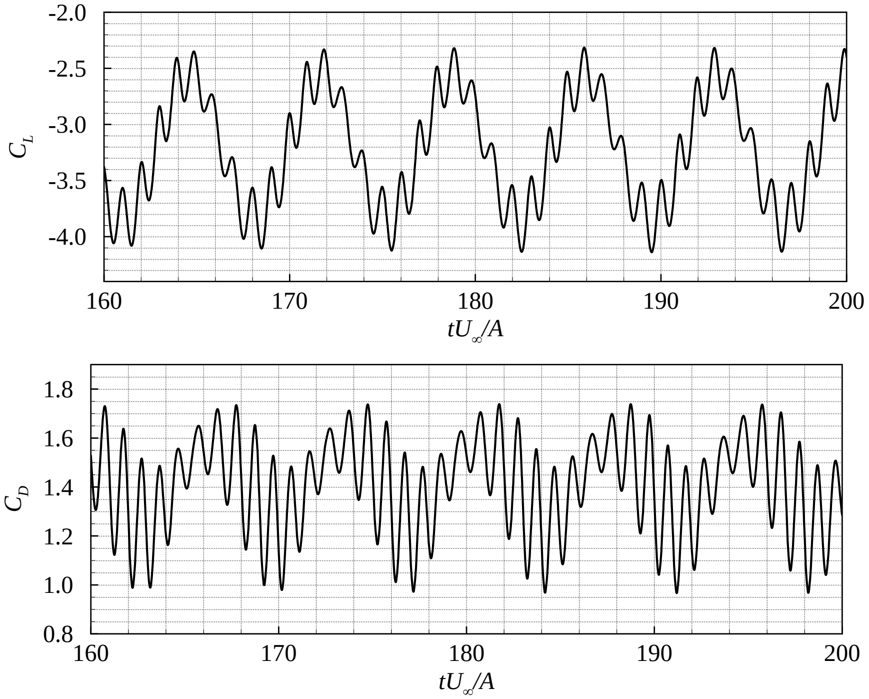


Figure 31. Converged lift and drag coefficients for flow over a rotating square cylinder.

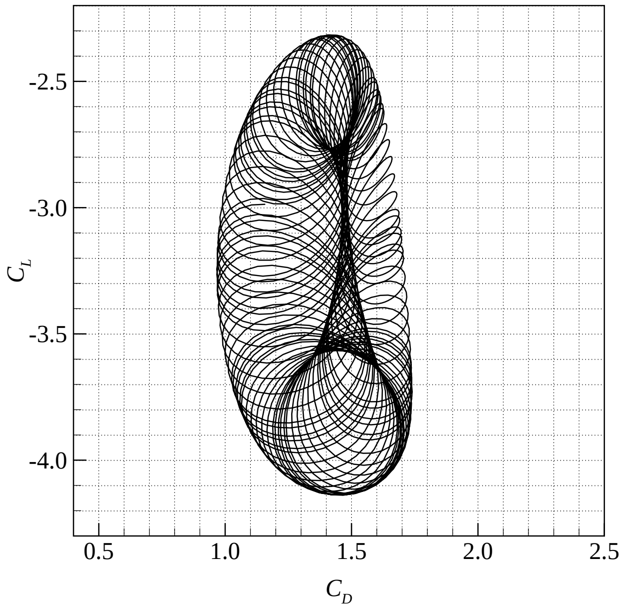


Figure 32. Lissajous curve of C_L and C_D in 20 rotating periods (from tU_∞/D = 120 to 200) for a rotating square cylinder.

