31st Symposium on Naval Hydrodynamics
Monterey, California, 11-16 September 2016

A High-Order Sliding-Mesh Spectral Difference Solver for Simulating Unsteady Flows around Rotating Objects

Bin Zhang and Chunlei Liang
(Department of Mechanical and Aerospace Engineering, The George Washington University, Washington, DC, USA)

ABSTRACT

In this paper, we report a high-order three-dimensional sliding-mesh algorithm for compressible Navier-Stokes equations and simulations of unsteady flows around 3D rotating objects. This 3D sliding-mesh method is an extension of our previously published 2D sliding-mesh methods [34, 35] based on the spectral difference method. Similar to other sliding-mesh methods, a computational domain is split into non-overlapping subdomains, and each one may have rotational motions relative to its neighbors, resulting in nonconforming sliding interfaces in between. Our sliding-mesh method is unique because two neighboring subdomains are coupled through curved dynamic mortar elements along the sliding interfaces. In order to compute common solutions and fluxes on mortars, both conservative variables and fluxes are first projected from cell faces to mortar elements. Subsequently, they are projected back to cell faces to ensure conservation. To demonstrate the accuracy of the solver, both inviscid and viscous benchmark flows are tested. The solver is shown capable of preserving the high-order accuracy of the background spectral difference method. This solver is also able to handle sliding interfaces with different orientations, is thus very flexible for flow simulations with complex geometries. Numerical tests are also carried out on more complex flows, such as flow around a rotating elliptic cylinder, transitional Taylor-Couette flow with multiple sliding interfaces, and flows around a rotating cube. This solver can be applied to a wide range of rotating flow problems, for example, flows around wind turbines, marine propellers, etc.

INTRODUCTION

High-order (third and above) numerical methods are becoming popular in recent years in the computational fluid dynamics (CFD) community. They are capable of producing more accurate solutions on relatively coarse grids at lower computational cost than low-order methods [32]. More and more applications of high-order methods to complex engineering flow problems have also been observed recently. Summaries of the recent development and applications of high-order methods can be found in several books [11, 31, 7, 4], and also in several review papers [5, 30, 32].

In the high-order "family", the spectral difference (SD) method is one of the most efficient methods as it solves equations in differential forms directly. This method was first introduced by Kopriva et al. [14, 12] with the name of "staggered-grid Chebyshev multidomain method" for the 2D compressible Euler equations on quadrilateral grids. Kopriva [13] further applied this method to the full compressible Navier-Stokes equations and showed spectral accuracy on a series of test problems.

Liu et al. [20], and Wang et al. [33] later extended this method to unstructured triangular grids for solving the wave equations and Euler equations, respectively. Since this method has spectral accuracy and directly solves the equations in differential forms, the involved authors named this method as the "spectral difference" method. Applications of the SD method to the Navier-Stokes equations on unstructured triangular or mixed-element grids were reported by several researchers [23, 16, 15].

Van den Abeele et al. [29] studied the stability of the SD method, and pointed out that stability and accuracy of the SD method depends only on the locations of flux points but not on solution points. They also found that the previously used Chebyshev-Gauss-Lobatto flux points make the scheme weakly instable. The stability of the SD method for linear wave equations for all orders of accuracy was proved by Jameson [9]. The same author confirmed the findings of Van den Abeele et al. [29], and reported that the SD method will be provably stable when the interior flux points are chosen as the zeros of the Legendre polynomials. A stable SD method for triangular grids using the Raviart-Thomas spaces was reported by Balan et al. [1].

The SD method is a relatively new method compared to many other high-order methods, such as the DG method, the spectral element method, etc. We have seen more applications of the SD method to a wide range of problems. For example, Liang et al. [19] performed large

eddy simulation (LES) of channel flows using the SD method; Castonguay et al. [3] simulated transitional flow over airfoils; Mohammad et al. [24] carried out LES of flow past a cylinder; Parsani et al. [26] studied flow induced noise; Lodato et al. [21] did wall modeled LES; Ou et al. [25], etc.

Liang et al. [17] applied the SD method to the Navier-Stokes equations on moving and deforming grids under the arbitrary-Lagrangian-Eulerian (ALE) framework. Their simulations showed excellent agreement with previous experimental studies. However, this deforming grid approach is limited to small grid motions, and it fails when the grid undergoes very large rotational motions that cause severe grid skewness. Therefore, alternative approaches are required to study flows about objects with arbitrary rotational motions.

Zhang and Liang [34], Zhang et al. [35] developed a novel sliding-mesh method for the SD method for simulating two-dimensional flows. This method can completely remove the rotation limitations on traditional conforming ALE methods, and it is shown to be high-order accurate and highly efficient. In the present work, we extend this method to three-dimensional, and we further develop a novel three-dimensional flow solver for rotating geometries on unstructured grids.

The rest of this paper is organized as follows: Section 2 gives the governing equations. Section 3 reviews the SD method and presents the interface method in details. Accuracy tests and applications are reported in Section 4. Finally, Section 5 concludes the paper.

GOVERNING EQUATIONS

In our method, a computational domain is split into non-overlapping subdomains, and some of them may rotate while others are fixed. This section describes the governing equations for rotating and fixed subdomains.

The Navier-Stokes Equations on Fixed Domain

The following three-dimensional unsteady compressible Navier-Stokes equations in conservative form are considered for fixed domains,

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

where $\mathbf{Q}$ is the vector of conservative variables, $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ are the flux vectors in each coordinate direction. These terms have the following expressions,

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

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

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

$$\mathbf{H}=\mathbf{H}_{inv}(Q)+\mathbf{H}_{vis}(Q,\nabla Q), \tag{5}$$

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

As shown in Equations (3)-(5), the fluxes have been split into inviscid and viscous parts. The inviscid fluxes are only functions of the conservative variables and take the following forms,

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

$$\mathbf{G}_{inv}=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ v(E+p)\end{bmatrix}, \tag{7}$$

$$\mathbf{H}_{inv}=\begin{bmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ w(E+p)\end{bmatrix}. \tag{8}$$

The viscous fluxes are functions of the conservative variables as well as their gradients. Their expressions are

$$\mathbf{F}_{vis}=-\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{yx}\\ \tau_{zx}\\ u\tau_{xx}+v\tau_{yx}+w\tau_{zx}+\mathrm{k}T_x\end{bmatrix}, \tag{9}$$

$$\mathbf{G}_{vis}=-\begin{bmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ \tau_{zy}\\ u\tau_{xy}+v\tau_{yy}+w\tau_{zy}+\mathrm{k}T_y\end{bmatrix}, \tag{10}$$

$$\mathbf{H}_{vis}=-\begin{bmatrix}0\\ \tau_{xz}\\ \tau_{yz}\\ \tau_{zz}\\ u\tau_{xz}+v\tau_{yz}+w\tau_{zz}+\mathrm{k}T_y\end{bmatrix}, \tag{11}$$

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

The Navier-Stokes Equations on Rotating Domain

On rotating domains, we use a set of simplified equations that are equivalent to the arbitrary Lagrange-Eulerian

(ALE) [8] form of Equation (1). Due to grid motion, the inviscid fluxes are modified to take the following forms,

$$\mathbf{F}_{inv} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{bmatrix} - u_g \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \tag{12}$$

$$\mathbf{G}_{inv} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho wv \\ v(E+p) \end{bmatrix} - v_g \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \tag{13}$$

$$\mathbf{H}_{inv} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{bmatrix} - w_g \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \tag{14}$$

where u_g, v_g and w_g are the grid velocity components in each coordinate direction. The viscous fluxes and all other variables stay unaffected and have the same expressions as those in the previous section.

