Received: 21 March 2018 | Revised: 4 December 2018 | Accepted: 19 January 2019

DOI: 10.1002/fld.4717

RESEARCH ARTICLE

WILEY

A high-order solver for simulating vortex-induced vibrations using the sliding-mesh spectral difference method and hybrid grids

Zihua Qiu[1,2] | Bin Zhang[2] | Chunlei Liang[2] | Min Xu[1]

[1]School of Astronautics, Northwestern Polytechnical University, Xi'an, China

[2]Department of Mechanical and Aerospace Engineering, The George Washington University, Washington, District of Columbia

Correspondence
Zihua Qiu, School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China.
Email: qiuzh@mail.nwpu.edu.cn

Bin Zhang, Department of Mechanical and Aerospace Engineering, The George Washington University, Washington, DC 20052.
Email: bzh@gwu.edu

Funding information
China Scholarship Council; Office of Naval Research

Summary

We present a high-order solver for simulating vortex-induced vibrations (VIVs) at very challenging situations, for example, VIVs of a row of very closely placed objects with large relative displacements. This solver works on unstructured hybrid grids by employing the high-order tensor-product spectral difference method for quadrilateral grids and the Raviart-Thomas spectral difference method for triangular grids. To deal with the challenging situations where a traditional conforming moving mesh is incapable, we split a computational domain into nonoverlapping subdomains, where each interior subdomain encloses an object and moves freely with respect to its neighbors. A nonuniform sliding-mesh method that ensures high-order accuracy is developed to deal with sliding interfaces between subdomains. A monolithic approach is adopted to seamlessly couple the fluid and solid vibration equations. Moreover, the solver is parallelized to further improve its efficiency on distributed-memory computers. Through a series of numerical tests, we demonstrate that this solver is high-order accurate for both inviscid and viscous flows and has good parallel efficiency, making it ideal for VIV studies.

KEYWORDS

high-order method, hybrid mesh, sliding mesh, spectral difference method, vortex-induced vibrations

1 | INTRODUCTION

Fluid flows can cause various types of vibrations on solid structures,[1] for example, instability vibrations like flutter and galloping on bridges and aircraft wings, resonance vibrations like buffeting and vortex-induced vibration (VIV) on tall buildings and electric power lines, and the auto-rotation of polygon cross sections and helicopter rotors. Among the many types, VIVs on bluff bodies have drawn a wide range of research interest due to their fundamental yet practical nature. VIV occurs when a flow passes a bluff body accompanied by vortex shedding. The unsteady vortices exert oscillatory forces on the body and cause vibrations. When the vortex shedding frequency approaches the natural frequency of the structure, resonance takes place, and the amplitude of vibration increases.

An elastically mounted circular cylinder immersed in an incoming flow has been widely adopted as a model problem for studying VIVs of bluff bodies. The experiment conducted by Feng[2] in 1968 is generally considered one of the first modern

studies on VIVs. In his experiment, a cylinder is restrained to transverse vibrations only. Khalak and Williamson[3,4] experimentally studied the effects of mass and damping on the vibration of a single cylinder. They found that the mass-damping system has different response modes compared to the classical Feng-type response. Mittal and Kumar,[5] Jeon and Gharib,[6] Jauvtis and Williamson,[7] and Prasanth et al[8] studied circular cylinders with two degrees of freedom (DOFs). Their results showed that the magnitude of parallel oscillation is orders of magnitude smaller than that of transverse oscillation. Bernitsas et al[9,10] designed a vortex-induced vibration aquatic clean energy (VIVACE) converter, which converts ocean/river current hydrokinetic energy into usable energy such as electricity. They noticed that VIV is very sensitive to the physical properties of the system and shows an extremely nonlinear response to flow conditions and structural properties. More information on the recent research progress on VIVs can be found, for example, in the review papers by Williamson and Govardhan.[11,12]