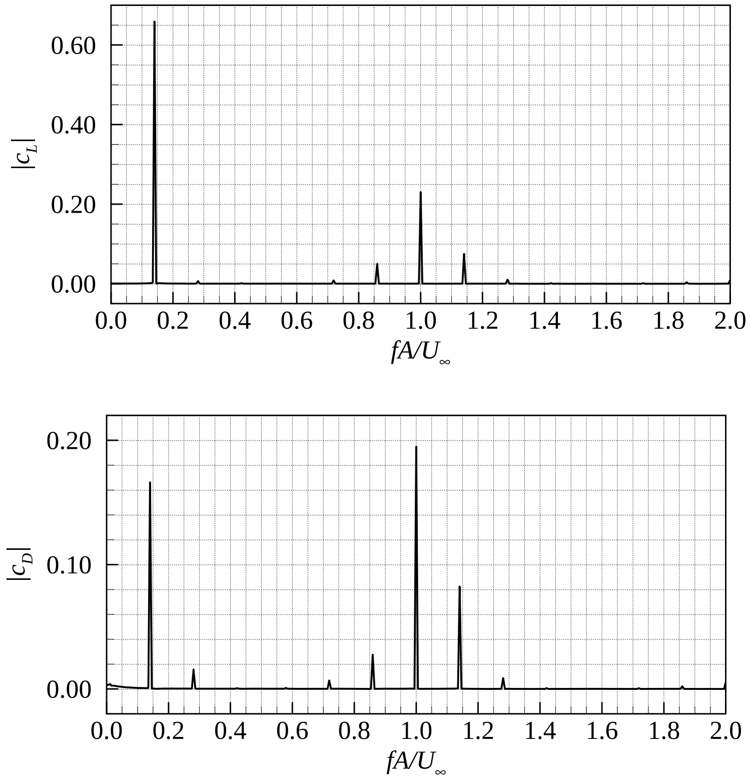
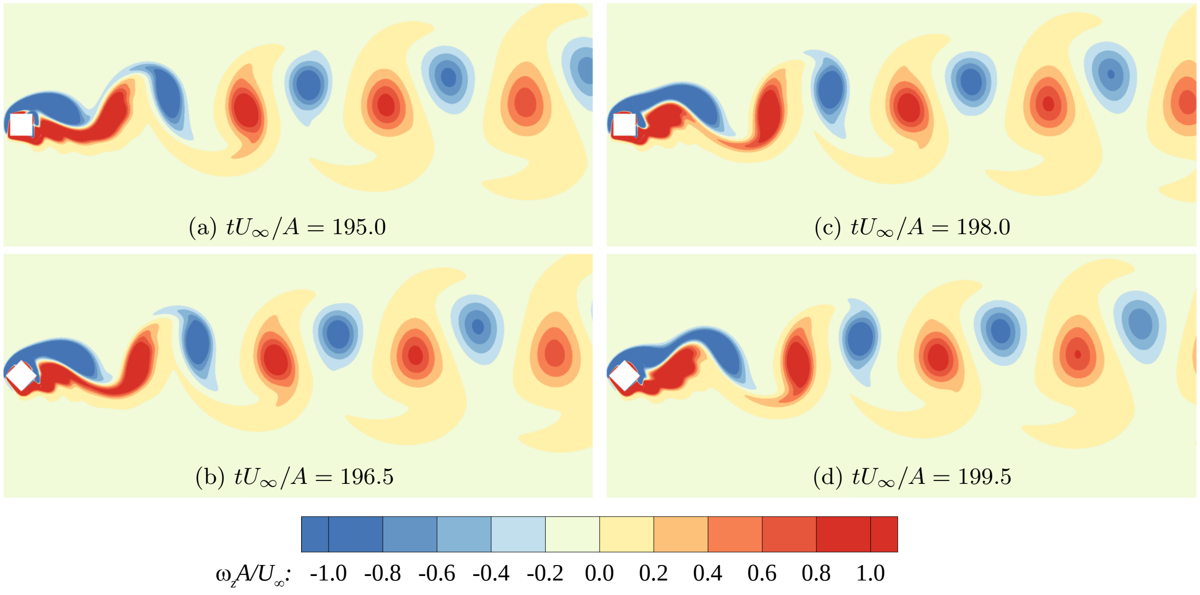


Figure 33. Spectrum of unsteady lift and drag coefficients for flow over a rotating square cylinder.

We visualize the flow field by the vorticity contours as shown in Fig. 34. This flow field again shares almost all the similarities of the previous ones.