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

The Transformed Equations

As will be discussed in the next section, the physical domains are discretized into hexahedral grid cells, and each cell is mapped to a standard cubic element in the computational space. This mapping facilitates the construction of solution and flux polynomials. As a result, we need to solve a set of transformed equations within each standard element. Let us assume that the physical coordinates (x,y,z) are mapped to the computational ones (ξ,η,ζ) through a transformation: $x = x(\xi,\eta,\zeta)$, $y = y(\xi,\eta,\zeta)$ and $z = z(\xi,\eta,\zeta)$. It can be shown that Equation (1) will take the following conservative form after coordinate transformation,

$$\frac{\partial \widetilde{\mathbf{Q}}}{\partial t} + \frac{\partial \widetilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \widetilde{\mathbf{G}}}{\partial \eta} + \frac{\partial \widetilde{\mathbf{H}}}{\partial \zeta} = 0, \tag{15}$$

where $\widetilde{\mathbf{Q}} = |\mathscr{J}|\mathbf{Q}$ is the computational variable, and the computational fluxes $\widetilde{\mathbf{F}}$, $\widetilde{\mathbf{G}}$ and $\widetilde{\mathbf{H}}$ are related to the physical ones as,

$$\begin{pmatrix} \widetilde{\mathbf{F}} \\ \widetilde{\mathbf{G}} \\ \widetilde{\mathbf{H}} \end{pmatrix} = |\mathscr{J}| \mathscr{J}^{-1} \begin{pmatrix} \mathbf{F} \\ \mathbf{G} \\ \mathbf{H} \end{pmatrix}, \tag{16}$$

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

$$|\mathscr{J}| = \left| \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} \right| = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_{z\eta} \\ z_\xi & z_\eta & z_\zeta \end{vmatrix}, \tag{17}$$

$$\mathscr{J}^{-1} = \frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix}. \tag{18}$$

NUMERICAL METHODS

In this section, we will first give a brief description of the SD method that is used for the spatial discretization of Equation (15). Following that, we give the formulations and procedures of the sliding-mesh method which is built into the SD method to deal with rotating geometries. For temporal discretization, an explicit five-stage fourth-order strong stability preserving Runge-Kutta method [28] is employed in this study.

The SD Method

In the SD method, each physical cell is mapped to a standard computational element to facilitate the construction of solution and flux polynomials. When it comes to hexahedral cell, each one is mapped to a standard cubic element, i.e. $0 \le \xi \le 1$, $0 \le \eta \le 1$ and $0 \le \zeta \le 1$, as shown in Figure 1. Liang, et al [15, 18] reported that using linear cells on curved boundaries might cause oscillations on solutions and even leads to divergence. For this reason, 32-node tricubic hexahedral cells, which can represent curved boundaries more smoothly, have been employed in the present study for all the test cases. The mapping is done through the following iso-parametric mapping

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sum_{i=1}^{K} M_i(\xi,\eta,\zeta) \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \tag{19}$$

where K is the total number of nodes that define a cell (e.g., $K = 20$ for a triquadratic cell and $K = 20$ for a tricubic cell), M_i is the shape function associated with the i-th node whose coordinates are (x_i, y_i, z_i).

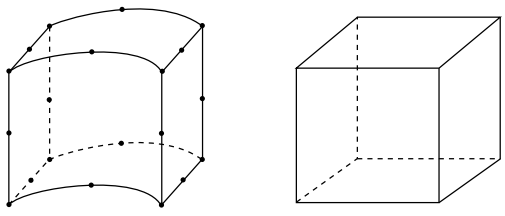

Figure 1: Mapping of physcal cell to a standard computational element: left, phsycal cell; right, standard element.

In order to construct a degree $(N-1)$ solution polynomial and a degree N flux polynomial in each coordinate direction, N solution points (SPs) and $(N+1)$ flux points (FPs) are required in that direction. In the present study, the SPs (denoted by X_s, where $s = 1, 2, \cdots, N$) are chosen as the Chebyshev-Gauss points, and the FPs (denoted by $X_{s+1/2}$, where $s = 0, 1, \cdots, N$) are set as $(N-1)$ Legendre-Gauss points plus two end points (see Liang et al. [15] for details). Figure 2 shows the SPs and FPs in the ξ-η plane for a third-order scheme.

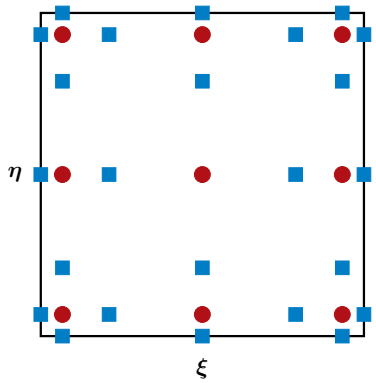

Figure 2: Distribution of solution points (circles) and flux points (squares) in the ξ-η plane for a third-order scheme.

From the SPs and FPs, Lagrange interpolation bases can be constructed. For example, at the i-th SP and i-th FP in the X direction (where X can be ξ, η or ζ), we have the following Lagrange bases,

$$h_i(X) = \prod_{s=1, s\neq i}^{N} \left(\frac{X - X_s}{X_i - X_s} \right), \tag{20}$$

$$l_{i+1/2}(X) = \prod_{s=0, s\neq i}^{N} \left(\frac{X - X_{s+1/2}}{X_{i+1/2} - X_{s+1/2}} \right). \tag{21}$$

The conservative solution variables can be reconstructed from tensor products of the Lagrange polynomials from each direction as,

$$\mathbf{Q} = \sum_{k=1}^{N} \sum_{j=1}^{N} \sum_{i=1}^{N} \frac{\widetilde{\mathbf{Q}}_{i,j,k}}{|\mathscr{J}_{i,j,k}|} h_i(\xi) \cdot h_j(\eta) \cdot h_k(\zeta). \tag{22}$$

Similarly, the fluxes within each standard element can be reconstructed as,

$$\widetilde{\mathbf{F}} = \sum_{k=1}^{N} \sum_{j=1}^{N} \sum_{i=0}^{N} \widetilde{\mathbf{F}}_{i+1/2,j,k} l_{i+1/2}(\xi) \cdot h_j(\eta) \cdot h_k(\zeta), \tag{23}$$

$$\widetilde{\mathbf{G}} = \sum_{k=1}^{N} \sum_{j=0}^{N} \sum_{i=1}^{N} \widetilde{\mathbf{G}}_{i,j+1/2,k} h_i(\xi) \cdot l_{j+1/2}(\eta) \cdot h_k(\zeta), \tag{24}$$

$$\widetilde{\mathbf{H}} = \sum_{k=0}^{N} \sum_{j=1}^{N} \sum_{i=1}^{N} \widetilde{\mathbf{H}}_{i,j,k+1/2} h_i(\xi) \cdot h_j(\eta) \cdot l_{k+1/2}(\zeta). \tag{25}$$

The above reconstructed solution and fluxes are only element-wise continuous, but discontinuous across cell boundaries. Therefore, common solutions and fluxes need to be computed on cell boundaries to ensure conservation. A Riemann solver can be used to compute the common inviscid fluxes. In the present implementation, the Rusanov solver [27] has been adopted for this purpose. The common viscous fluxes are computed from the common solutions and the common gradients. More details on the computation of the common fluxes can be found in a previous paper by Liang et al. [19].