When it comes to the numerical simulation of VIVs, there are at least three challenges that need to be better resolved. Firstly, to get the correct fluid force on a bluff body, the sources of the force, ie, the vortices, must be captured as accurately as possible. Since numerical dissipation is detrimental to vortical flow structures, a numerical scheme must therefore introduce as little dissipation as possible to a simulation. However, most of the available solvers so far for VIV simulations are still limited to, at most, second-order accuracy, which introduce strong dissipations. Secondly, to accommodate the displacement of a vibrating body, a computational mesh usually needs to be deformed; however, when the magnitude of vibration is large, the mesh may become very skewed, resulting in excessive numerical errors. In the case of multiple closely placed bodies with relatively large motions, the simulation may even fail on a traditional conforming mesh. Thirdly, VIV is a bidirectional process between fluid and solid: the fluid causes vibration on the solid, and the solid, in turn, modifies the flow. Therefore, accurate and efficient coupling of fluid motion and solid motion needs to be treated very carefully. To the authors' knowledge, there has been no reported solver that thoroughly addressed all these three challenges. The aim of this work is to fill this gap.

To deal with the first challenge, we employ the high-order spectral difference (SD) method,[13-18] but with further extension to unstructured grids with mixed triangular and quadrilateral elements to improve the overall flexibility of mesh distribution. To deal with the second challenge, we split a computational domain into nonoverlapping subdomains that are coupled through nonconforming sliding-mesh interfaces. Zhang and Liang[19] and Zhang et al[20] developed a sliding-mesh spectral difference method for simulating flows around freely rotating objects. The sliding-mesh spectral difference method utilizes curved dynamic mortar elements to couple nonoverlapping sliding meshes. This method was shown to be able to maintain the high-order accuracy of the SD method and is highly efficient. To apply this method to VIV simulations, we further extended it to work with nonuniform sliding-interface meshes. To deal with the third challenge, we employ a monolithic approach and write the fluid and vibration equations into a single conservative system, which is then time marched simultaneously. This coupling approach ensures that there is no lag for information exchange between fluid and solid.

The rest of this paper is organized as follows. Section 2 gives the governing equations for fluid and solid. Section 3 describes the numerical methods for solving these equations. Verification studies and applications are reported in Section 4. Finally, Section 5 concludes this paper.

2 | THE GOVERNING EQUATIONS

2.1 | The Navier-Stokes equations for fluid

For fluid, we consider the following two-dimensional (2D) unsteady Navier-Stokes equations in conservative form:

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

where $\mathbf{Q}_f$ is the vector of conservative variables, and $\mathbf{F}_f$ and $\mathbf{G}_f$ are the flux vectors in the x and y directions, respectively. Note that we have used the subscript "f" to denote "fluid." These terms have the following expressions:

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

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

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

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

As expressed in Equations (3) and (4), the fluxes were divided into two parts: the inviscid and viscous fluxes. Inviscid fluxes are only functions of the conservative variables and have the following detailed expressions:

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

Viscous fluxes are functions of the conservative variables and the gradients. Their expressions are

$$\mathbf{F}_{\text{vis}} = -\begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ u\tau_{xx} + v\tau_{yx} + \mathrm{k}T_x \end{bmatrix}, \quad \mathbf{G}_{\text{vis}} = -\begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + \mathrm{k}T_y \end{bmatrix}, \tag{6}$$

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

To simulate flows on dynamic grids, we take an arbitrary Lagrangian-Eulerian approach. In this approach, the physical time and space $(t, x(t), y(t))$ are mapped to the computational ones (τ, ξ, η), where $\tau = t$ is time, and (ξ, η) represents time-independent computational space. It can be shown that the Navier-Stokes equations will take the following conservative form in the computational space:

$$\frac{\partial \widetilde{\mathbf{Q}}_f}{\partial t} + \frac{\partial \widetilde{\mathbf{F}}_f}{\partial \xi} + \frac{\partial \widetilde{\mathbf{G}}_f}{\partial \eta} = 0, \tag{7}$$

where

$$\begin{bmatrix} \widetilde{\mathbf{Q}}_f \\ \widetilde{\mathbf{F}}_f \\ \widetilde{\mathbf{G}}_f \end{bmatrix} = |\mathcal{J}|\mathcal{J}^{-1}\begin{bmatrix} \mathbf{Q}_f \\ \mathbf{F}_f \\ \mathbf{G}_f \end{bmatrix}. \tag{8}$$