(a) $tU_\infty/A = 195.0$

(c) $tU_\infty/A = 198.0$

(b) $tU_\infty/A = 196.5$

(d) $tU_\infty/A = 199.5$

Figure 34. Vorticity contours for flow over a rotating square cylinder.

IV.G. Flow over Two Rotating Square Cylinders

The first aim of this test is to demonstrate the solver's capability of dealing with multiple rotating objects. The second aim is to investigate the effects of adding a tandem cylinder to the flow. The two cylinders are separated by a distance of $D = 3$ (measured from center to center) and have in phase rotating motions. Each cylinder has an edge length of $A = 1.0$, and rotates counterclockwise around its center at an angular speed of $\omega_0 A/U_\infty = \pi/2$. The freestream flow again has a Mach number of $Ma = 0.1$, and a Reynolds number of $Re = 100$ (based on cylinder edge length and freestream flow properties).

Fig. 35 shows global and local views of the mesh. The overall domain size is 100 × 100, and the first cylinder locates 28.5 units away from the inlet. The domain is discretized into 9,353 cells, with 533 within each rotating subdomain whose radius is 1.0. The cylinder surface is treated as adiabatic wall, and all other boundaries are treated as characteristic farfield. The time step size for computation is $\Delta t U_\infty/A = 2.0 \times 10^{-4}$. Three schemes with orders of accuracy from 4 to 6 have been tested to ensure mesh- and scheme-independent solution, and all results reported below are from the 6-th order scheme that ensures mesh and scheme independent solutions.

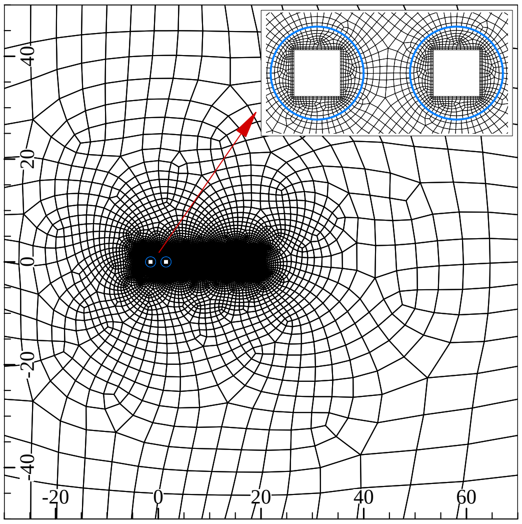


Figure 35. Mesh for two tandem rotating square cylinders (blue circles indicate sliding interfaces).

Fig. 36 plots the time history of the converged C_L and C_D from $tU_\infty/D = 160$ to 200 for both the front and the rear cylinders. As can be seen, the curves seem rather periodic, but the effects of multiple frequencies are still visible, for example on the C_D curve for the rear cylinder. This is also reflected in the Lissajous curve as shown in Fig. 37, as the Lissajous curves do not perfectly repeat themselves. Overall, the C_L and C_D curves show that the magnitudes of C_L and C_D are both smaller for the rear cylinder than those for the front cylinder. The time averaged values are: $\bar{C}_{L1} = -2.954$, $\bar{C}_{D1} = 0.848$, $\bar{C}_{L2} = -1.338$ and $\bar{C}_{D2} = 0.115$, where the subscript '1' stands for the front cylinder and '2' for the rear cylinder. It is of interest to compare the front cylinder with the previous single square cylinder case. We see an obvious reduction of drag (about 40.3%) and a minor drop of lift (about 7.9%) on the front cylinder when a tandem cylinder presents. If we take the two tandem cylinders as a combined system and compare it to the previous single square cylinder, we see an overall drag reduction of 32.3% and lift increase of 33.8%.

Fig. 38 shows the power spectra of the unsteady lift and drag coefficients. As can be seen, only one dominant frequency of $fD/U_\infty = 1.0$ appears on all curves, and this frequency is four times the rotating frequency. Therefore, this flow is mostly dominated the rotating motion. This single dominant frequency also answers why the C_L and C_D curves have very 'periodic' appearances.