The Nonconforming Interface Method

The present implementation allows two kinds of sliding interfaces: one that is parallel to the rotating axis, and the other that is perpendicular to the axis. These two kinds of interfaces have cylindrical shapes and circular annulus shapes, respectively. An example is given in Figure 3, where the rotating axis is the z axis. The two sides of

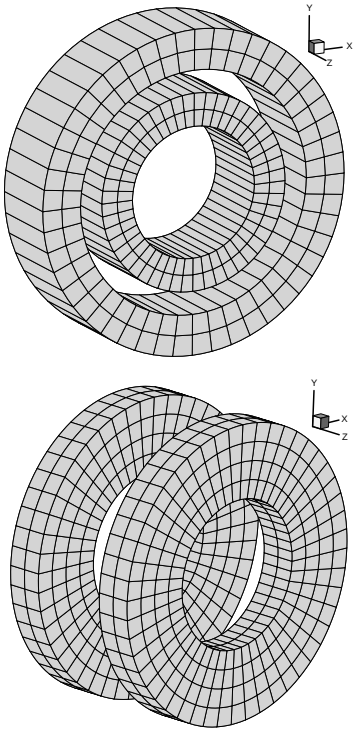

Figure 3: Top: two subdomains with a sliding interface parallel to the axis (inner subdomain has been scaled); bottom: with a sliding interface perpendicular to the axis (subdomains have been separated).

a sliding interface are meshed uniformly in the circumferential direction, while meshes along the longitudinal (of the first kind) or radial (of the second kind) direction are matched and are not necessarily uniform. The resulting cell faces and mortars are either arch-shaped (for the

first kind) or fan-shaped (for the second kind). Figure 4 shows a 2D schematic of the distributions of mortars for the first-kind sliding interface. It can be seen that, at each time instant, a cell face is connected to two mortars and a mortar is associated with one left and one right cell face. This connectivity needs to be updated at every stage of a time-marching scheme.

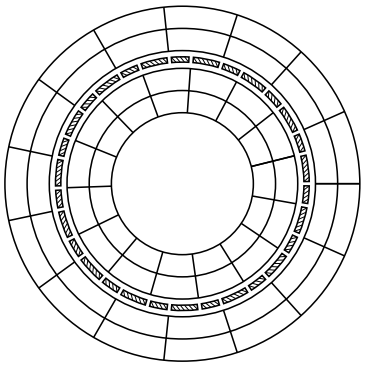

Figure 4: A 2D schematic of distribution of mortars (hatched) for the first-kind sliding interface.

Figure 5 is a schematic of the mapping of an arch-shaped and a fan-shaped cell face from the physical space to a square face (e.g., $0 \leq \xi, \eta \leq 1$) in the computational space, where Ω denotes cell face and Ξ denotes mortar. Meanwhile, a mortar is mapped to a square element (i.e., $0 \leq \xi', \eta' \leq 1$) in the mortar space. This is achieved via 2D isoparametric mapping. Let's assume

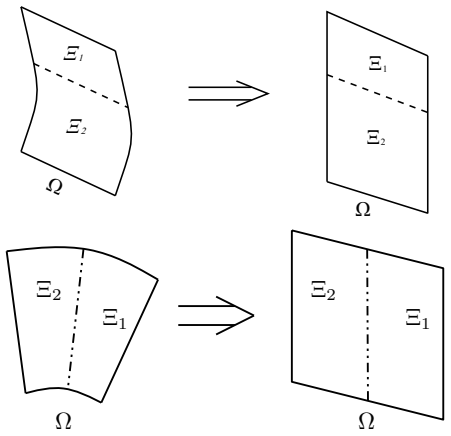

Figure 5: Mapping of curved cell face and mortars to planner ones: top, curved face and mortars in the physical space; bottom, planner face and mortars in the computational space.

that all circular edges are mapped to straight ones in the ξ direction (for cell faces) or ξ' direction (for mortars), while straight edges are mapped to the η (for cell faces) or η' (for mortars) direction. Then, the computational space and mortar space are related as

$$\xi = o(t) + s(t)\xi', \tag{26}$$

$$\eta = \eta', \tag{27}$$

where $o(t)$ is the offset of a mortar with respect to a cell face at time t, and $s(t)$ is the scaling. For example, in Figure 5(a), we have $o_1 = 0$, $s_1 = L^{\Xi_1}/L^{\Omega}$ for Ξ_1, and $o_2 = L^{\Xi_1}/L^{\Omega}$, $s_2 = L^{\Xi_2}/L^{\Omega}$ for Ξ_2, where L represents the physical length of a cell face or a mortar in the circumferential direction.

According to Equation (22), the solution on Ω can be represented as,

$$\mathbf{Q}^{\Omega}(\xi,\eta) = \sum_{j=1}^{N}\sum_{i=1}^{N} \mathbf{Q}^{\Omega}_{i,j} h_i(\xi) h_j(\eta), \tag{28}$$

where $\mathbf{Q}^{\Omega}_{i,j}$ represents the discrete solution at the (i,j)-th SP on Ω, h_i and h_j are the Lagrange bases defined in Equation (20). If we define the same set of SPs on $0 \leq \xi' \leq 1$, $0 \leq \eta' \leq 1$ for each mortar, then solution on each mortar can be reconstructed similarly,

$$\mathbf{Q}^{\Xi}(\xi',\eta') = \sum_{j=1}^{N}\sum_{i=1}^{N} \mathbf{Q}^{\Xi}_{i,j} h_i(\xi') h_j(\eta'), \tag{29}$$

where $\mathbf{Q}^{\Xi}_{i,j}$ is the solution at the (i,j)-th SP on a mortar Ξ.