In the above equation, $|\mathcal{J}|$ is the determinant of the Jacobian matrix for mapping, and $\mathcal{J}^{-1}$ is the inverse Jacobian matrix, and their expressions are

$$|\mathcal{J}| = \left|\frac{\partial(t,x,y)}{\partial(\tau,\xi,\eta)}\right| = \begin{vmatrix} 1 & 0 & 0 \\ x_\tau & x_\xi & x_\eta \\ y_\tau & y_\xi & y_\eta \end{vmatrix} = x_\xi y_\eta - x_\eta y_\xi, \tag{9}$$

$$\mathcal{J}^{-1} = \frac{\partial(\tau,\xi,\eta)}{\partial(t,x,y)} = \begin{bmatrix} 1 & 0 & 0 \\ \xi_t & \xi_x & \xi_y \\ \eta_t & \eta_x & \eta_y \end{bmatrix} = \frac{1}{|\mathcal{J}|}\begin{bmatrix} |\mathcal{J}| & 0 & 0 \\ -x_t y_\eta + y_t x_\eta & y_\eta & -x_\eta \\ x_t y_\xi - y_t x_\xi & -y_\xi & x_\xi \end{bmatrix}. \tag{10}$$

It is worth mentioning that in Equation (8), we have used bold vector symbols to represent their scalar components in order to make the expression simpler. Hereinafter, we follow this convention, and a bold symbol could be either a vector or its scalar components, whichever makes the operation permissible.

Ideally, grid motion should not contaminate a flow field. The simplest situation is that a constant free-stream flow stays constant all the time, which is called free-stream preservation. To satisfy free-stream preservation, we simply substitute a constant flow solution into Equation (7). After some algebra, we will arrive at the following equation system:

$$\begin{cases} \frac{\partial(|\mathcal{J}|\xi_x)}{\partial\xi} + \frac{\partial(|\mathcal{J}|\eta_x)}{\partial\eta} = 0, & (11) \\ \frac{\partial(|\mathcal{J}|\xi_y)}{\partial\xi} + \frac{\partial(|\mathcal{J}|\eta_y)}{\partial\eta} = 0, & (12) \\ \frac{\partial|\mathcal{J}|}{\partial t} + \frac{\partial(|\mathcal{J}|\xi_t)}{\partial\xi} + \frac{\partial(|\mathcal{J}|\eta_t)}{\partial\eta} = 0. & (13) \end{cases}$$

This system is only related to the geometric variables and is independent of the flow field. For this reason, it is usually called the geometric conservation law (GCL).[21] These GCL equations also need to be solved together with Equation (7) on a dynamic grid.

2.2 | The oscillator equation for solid

If we limit the motion to the vertical (ie, y) direction only, then the vibration of a solid is governed by the following driven damped harmonic oscillator equation:

$$m\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + c\frac{\mathrm{d}y}{\mathrm{d}t} + ky = F_L, \tag{14}$$

where m and y are the mass and the vertical displacement of a solid, respectively, c is the damping coefficient, k is the spring coefficient, and F_L is the lift (ie, the vertical force) from the fluid. Figure 1 shows a schematic of the parameters for a vibrating circular cylinder system.