Finally, we visualize the flow field by the vorticity contours and shown it in Fig. 34 for different time instants. Very different from all previous single cylinder cases, the wake of this case is very stable and we see no obvious vortex shedding even in the very far wake region. This pattern also answers why the lift and drag coefficients have smaller fluctuations than the single cylinders that have vortex shedding in the flow field.

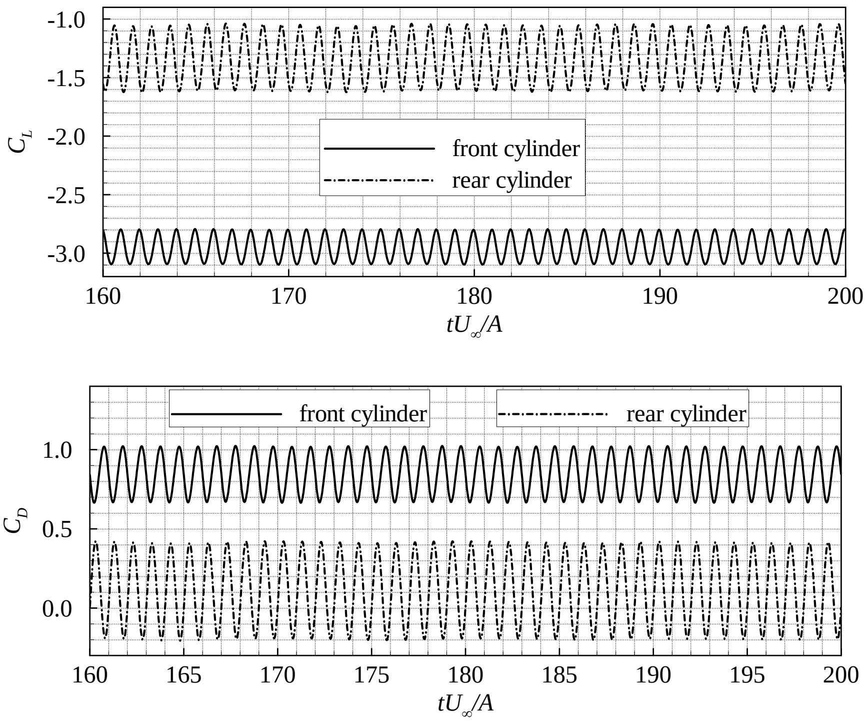


Figure 36. Converged lift and drag coefficients of two tandem rotating square cylinder.