The procedure for computing $\mathbf{Q}^{\Xi}_{i,j}$ is demonstrated in Figure 6(a). For simplicity, we only show the process on the left side of mortar Ξ. To get the solutions, we require that,

$$\int_0^1 (\mathbf{Q}^{\Xi,L}(\xi',\eta') - \mathbf{Q}^{\Omega}(\xi,\eta)) h_\alpha(\xi') h_\beta(\eta') d\xi' d\eta' = 0, \quad \alpha,\beta = 1,2,...,N. \tag{30}$$

Substituting Equations (26)-(29) into the above equation and evaluating it at each SP on Ξ will give a system of linear equations. The solution of this system when written in matrix form is

$$\mathbf{Q}^{\Xi,L} = \mathbf{P}^{\Omega\to\Xi}\mathbf{Q}^{\Omega} = \mathbf{M}^{-1}\mathbf{S}^{\Omega\to\Xi}\mathbf{Q}^{\Omega}, \tag{31}$$

where $\mathbf{P}^{\Omega\to\Xi} = \mathbf{M}^{-1}\mathbf{S}^{\Omega\to\Xi}$ is the projection matrix from Ω to Ξ. The matrices $\mathbf{M}$ and $\mathbf{S}^{\Omega\to\Xi}$ have the following expressions,

$$\mathbf{M} = \int_0^1\int_0^1 h_\alpha(\xi') h_\beta(\eta') h_i(\xi') h_j(\eta') d\xi' d\eta', \quad \alpha,\beta,i,j = 1,2,...,N, \tag{32}$$

$$\mathbf{S}^{\Omega\to\Xi} = \int_0^1\int_0^1 h_\alpha(\xi') h_\beta(\eta') h_i(o+s\xi') h_j(\eta') d\xi' d\eta', \quad \alpha,\beta,i,j = 1,2,...,N, \tag{33}$$

where o and s are the offset and scaling of Ξ with respect to Ω. It is important to note that o and s are time-dependent.

The right solution vector $\mathbf{Q}^{\Xi,R}$ can be computed in the same way. With both left and right solutions on a mortar, the Rusanov solver [27] is employed to compute the common inviscid flux $\mathbf{F}^{\Xi}_{inv}$. This common flux is then transformed to the computational flux as $\widetilde{\mathbf{F}}^{\Xi}_{inv}$ according to Equation (16).

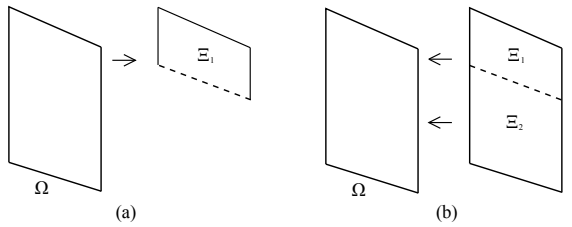

Figure 6: Projection between cell face and mortar: (a) from left face to left side of mortar, (b) from two mortars back to the associated left face.

As shown in Figure 6(b), to project the common inviscid fluxes $\widetilde{\mathbf{F}}^{\Xi_1}_{inv}$ and $\widetilde{\mathbf{F}}^{\Xi_2}_{inv}$ back to a cell face Ω, we require that,

$$\int_{\eta=0}^{\eta=1}\int_{\xi=0}^{\xi=o_2}(\widetilde{\mathbf{F}}^{\Omega}_{inv}(\xi,\eta)-\widetilde{\mathbf{F}}^{\Xi_1}_{inv}(\xi',\eta'))h_\alpha(\xi)h_\beta(\eta)d\xi d\eta + \int_0^1\int_{\xi=o_2}^{\xi=1}(\widetilde{\mathbf{F}}^{\Omega}_{inv}(\xi,\eta)-\widetilde{\mathbf{F}}^{\Xi_2}_{inv}(\xi',\eta'))h_\alpha(\xi)h_\beta(\eta)d\xi d\eta = 0,\ \ \alpha,\beta=1,2,...,N, \tag{34}$$

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

$$\begin{aligned}\widetilde{\mathbf{F}}^{\Omega}_{inv} &= \mathbf{P}^{\Xi_1\to\Omega}\widetilde{\mathbf{F}}^{\Xi_1}_{inv}+\mathbf{P}^{\Xi_2\to\Omega}\widetilde{\mathbf{F}}^{\Xi_2}_{inv}\\ &= s_1\mathbf{M}^{-1}\mathbf{S}^{\Xi_1\to\Omega}\widetilde{\mathbf{F}}^{\Xi_1}_{inv}+s_2\mathbf{M}^{-1}\mathbf{S}^{\Xi_2\to\Omega}\widetilde{\mathbf{F}}^{\Xi_2}_{inv},\end{aligned} \tag{35}$$

where the matrix $\mathbf{M}$ is identical to that in Equation (31), and matrices $\mathbf{S}^{\Xi_1\to\Omega}$ and $\mathbf{S}^{\Xi_2\to\Omega}$ are simply transposes of $\mathbf{S}^{\Omega\to\Xi_1}$ and $\mathbf{S}^{\Omega\to\Xi_2}$, respectively.

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

$$\mathbf{Q}^{\Xi}=\frac{1}{2}(\mathbf{Q}^{\Xi,L}+\mathbf{Q}^{\Xi,R}). \tag{36}$$

This common solution is then projected back to cell faces in the same procedure as for the inviscid flux in Equation (35). After that, solution gradients and viscous fluxes are updated on both side of the interfaces. The viscous fluxes $\widetilde{\mathbf{F}}^{\Omega}_{vis}$ on cell faces are projected to mortars in the same way as Equation (31). The common viscous flux $\widetilde{\mathbf{F}}^{\Xi}_{vis}$ on a mortar is the average of left and right viscous fluxes,

$$\widetilde{\mathbf{F}}^{\Xi}_{vis}=\frac{1}{2}(\widetilde{\mathbf{F}}^{\Xi,L}_{vis}+\widetilde{\mathbf{F}}^{\Xi,R}_{vis}). \tag{37}$$

The final step is to project $\widetilde{\mathbf{F}}^{\Xi}_{vis}$ back to cell faces, which is identical to the process in Equation (35).

Since uniform mesh is used for cell faces on the sliding interface, the $\mathbf{S}$ matrix only needs 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 (32) and (33), one can use any numerical or analytical integration techniques, for example, the Clenshaw-Curtis quadrature method [2].

NUMERICAL TESTS

In this section we test the accuracy of the solver on both inviscid and viscous flows, and then apply this method to study flow around a rotating elliptic cylinder and flow over a stream-wise rotating cube.

Euler Vortex Flow

Euler vortex flow is widely used for accuracy tests of inviscid flow solvers, one of such examples can be found in [6]. In Euler vortex flow problem, an isentropic vortex is superimposed to and convected by a uniform background flow. The flow field of a two-dimensional Euler vortex flow in an infinite domain at time t can be described as,

$$u=U_\infty\left\{\cos\theta-\frac{\varepsilon y_r}{r_c}\exp\left(\frac{1-x_r^2-y_r^2}{2r_c^2}\right)\right\} \tag{38}$$

$$v=U_\infty\left\{\sin\theta+\frac{\varepsilon x_r}{r_c}\exp\left(\frac{1-x_r^2-y_r^2}{2r_c^2}\right)\right\} \tag{39}$$

$$\rho=\rho_\infty\left\{1-\frac{(\gamma-1)(\varepsilon M_\infty)^2}{2}\exp\left(\frac{1-x_r^2-y_r^2}{r_c^2}\right)\right\}^{\frac{1}{\gamma-1}} \tag{40}$$

$$p=p_\infty\left\{1-\frac{(\gamma-1)(\varepsilon M_\infty)^2}{2}\exp\left(\frac{1-x_r^2-y_r^2}{r_c^2}\right)\right\}^{\frac{\gamma}{\gamma-1}} \tag{41}$$

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

$$x_r=x-x_0-\bar{u}t, \tag{42}$$

$$y_r=y-y_0-\bar{v}t, \tag{43}$$

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.