2.3 | The coupled equations

To couple the solid equation with the fluid equations (with first-order derivatives in time), we rewrite Equation (14) into the following two first-order differential equations:

$$\begin{cases} \frac{\mathrm{d}y}{\mathrm{d}t} = \dot{y}, & (15) \\ \frac{\mathrm{d}\dot{y}}{\mathrm{d}t} = \frac{1}{m}F_L - \frac{k}{m}y - \frac{c}{m}\dot{y}. & (16) \end{cases}$$

This system can be further written into a residual form, ie,

$$\frac{\partial\mathbf{Q}_s}{\partial t} = \mathcal{R}_s(\mathbf{Q}_f, \mathbf{Q}_s), \tag{17}$$

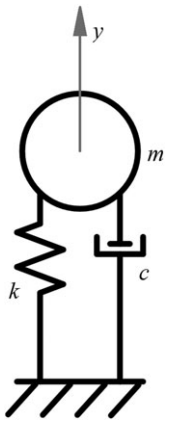

FIGURE 1 Diagram of a mass-spring-damper system for a circular cylinder

where

$$\mathbf{Q}_s = [y,\ \dot{y}]^{\mathsf{T}}, \tag{18}$$

$$\mathcal{R}_s = \left[\dot{y},\ \frac{1}{m}F_L - \frac{k}{m}y - \frac{c}{m}\dot{y}\right]^{\mathsf{T}} \tag{19}$$

are the solution vector and the residual vector, respectively. Note that we have used the subscript "s" to denote solid.

Similarly, the fluid equation system (ie, Equation (7)) can also be written into the following residual form:

$$\frac{\partial \widetilde{\mathbf{Q}}_f}{\partial t} = \mathcal{R}_f(\mathbf{Q}_f, \mathbf{Q}_s), \tag{20}$$

where

$$\mathcal{R}_f = -\left(\frac{\partial \widetilde{\mathbf{F}}_f}{\partial \xi} + \frac{\partial \widetilde{\mathbf{G}}_f}{\partial \eta}\right) \tag{21}$$

is the residual vector of the fluid equation.

Equations (17) and (20) can now be combined into a single monolithic conservative system as

$$\frac{\partial \mathbf{Q}}{\partial t} = \mathcal{R}(\mathbf{Q}), \tag{22}$$

where

$$\mathbf{Q} = \begin{bmatrix} \widetilde{\mathbf{Q}}_f \\ \mathbf{Q}_s \end{bmatrix},\ \mathcal{R}(\mathbf{Q}) = \begin{bmatrix} \mathcal{R}_f(\mathbf{Q}_f, \mathbf{Q}_s) \\ \mathcal{R}_s(\mathbf{Q}_f, \mathbf{Q}_s) \end{bmatrix} \tag{23}$$

are the coupled vectors of conservative variables and the residual, respectively. Equation (22) is the system that we are going to march over time numerically.

3 | NUMERICAL METHODS

3.1 | The SD method for quadrilateral grids

Each quadrilateral element is first mapped from the physical space to a standard unit square element in the computational space using isoparametric mapping. To facilitate the construction of solution and flux polynomials, we then define solution points (SPs) and flux points (FPs) within each standard element. Figure 2 shows a schematic of the distribution of SPs and FPs for a third-order SD scheme. Generally, for an Nth-order scheme, N SPs and $(N + 1)$ FPs are defined along each coordinate direction. In the present implementation, the SPs are chosen as the following Gauss points:

$$X_s = \frac{1}{2}\left[1 - \cos\left(\frac{2s-1}{2N}\pi\right)\right],\ s = 1, 2, \ldots, N. \tag{24}$$

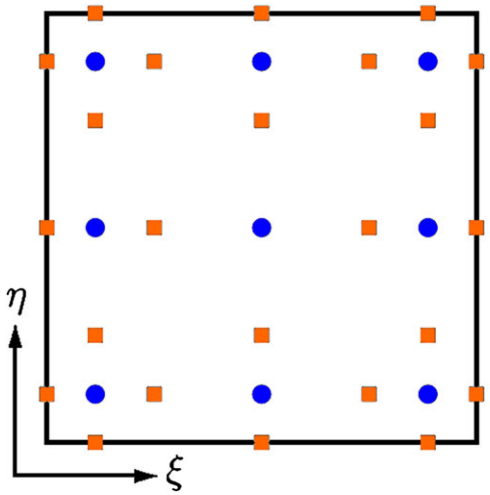

FIGURE 2 Schematic of the distribution of solution points (blue circles) and flux points (orange squares) for a third-order spectral difference scheme on a quadrilateral element