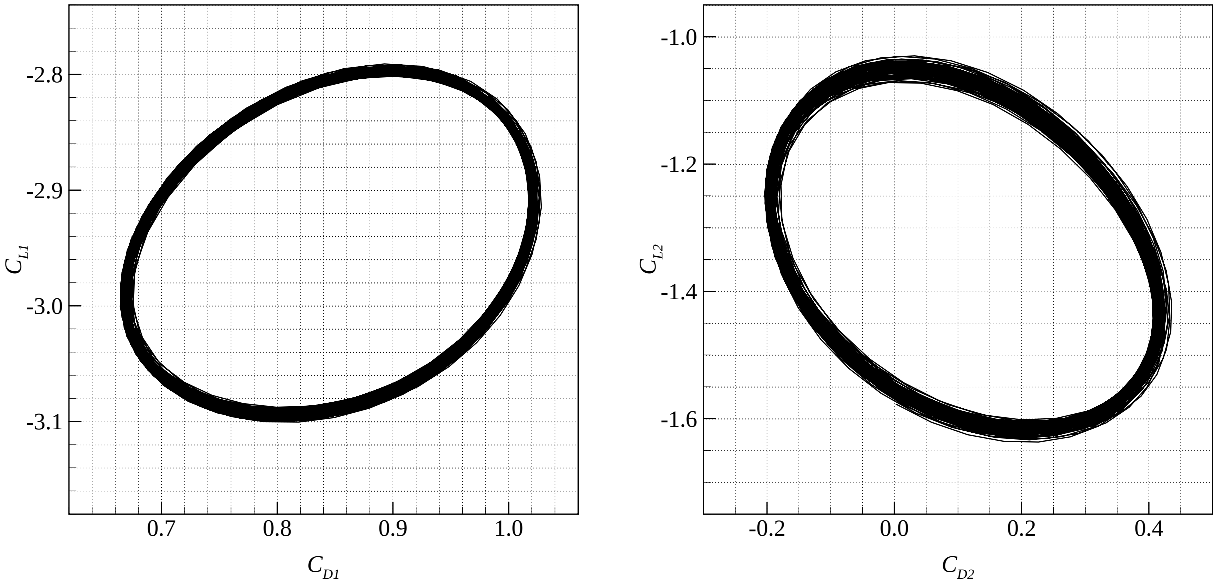


Figure 37. Lissajous curves of C_L and C_D in 20 rotating periods (from tU_∞/D = 120 to 200) for two tandem rotating square cylinders.

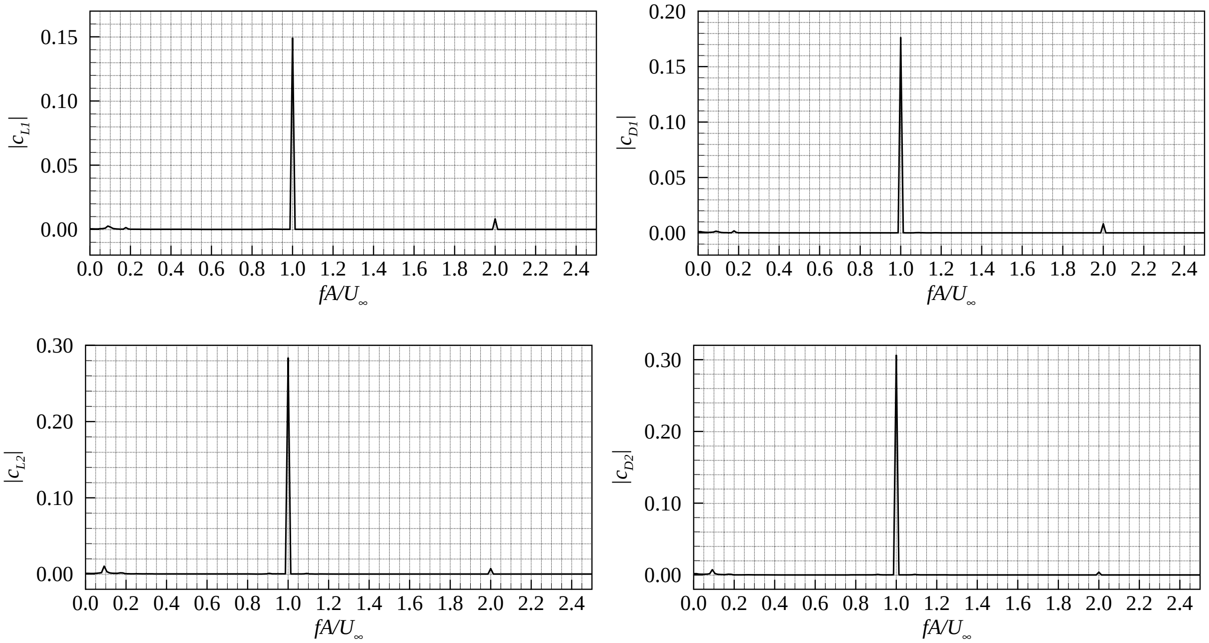
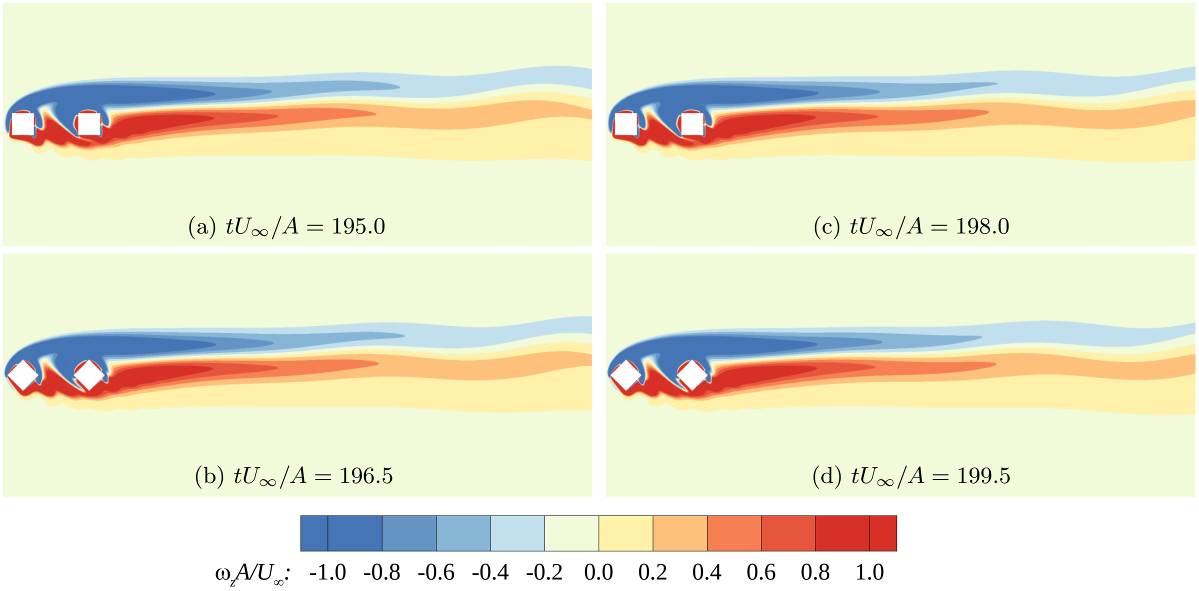