In the present study, 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: $\varepsilon = 1$, $r_c = 1$, is superimposed to the mean flow. The domain size is $0 \le x, y \le 10$ (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 all three coordinate directions.

Three meshes with 172, 1376 and 11,008 cells are used for accuracy tests. Figure 7 shows the overall mesh with 1376 cells and the mesh on its sliding interface. The overall mesh has 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. For all three cases, the inner part is set to rotate at an angular speed of $\omega = 1.0$.

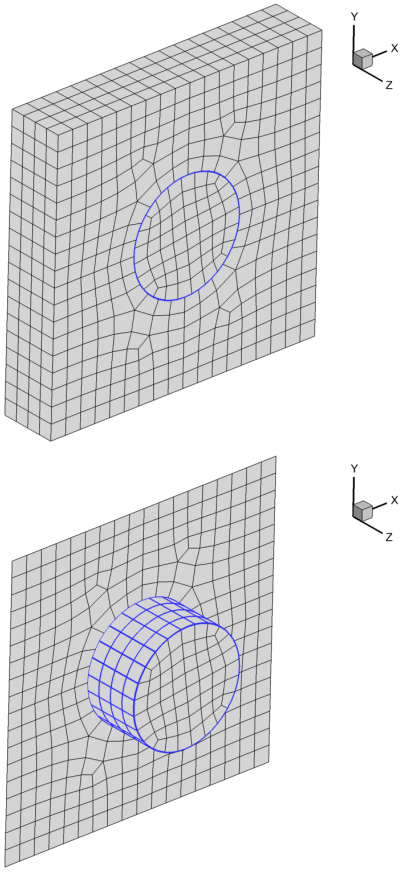

Figure 7: Mesh with 1376 cells at a time instant for Euler vortex flow simulation: top, volume mesh; bottom, some surface mesh (blue color indicates sliding interface).

Figure 8 shows the density contours at $t = 2$ from the fourth-order scheme. At this time instant, the vortex is traveled to a position with its center right on the sliding interface. As we can see, the nonconforming sliding interface does not cause any alteration to the shape of the vortex.

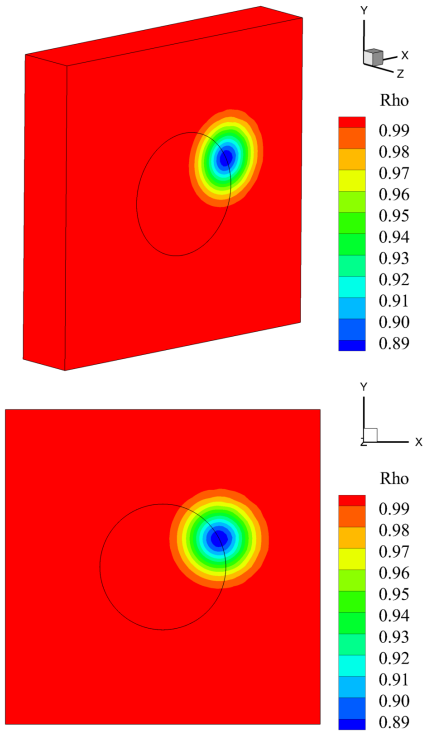

Figure 8: Contours of density at time instant $t = 2$ for the Euler vortex flow.

Furthermore, Table 1 and Table 2 reports 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 method gives good orders of accuracy.

Table 1: Errors and orders of accuracy of the 3^{rd} order scheme on the Euler vortex flow.

cells	L1 error	order	L2 error	order
172	1.57E-3	-	1.68E-3	-
1376	1.87E-4	3.07	2.34E-4	2.85
11008	2.72E-5	2.93	3.71E-5	2.75

Table 2: Errors and orders of accuracy of the 4^{th} order scheme on the Euler vortex flow.

cells	L1 error	order	L2 error	order
172	5.00E-4	-	4.84E-4	-
1376	2.31E-5	4.44	2.72E-5	4.15
11008	8.98E-7	4.56	1.09E-6	4.39

Laminar Taylor-Couette Flows

This test includes two cases, one with a sliding interface parallel to the z direction and the other with a sliding interface perpendicular to the z axis. Periodic boundary condition is applied in the spanwise direction. The exact solution for the circumferential velocity has the following relation to radius r,

$$v_\theta = \omega_i r_i \frac{r_o/r - r/r_o}{r_o/r_i - r_i/r_o}. \tag{44}$$

Figure 9 shows two views of a mesh whose sliding interface is parallel to the z axis. The inner boundary has a radius of $r_i = 1$, and the outer one has radius of $r_o = 2$. The sliding interface locates at $r = 1.5$. The inner subdomain rotates at an angular speed of 1.0, while the outer subdomain is fixed.

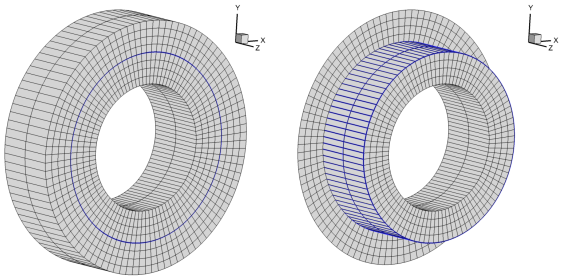

Figure 9: A mesh with one parallel sliding interface for the Taylor-Couette flow simulation.

The steady state density contours from the third-order scheme on the coarsest mesh are shown Figure 10. We see that the contours are a series of concentric circles, which is consistent with our expectation. Again, The sliding interface does not cause any contamination to the solutions.

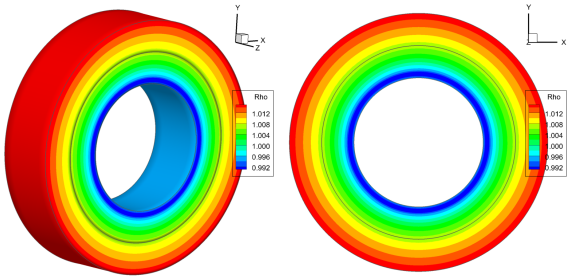

Figure 10: Density contours of the Taylor-Couette flow with one parallel sliding interface.

The errors are computed based on the x velocity component, and the results together with the orders of accuracy are shown in Table 3 and Table 4. It is obvious that the solver gives good accuracies on this viscous flow test with a parallel sliding interface.