The FPs (denoted by $X_{s+1/2}$, where $s = 0, 1, \ldots, N$) are chosen as the $(N-1)$ Legendre points (ie, roots of the $(N-1)$th Legendre polynomial) plus two end points, where the nth Legendre polynomial is defined as

$$P_n(\xi) = \frac{2n-1}{n}(2\xi - 1)P_{n-1}(\xi) - \frac{n-1}{n}P_{n-2}(\xi), \quad \text{with } P_{-1} = 0,\ P_0 = 1. \tag{25}$$

Next, the following Lagrange interpolation bases are readily constructed at each SP and FP, respectively, ie,

$$h_i(X) = \prod_{s=1, s\neq i}^{N} \left(\frac{X - X_s}{X_i - X_s}\right), \tag{26}$$

$$l_{i+1/2}(X) = \prod_{s=0, s\neq i}^{N} \left(\frac{X - X_{s+1/2}}{X_{i+1/2} - X_{s+1/2}}\right). \tag{27}$$

The solution and flux polynomials are constructed using tensor products of the above bases, ie,

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

$$\widetilde{\mathbf{F}}_f(\xi, \eta) = \sum_{j=1}^{N}\sum_{i=0}^{N}(\widetilde{\mathbf{F}}_f)_{i+1/2,j}l_{i+1/2}(\xi)h_j(\eta), \tag{29}$$

$$\widetilde{\mathbf{G}}_f(\xi, \eta) = \sum_{j=0}^{N}\sum_{i=1}^{N}(\widetilde{\mathbf{G}}_f)_{i,j+1/2}h_i(\xi)l_{j+1/2}(\eta), \tag{30}$$

where $(\widetilde{\mathbf{Q}}_f)_{i,j}$ is the discrete solution at the (i,j)th SP, and $(\widetilde{\mathbf{F}}_f)_{i+1/2,j}$ and $(\widetilde{\mathbf{G}}_f)_{i,j+1/2}$ are the discrete fluxes at the corresponding FPs.

The above constructed solution and flux polynomials are only element-wise continuous, but discontinuous across cell boundaries. To ensure conservation and stability, a Riemann solver is employed to compute the common inviscid fluxes at cell boundaries. In this work, the Rusanov solver[22] has been employed for this purpose. The common viscous fluxes are computed from the common solutions and common gradients that are algebraic averages of the left and right values. Finally, the residual is computed by a direct differentiation of the continuous flux polynomials.

3.2 | The SDRT method for triangular grids

Similar to the SD method on quadrilateral grids, we map each triangular grid element to a standard unit equilateral triangular element using isoparametric mapping. Since flux interpolation in the Raviart-Thomas (RT) space stabilizes the SD method on triangular grids,[18] this method is called the SDRT method. Figure 3 shows a schematic of the distributions of SPs and FPs for the third-order SDRT schemes. Generally, for an Nth-order SDRT scheme, the number of SPs is

$$N_{\text{SP}} = \frac{N(N+1)}{2}. \tag{31}$$

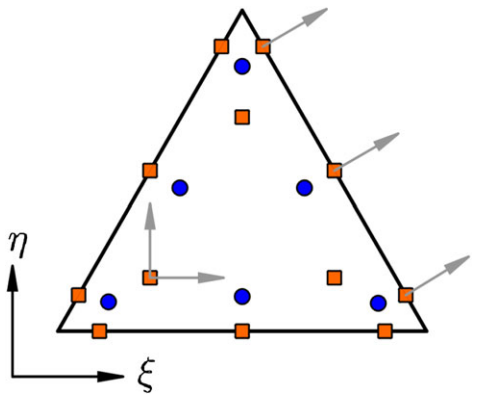

FIGURE 3 Distribution of solution points (blue circles) and flux points (orange squares) for a third-order SDRT scheme