Figure 38. Spectrum of unsteady lift and drag coefficients for two tandem rotating square cylinder.


(a) $tU_\infty/A = 195.0$

(c) $tU_\infty/A = 198.0$

(b) $tU_\infty/A = 196.5$

(d) $tU_\infty/A = 199.5$

Figure 39. Vorticity contours for flow over two tandem rotating square cylinders.

V. Conclusions

We have successfully developed a high-order nonuniform sliding-mesh flux reconstruction method for simulating flows about rotating geometries. This method greatly simplifies mesh generation compared to our previous method that requires uniform mesh on a sliding interface. Theoretically, this method could be arbitrarily high-order accurate, and in numerical tests we have successfully achieved 2-nd to 10-th order accuracies on both inviscid and viscous flows. We further compared the present nonconforming sliding-mesh method to traditional conforming mesh approaches for simulating flow about rotating geometries, and very good agreements have been obtained. Moreover, from these tests, the present method is shown to be superior to traditional whole-domain-rotating methods, for the present method allows much better grid resolution in the wake region and could potentially save a great amount of computational cost.

As applications, we studied flow over a series of single rotating cylinders with different shapes, and also two tandem rotating square cylinders. In the single cylinder cases, we observed very similar vortex structures in the flow fields in spite of different cylinder shapes. But from spectrum analysis, it is interesting to notice that as the number of edges of a cylinder increases, the rotating frequency becomes less dominant in the flow field. In the limit of circular cylinder (i.e., infinite number of edges), the rotating frequency even does not show up in the spectrum at all. By adding a second rotating square cylinder to the flow (i.e., forming a two tandem rotating cylinder system), we noticed that vortex shedding has been greatly suppressed. Meanwhile, the drag of the front cylinder has been reduced by more than 40%, and the overall system also experiences a drag reduction and lift enhancement. For this reason, a tandem rotating cylinder has the potential for effective active flow control, which requires further and boarder investigation in the future.