Table 3: Accuracy of the third-order scheme on the Taylor-Couette flow with one parallel sliding interface.

cells	L1 error	order	L2 error	order
192	1.87E-3	-	1.43E-3	-
1536	2.17E-4	3.10	1.86E-4	2.94
12288	2.00E-5	3.27	2.32E-5	2.97

Table 4: Accuracy of the fourth-order scheme on the Taylor-Couette flow with one parallel sliding interface.

cells	L1 error	order	L2 error	order
192	1.87E-4	-	1.23E-5	-
1536	1.25E-5	3.90	9.01E-7	3.77
12288	9.08E-7	3.85	6.86E-8	3.75

Figure 11 shows two views of the coarsest mesh for the second case. The domain has been split into two subdomains at the mid-spanwise location. The background subdomain is fixed, on which the foreground subdomain slides. The velocities on the outer boundaries are set to zero. Velocity on the background inner boundary is prescribed, and on the foreground one is computed from no-slip boundary condition.

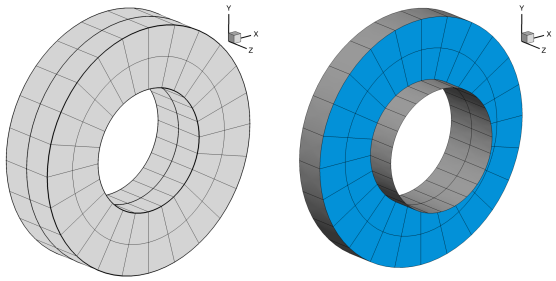

Figure 11: A mesh with one perpendicular sliding interface for the simulation of Taylor-Couette flow.

The steady state density contours are plotted in Figure 12, which are again from a third-order scheme on the coarsest mesh. It is obvious that the solver gives correct solution both inside the domain and on all boundaries including the sliding interfaces.

The errors and orders of accuracy are reported in Tables 5 and 6. Again, the solver gives the desired orders of accuracy for this case with a sliding interface perpendicular to the rotating axis.

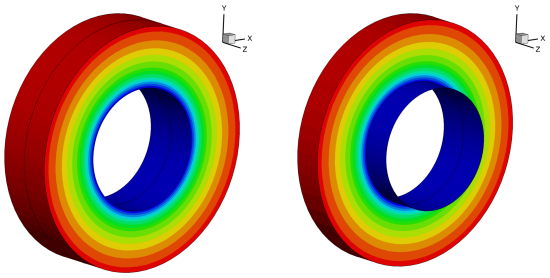

Figure 12: Density contours of the Taylor-Couette flow with one perpendicular sliding interface.

Table 5: Accuracy of the third-order scheme on the Taylor-Couette flow with one perpendicular sliding interface.

cells	L1 error	order	L2 error	order
96	5.73E-4	-	8.31E-4	-
768	7.36E-5	2.96	1.09E-4	2.93
6144	8.35E-6	3.10	1.29E-5	3.01

Table 6: Accuracy of the fourth-order scheme on the Taylor-Couette flow with one perpendicular sliding interface.

cells	L1 error	order	L2 error	order
96	7.04E-5	-	9.38E-5	-
768	4.78E-6	3.88	6.64E-6	3.77
6144	2.82E-7	3.85	4.24E-7	3.79

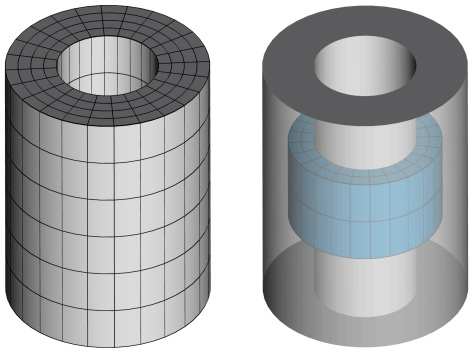

Figure 13: Schematic of the mesh for a transitional Taylor-Couette flow simulation with three sliding interfaces.

A Transitional Taylor-Couette Flow

Figure 13 shows a schematic of the mesh for this case. The overall computational domain is defined as $1 \leq r \leq 2$ and $0 \leq z \leq 6$. Periodic boundary condition is applied along the z direction. Isothermal wall boundary conditions are applied on the inner and the outer boundaries. The inner subdomain rotates at an angular speed of 1, while the same angular speed is prescribed on the inner boundaries of the outer subdomain. The resulting Reynolds number is 2,000. The aim of this test is to see if flow transition from laminar to turbulent can be well captured when nonconforming sliding interfaces present.

Figure 14 shows the development of the flow field by visualizing isosurfaces of the Q-criterion [10]. The flow field is initialized as a uniform flow. It is seen that at this high Reynolds number, the flow gradually lose its stability due to the weak viscous effects compared to the low Reynolds number flows. Three dimensionality becomes obvious after about four rotating periods as shown in Figure 14(a). More and more 3D flow structures appear in Figures 14(b) and (c) as the rotation continues, however, those structures are still very coherent and well organized. The development to turbulent flow is evident in Figures 14(d) to (f), in which large vortex structures finally break down into small flow structures. This case clearly shows that the solver can resolve flow structures very well for this transitional flow even on a very coarse grid.

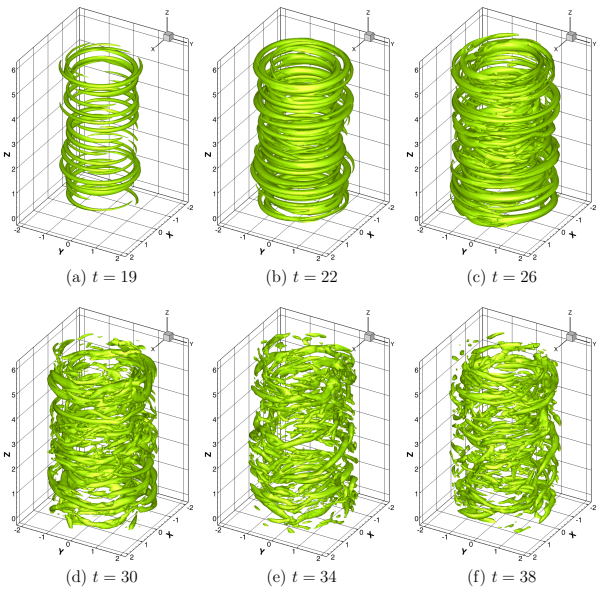

Figure 14: Isosurfaces of the Q-criterion $Q_{cr} = 1$ for a transitional Taylor-Couette flow at $Re = 2,000$ with three sliding interfaces.

Flow over a Rotating Elliptic Cylinder

To further verify the solver, we simulate flow over an elliptic cylinder in this section. Maruoka [22] and Zhang et al. [36] studied incompressible flow over a rotating elliptic cylinder using Finite Element and Finite Volume method, 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.

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 counterclockwise at an angular speed of $\omega = 0.5\pi$. The Reynolds number based on freestream velocity and major axis length is 200. Figure 15 shows a schematic of the computational domain. The top and bottom boundaries are set as far-field boundaries. 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.