The vector of computational fluid variables in the SDRT method is represented by the following polynomial:

$$\widetilde{\mathbf{Q}}_f(\xi,\eta)=\sum_{i=1}^{N_{\mathrm{SP}}}(\widetilde{\mathbf{Q}}_f)_i L_i(\xi,\eta), \tag{32}$$

where $(\widetilde{\mathbf{Q}}_f)_i$ is the discrete solution at the ith SP, and $L_i(\xi,\eta)$ is a 2D Lagrange interpolation basis function at the ith SP with the following expression:

$$L_i(\xi,\eta)=\sum_{\alpha=0}^{N-1}\sum_{\beta=0}^{\alpha}a_{\alpha,\beta}^{(i)}\xi^{\beta}\eta^{\alpha-\beta}, \tag{33}$$

where the coefficients $a_{\alpha,\beta}^{(i)}$'s are obtained by solving a system of equations following the definition

$$L_i(\xi_j,\eta_j)=\delta_{ij},\ \text{for } j=1,2,\ \ldots,N_{\mathrm{SP}}, \tag{34}$$

where (ξ_j,η_j) are coordinates of the jth SP. This interpolation basis is of degree $(N-1)$ in both ξ and η.

For the flux polynomials, the SDRT method employs vector interpolation bases in the RT space. The constructed flux polynomials can be expressed as

$$\begin{bmatrix}\widetilde{\mathbf{F}}_f(\xi,\eta)\\ \widetilde{\mathbf{G}}_f(\xi,\eta)\end{bmatrix}=\sum_{i=1}^{N_{\mathrm{DOF}}}(\hat{\mathbf{F}}_f)_i\vec{\Psi}_i(\xi,\eta), \tag{35}$$

where N_{DOF} is the total number of DOFs at the FPs. For an Nth-order SDRT scheme, this number is

$$N_{\mathrm{DOF}}=N(N+2). \tag{36}$$

The DOFs are denoted by gray arrows in Figure 3: an FP on the boundaries has one DOF, and an FP in the interior has two DOFs. The scalar flux (ie, $(\hat{\mathbf{F}}_f)_i$) in Equation (35) at the ith DOF is defined as

$$(\hat{\mathbf{F}}_f)_i=\left((\widetilde{\mathbf{F}}_f)_i,(\widetilde{\mathbf{G}}_f)_i\right)\cdot\vec{\mathbf{s}}_i, \tag{37}$$

with $(\widetilde{\mathbf{F}}_f)_i$ and $(\widetilde{\mathbf{G}}_f)_i$ being the computational fluxes in the ξ and η directions, respectively. Finally, $\vec{\Psi}_i(\xi,\eta)$ is a degree N interpolation polynomial vector in the RT space with the following expression:

$$\vec{\Psi}_i(\xi,\eta)=\begin{bmatrix}\left(\sum_{\alpha=0}^{N-1}\sum_{\beta=0}^{\alpha}b_{\alpha,\beta}^{(i)}\xi^{\beta}\eta^{\alpha-\beta}\right)+\xi\left(\sum_{k=0}^{N-1}d_k^{(i)}\xi^k\eta^{N-k-1}\right)\\ \left(\sum_{\alpha=0}^{N-1}\sum_{\beta=0}^{\alpha}c_{\alpha,\beta}^{(i)}\xi^{\beta}\eta^{\alpha-\beta}\right)+\eta\left(\sum_{k=0}^{N-1}d_k^{(i)}\xi^k\eta^{N-k-1}\right)\end{bmatrix}. \tag{38}$$