VI. Acknowledgment

Chunlei Liang and Bin Zhang would like to express our acknowledgments of an ONR Young Investigator Program (YIP) grant awarded to Chunlei Liang, administrated by Dr. Ki-Han Kim. Zihua Qiu is supported by an oversea study program from the China Scholarship Council.

References

[1] Zhang, B. and Liang, C., "A simple, efficient, high-order accurate sliding-mesh interface approach to FR/CPR method on coupled rotating and stationary domains," *AIAA paper 2015-1742*, 2015.

[2] Zhang, B. and Liang, C., "A simple, efficient, and high-order accurate curved sliding-mesh interface approach to spectral difference method on coupled rotating and stationary domains," *Journal of Computational Physics*, Vol. 295, 2015, pp. 147–160.

[3] Zhang, B., Liang, C., Yang, J., and Rong, Y., "A 2D parallel high-order sliding and deforming spectral difference method," *Computers & Fluids*, Vol. 139, 2016, pp. 184–196.

[4] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., et al., "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845.

[5] 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.

[6] 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.

[7] Lele, S. K., "Compact finite difference schemes with spectral-like resolution," *Journal of Computational Physics*, Vol. 103, No. 1, 1992, pp. 16–42.

[8] Visbal, M. R. and Gaitonde, D. V., "On the use of higher-order finite-difference schemes on curvilinear and deforming meshes," *Journal of Computational Physics*, Vol. 181, No. 1, 2002, pp. 155–185.

[9] Patera, A. T., "A spectral element method for fluid dynamics: laminar flow in a channel expansion," *Journal of Computational Physics*, Vol. 54, No. 3, 1984, pp. 468–488.

[10] Karniadakis, G. E. and Sherwin, S. J., *Spectral/hp Element Methods for Computational Fluid Dynamics*, Oxford University Press, Oxford, 2nd ed., 2005.

[11] 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.

[12] Wang, Z. J. and Liu, Y., "Spectral (finite) volume method for conservation laws on unstructured grids: II. Extension to two-dimensional scalar equation," *Journal of Computational Physics*, Vol. 179, No. 2, 2002, pp. 665–697.

[13] Kopriva, D. A. and Kolias, J. H., "A conservative staggered-grid Chebyshev multidomain method for compressible flows," *Journal of Computational Physics*, Vol. 125, 1996, pp. 244–261.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] Van den Abeele, K., Lacor, C., and Wang, Z. J., "On the stability and accuracy of the spectral difference method," *Journal of Scientific Computing*, Vol. 37, 2008, pp. 162–188.

[19] Jameson, A., "A proof of the stability of the spectral difference method for all orders of accuracy," *Journal of Scientific Computing*, Vol. 45, 2010, pp. 348–358.

[20] 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.

[21] Liang, C., Kannan, R., and Wang, Z. J., "A p-multigrid spectral difference method with explicit and implicit smoothers on unstructured triangular grids," *Computers & Fluids*, Vol. 38, 2009, pp. 254–265.

[22] Premasuthan, S., Liang, C., Wang, Z. J., and Jameson, A., "A p-multigrid spectral difference method for viscous flow," *AIAA paper 2009-950*, 2009.

[23] Balan, A., May, G., and Schöberl, J., "A stable high-order spectral difference method for hyperbolic conservation laws on triangular elements," *Journal of Computational Physics*, Vol. 231, No. 5, 2012, pp. 2359–2375.

[24] 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.

[25] Liang, C., Premasuthan, S., Jameson, A., and Wang, Z. J., "Large eddy simulation of compressible turbulent channel flow with spectral difference method," *AIAA paper 2009-402*, 2009.

[26] Mohammad, A. H., Wang, Z. J., and Liang, C., "LES of turbulent flow past a cylinder using spectral difference method," *Advances in Applied Mathematics and Mechanics*, Vol. 2, 2010, pp. 451–466.

[27] Parsani, M., Ghorbaniasl, G., Lacor, C., and Turkel, E., "An implicit high-order spectral difference approach for large eddy simulation," *Journal of Computational Physics*, Vol. 229, 2010, pp. 5373–5393.