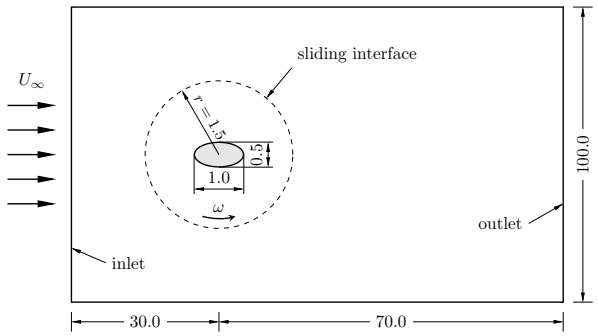

Figure 15: Schematic of the computational domain for flow over a rotating elliptic cylinder (not to scale).

The inner rotating domain has a radius of 1.5 and is meshed with 3072 cells. The rest of the domain is stationary and has 19496 cells in all. Mesh refinement are performed around the leading and trailing edges, and in the wake region. Figure 16 shows part of the mesh around the cylinder. The first layer of the mesh around the airfoil has a thickness of about 0.005, and the maximum aspect ratio is around 2. The non-dimensional time step size $\Delta t U_\infty / L$ for the simulation is set to 1.0×10^{-4}, where L is the major axis and U_∞ is the free-stream velocity.

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 [22] and Zhang et al. [36], the fully developed flow takes a periodic pattern as the cylinder rotates. The lift and drag coefficients in one period are shown in Figure 17. It is seen that the present result agrees very well with the previous results. Figure 18 shows the contour of vorticity component in the z direction in the middle plane at a time instant. We see that at the given Reynolds number, the flow is laminar and a vortex street with vortices rotating at alternative directions formed behind the cylinder.

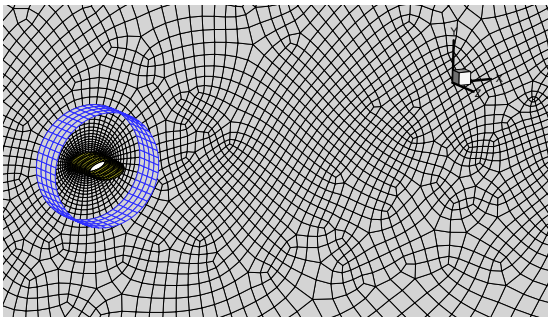

Figure 16: A local view of surface mesh around the elliptic cylinder (blue color indicates sliding interface, yellow color indicates cylinder surface).

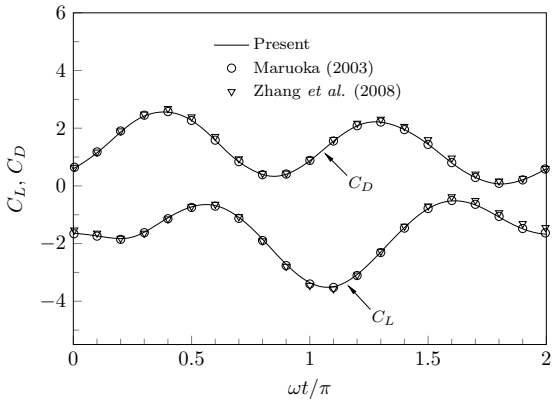

Figure 17: Lift and drag coefficients for flow over an elliptic cylinder.

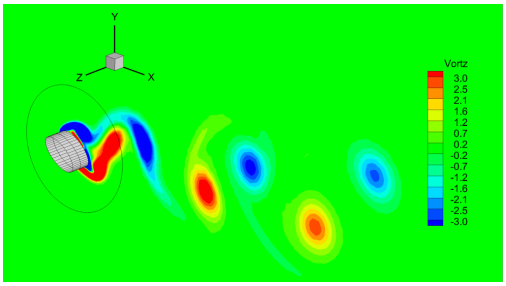

Figure 18: Streamlines and vorticity contours (blue means negative value, red means positive) for flow over a counterclockwise rotating elliptic cylinder (big circle is sliding interface).

Flows around a Rotating Cube

This section gives some preliminary results of flows over a stream-wise rotating cube. Figure 19 shows part of the mesh used for this simulation. The cube has a length of $L = 1$ in each direction, and locates $10L$ away from the inlet, $30L$ from the outlet. The computational domain has a size of $40L \times 30L \times 8L$, and is decomposed into two parts: one that inside the blue surface and rotates at angular speed of $\omega = 0.5\pi$, the other that takes the rest of the domain and stays stationary. The inner domain has a total number of 17982 mesh cells, and the outer has 29295. Mesh has been refined around the cube as well as in the near weak region. Dirichlet boundary condition is applied at the inlet, where the Mach number is $Ma = 0.2$. Pressure boundary condition is used at the outlet. The top and bottom boundaries are set to have symmetric boundary conditions. Periodic boundary condition is used in the spanwise direction. Two Reynolds number based on the freestream velocity and cube length, $Re = 100$ and $1,000$, are tested.

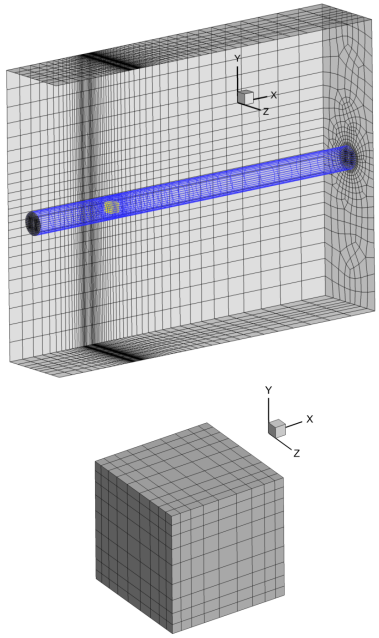

Figure 19: Mesh for flow over a cube: top, some surface mesh of the domain (blue color indicates sliding interface, yellow color indicates cube surface); bottom, mesh on the cube surface.

Figure 20 shows the Q criterion [10] colored by pressure at several time instants for $Re = 100$. It can be seen that as the cube rotates, vortex structures will appear around each corner of the cube, and form a spiral pattern. It is also very interesting to notice that a vortex core shows up slightly downstream of the cube with its size and location almost irrelevant to the rotation. The pressure distribution indicates that the front surface of the cube experiences high pressure force, which is consistent with the fact the flow goes to stagnation within that area.

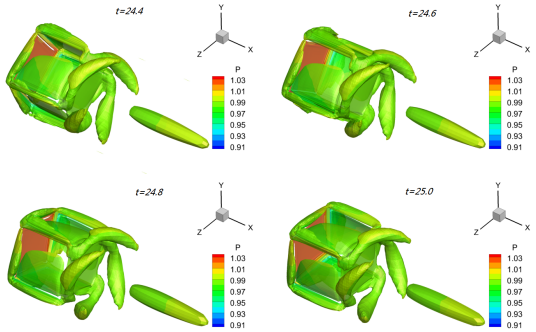

Figure 20: Isosurface of the Q criterion (colored by pressure) for flow over a rotating cube at $Re = 100$.