There are a total number of N_{DOF} unknown coefficients (ie, the b's, c's, and d's) in the above expression, and they are obtained by solving the following equation system:

$$\vec{\Psi}_i(\xi_j,\eta_j)\cdot\vec{\mathbf{s}}_j=\delta_{ij},\ \text{for } j=1,2,\ \ldots,N_{\mathrm{DOF}}, \tag{39}$$

where (ξ_j,η_j) are the coordinates of the jth DOF. In this work, the $\vec{\mathbf{s}}$ vectors are chosen as the unit normals on cell boundaries and either $(1,0)$ or $(0,1)$ in the cell interior, as shown in Figure 3. The flux polynomials in this way are of degree N in both ξ and η. The common solution and common fluxes on cell boundaries are computed in a similar way as for quadratic cells. Again, the final residual is computed by a direct differentiation of the continuous flux polynomials.

3.3 | Mesh movement control

We take the domain in Figure 4 to explain how mesh movement is controlled. The overall domain in this example is split into three subdomains by two sliding interfaces. The two side subdomains are static, whereas the middle one is dynamic to accommodate the motion of the cylinder. The middle subdomain is further divided into three virtual regions: in region I, the mesh is rigid and moves together with the cylinder; in region II, the mesh is deformed; and in region III, the mesh stays stationary.

Assume the initial position of a grid point is (x_0,y_0) and the cylinder center is at (x_c,y_c). We first define an intermediate variable on the initial mesh as

$$r=(|y_0-y_c|-d_1)/(d_2-d_1), \tag{40}$$

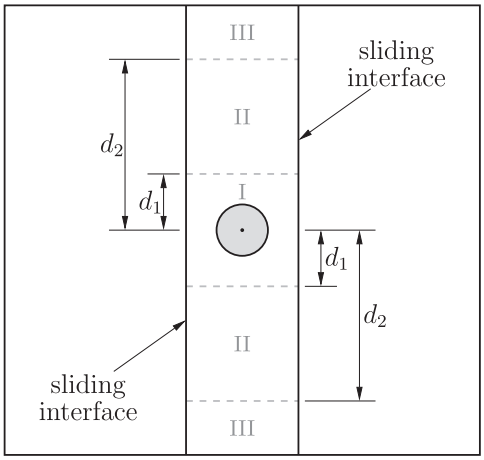

FIGURE 4 Schematic of a computational domain with two sliding interfaces for a vibrating cylinder

where d_1 and d_2 are the two vertical distances (measured from the cylinder center) used to define the virtual regions. The following blending function[23] is then defined to control the mesh motion:

$$b(r) = \begin{cases} 1, & \text{if } r \le 0 \quad \text{(ie, in region I)} \\ 1 - 10r^3 + 15r^4 - 6r^5, & \text{if } 0 < r < 1 \quad \text{(ie, in region II)} \\ 0, & \text{if } r \ge 1 \quad \text{(ie, in region III).} \end{cases} \tag{41}$$

When the cylinder has a vertical displacement of Δy_c (measured from its initial position), the coordinates of a grid point in the middle subdomain are updated to

$$x = x_0, \tag{42}$$

$$y = y_0 + b(r) \cdot \Delta y_c. \tag{43}$$

The grid velocities are updated in a similar way using the same blending function.

We illustrate the advantages of this sliding-mesh approach in Figure 5 for two vibrating cylinders. Figure 5A shows the initial mesh, where the two cylinders are separated by a distance of two times the diameter. Figure 5B shows the conforming mesh when the two cylinders have moved one diameter apart in the vertical direction. It is obvious that the mesh has become very skewed above the leading cylinder, below the trailing cylinder, and in-between the two cylinders. Figure 5C shows the sliding mesh, and it is evident that the mesh quality is much improved by introducing sliding interfaces.