[28] Castonguay, P., Liang, C., and Jameson, A., "Simulation of transitional flow over airfoils using the spectral difference method," *AIAA paper 2010-4626*, 2010.

[29] Liang, C., Ou, K., Premasuthan, S., Jameson, A., and Wang, Z. J., "High-order accurate simulations of unsteady flow past plunging and pitching airfoils," *Computers & Fluids*, Vol. 40, 2011, pp. 236–248.

[30] Parsani, M., Ghorbaniasl, G., and Lacor, C., "Validation and application of an high-order spectral difference method for flow induced noise simulation," *Journal of Computational Acoustics*, Vol. 19, No. 03, 2011, pp. 241–268.

[31] Lodato, G., Castonguay, P., and Jameson, A., "Structural wall-modeled LES using a high-order spectral difference scheme for unstructured meshes," *Flow, Turbulence and Combustion*, Vol. 92, No. 1-2, 2014, pp. 579–606.

[32] Wang, J., Liang, C., and Miesch, M. S., "A compressible high-order unstructured spectral difference code for stratified convection in rotating spherical shells," *Journal of Computational Physics*, Vol. 290, 2015, pp. 90–111.

[33] Huynh, H. T., "A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods," *AIAA paper 2007-4079*, 2007.

[34] Huynh, H. T., "A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion," *AIAA paper 2009-403*, 2009.

[35] 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.

[36] Vincent, P. E., Castonguay, P., and Jameson, A., "A new class of high-order energy stable flux reconstruction schemes," *Journal of Scientific Computing*, Vol. 47, 2011, pp. 50–72.

[37] Jameson, A., Vincent, P. E., and Castonguay, P., "On the non-linear stability of flux reconstruction schemes," *Journal of Scientific Computing*, Vol. 50, No. 2, 2012, pp. 434–445.

[38] Huynh, H. T., Wang, Z. J., and Vincent, P. E., "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.

[39] Wang, Z. J. and Huynh, H. T., "A review of flux reconstruction or correction procedure via reconstruction method for the Navier-Stokes equations," *Mechanical Engineering Reviews*, Vol. 3, No. 1, 2016.

[40] 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.

[41] 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.

[42] Steger, J. L., Dougherty, F. C., and Benek, J. A., "A Chimera grid scheme," *Proceedings of the Applied Mechanics, Bioengineering, and Fluids Engineering Conference*, Houston, TX, 1983, pp. 59–69.

[43] Mittal, R. and Iaccarino, G., "Immersed boundary methods," *Annual Review of Fluid Mechanics*, Vol. 37, 2005, pp. 239–261.

[44] 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.

[45] Thomas, P. D. and Lombard, C. K., "Geometric conservation law and its application to flow computations on moving grids," *AIAA Journal*, Vol. 17, 1979, pp. 1030–1037.

[46] Ergatoudis, I., Irons, B. M., and Zienkiewicz, O. C., "Curved, isoparametric, "quadrilateral" elements for finite element analysis," *International Journal of Solids and Structures*, Vol. 4, No. 1, 1968, pp. 31–42.

[47] Gordon, W. J. and Hall, C. A., "Transfinite element methods: blending-function interpolation over arbitrary curved element domains," *Numerische Mathematik*, Vol. 21, No. 2, 1973, pp. 109–129.

[48] Gordon, W. J. and Hall, C. A., "Construction of curvilinear co-ordinate systems and applications to mesh generation," *International Journal for Numerical Methods in Engineering*, Vol. 7, No. 4, 1973, pp. 461–477.

[49] 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.

[50] Kraaijevanger, J. F. B. M., "Contractivity of Runge-Kutta methods," *BIT Numerical Mathematics*, Vol. 31, No. 3, 1991, pp. 482–528.

[51] Ruuth, S., "Global optimization of explicit strong-stability-preserving Runge-Kutta methods," *Mathematics of Computation*, Vol. 75, No. 253, 2006, pp. 183–207.