Figures 21 shows the flow fields for $Re = 1,000$. It is obvious that the developments of flow structures are quite different from the previous case. In the early rotating period ($t \leq 6$), it is seen that hair-pin type vortices are generated from the corners of the cube. These vortices are then convected downstream to form a few rows of well

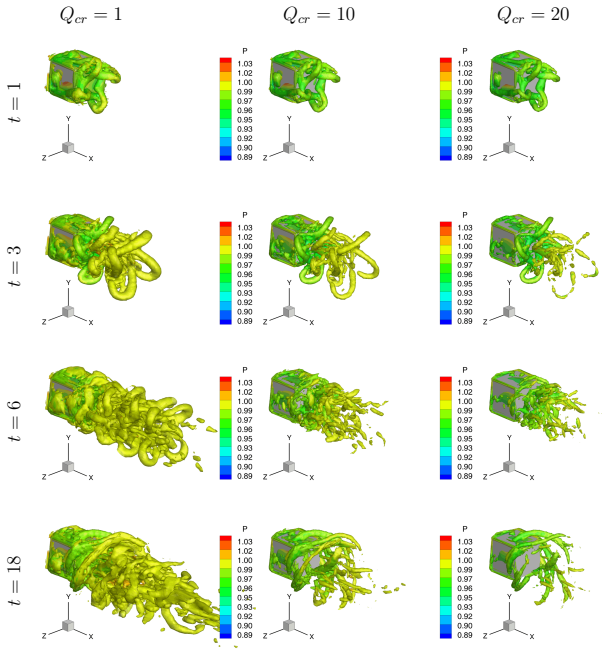

Figure 21: Isosurfaces of the Q-criterion (colored by pressure) for flow over a rotating cube at $Re = 1,000$.

organized patterns. However, in the later rotating periods

($t \geq 18$), those structures are not able to sustain themselves due to insufficient viscous forces, and they finally break down into small turbulent structures. The converged flow ($t \geq 18$) shows that the most coherent structures are still the spiral type vortices similar to those that have been observed in the previous case. However, due to smaller viscous effects, these structures show thinner and longer shapes than their counterparts in the low Reynolds number case.

CONCLUSIONS

We have successfully developed a high-order accurate three-dimensional unsteady flow solver using a novel sliding-mesh mesh method and the spectral difference method. This solver has been shown capable of preserving the high-order accuracy of the spectral difference method on nonconforming sliding grids. This 3D solver is also able to handle multiple sliding interfaces with different orientations. Future work of our research includes parallelization of the 3D solver and validation on transitional and turbulent flows.

ACKNOWLEDGMENT

The authors are supported by an ONR Young Investigator Program (YIP) award to Chunlei Liang administrated by Dr. Ki-Han Kim.

REFERENCES

[1] Balan, A., May, G., Schöberl, J., 2012. A stable high-order spectral difference method for hyperbolic conservation laws on triangular elements. Journal of Computational Physics 231 (5), 2359–2375.

[2] Canuto, C., Hussaini, M. Y., Quarteroni, A., Zang, T. A., 1987. Spectral Methods in Fluid Mechanics. Springer-Verlag, New York.

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

[4] Cockburn, B., Karniadakis, G. E., Shu, C.-W. (Eds.), 2011. Discontinuous Galerkin Methods: Theory, Computation and Applications. Vol. 11 of Lecture Notes in Computational Science and Engineering. Springer, New York.

[5] Ekaterinaris, J. A., 2005. High-order accurate, low numerical diffusion methods for aerodynamics. Progress in Aerospace Sciences 41 (3-4), 192–300.

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

[7] Hesthaven, J. S., Warburton, T., 2008. Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications. Vol. 54 of Texts in Applied Mathematics. Springer, New York.

[8] Hirt, C. W., Amsden, A. A., Cook, J. L., 1974. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. Journal of Computational Physics 14, 227–253.

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

[10] Jeong, J., Hussain, F., 1995. On the identification of a vortex. Journal of Fluid Mechanics 285, 69–94.

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

[12] Kopriva, D. A., 1996. A conservative staggered-grid Chebyshev multidomain method for compressible flows. II. A semi-structured method. Journal of Computational Physics 128, 475–488.

[13] Kopriva, D. A., 1998. A staggered-grid multidomain spectral method for the compressible Navier-Stokes equations. Journal of Computational Physics 143, 125–158.

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

[15] Liang, C., Jameson, A., Wang, Z. J., 2009. Spectral difference method for two-dimensional compressible flow on unstructured grids with mixed elements. Journal of Computational Physics 228, 2847–2858.

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

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

[18] Liang, C., Premasuthan, S., Jameson, A., 2009. High-order accurate simulation of low-mach laminar flow past two side-by-side cylinders using spectral difference method. Computers & Structures 87, 812–817.

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

[20] Liu, Y., Vinokur, M., Wang, Z. J., 2006. Spectral

difference method for unstructured grids I: Basic formulation. Journal of Computational Physics 216, 780–801.

[21] Lodato, G., Castonguay, P., Jameson, A., 2014. Structural wall-modeled LES using a high-order spectral difference scheme for unstructured meshes. Flow, Turbulence and Combustion 92 (1-2), 579–606.

[22] Maruoka, A., 2003. Finite element analysis for flow around a rotating body using Chimera method. International Journal of Computational Fluid Dynamics 17 (4), 289–297.

[23] May, G., Jameson, A., 2006. A spectral difference method for the Euler and Navier-Stokes equations on unstructured meshes. AIAA paper 2006-304.

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

[25] Ou, K., Liang, C., Jameson, A., 2010. High-order spectral difference method for the Navier-Stokes equations on unstructured moving deforming grids. AIAA paper 2010-0541.

[26] Parsani, M., Ghorbaniasl, G., Lacor, C., 2011. Validation and application of an high-order spectral difference method for flow induced noise simulation. Journal of Computational Acoustics 19 (03), 241–268.

[27] Rusanov, V. V., 1961. Calculation of interaction of non-steady shock waves with obstacles. Journal of Computational and Mathematical Physics USSR 1, 267–279.

[28] Ruuth, S., 2006. Global optimization of explicit strong-stability-preserving Runge-Kutta methods. Mathematics of Computation 75 (253), 183–207.

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

[30] Wang, Z. J., 2007. High-order methods for the Euler and Navier-Stokes equations on unstructured grids. Progress in Aerospace Sciences 43 (1-3), 1–41.

[31] Wang, Z. J. (Ed.), 2011. Adaptive High-Order Methods in Computational Fluid Dynamics. Vol. 2 of Advances in Computational Fluid Dynamics. World Scientific.

[32] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., et al., 2013. High-order CFD methods: current status and perspective. International Journal for Numerical Methods in Fluids 72 (8), 811–845.

[33] Wang, Z. J., Liu, Y., May, G., Jameson, A., 2007. Spectral difference method for unstructured grids II: Extension to the Euler equations. Journal of Scientific Computing 32, 45–71.

[34] Zhang, B., Liang, C., 2015. 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 295, 147–160.

[35] Zhang, B., Liang, C., Yang, J., Rong, Y., 2016. A 2D parallel high-order sliding and deforming spectral difference method. Computers & Fluids.

[36] Zhang, X., Ni, S., He, G., 2008. A pressure-correction method and its applications on an unstructured Chimera grid. Computers & Fluids 37, 993–1010.