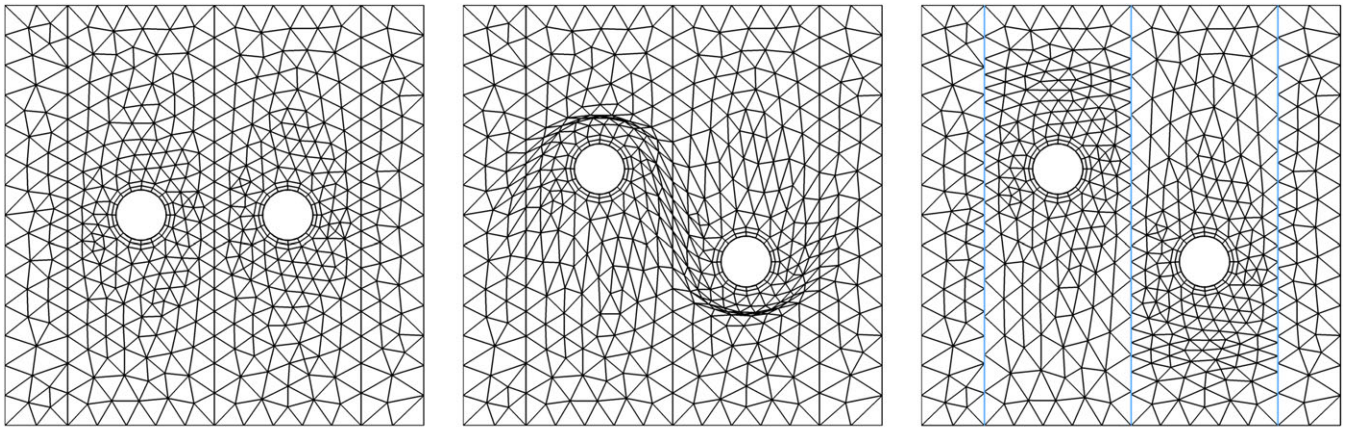

FIGURE 5 Schematic of meshes for two vibrating cylinders. A, Initial mesh; B, Conforming deforming mesh; C, Sliding mesh (blue lines are sliding interfaces)

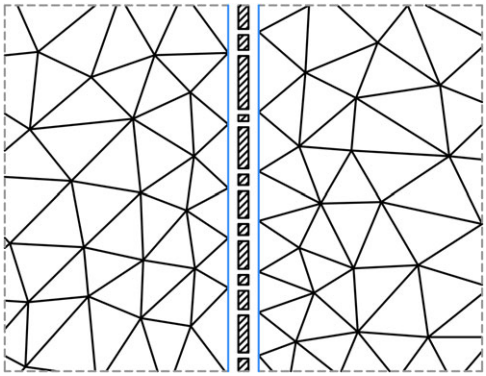

FIGURE 6 Schematic of the distribution of mortar elements (hatched) between two sliding meshes [Colour figure can be viewed at wileyonlinelibrary.com]

3.4 | A nonuniform sliding-mesh method

Meshes on the two sides of a sliding interface are nonconforming and nonuniform in this work, and this creates geometric incompatibility. To overcome this issue, we extend a previous uniform sliding-mesh method[19,20] to the general nonuniform case. This method employs dynamic mortar elements[14,24] to exchange information between the two sides of an interface.

Figure 6 shows a schematic of the distribution of mortar elements between two sliding meshes. As we can see, a mortar is formed between two successive mesh points along the sliding interface, and these two points could come from either one side or two sides of an interface. A mortar element is always connected to two cell faces: one on its left and one on its right. In contrast, a cell face may have one or multiple mortar elements, and this number may change with time. These mortar and face connectivities need to be updated at every sub–time step of a time-marching scheme.

To make connectivity updating more efficient, we reorder cell faces on each side of a sliding interface to a bottom-to-top order and put the reordered ones into a list (faces on the left side are stored first, and then those from the right). This reordering is done during preprocessing and only needs to be done once. We denote the left side of an interface as "*l*" and the right side as "*r*." For a sliding interface, assume we have *nfl* faces on the left and *nfr* on the right, then the total number of faces is $nf = nfl + nfr$. It is evident that the total number of mortars is $nm = nf - 1$, and this number does not change with time. Knowing this number allows us to allocate memories during preprocessing. Next, we define four arrays to store the connectivities: `vof`(1:*nf*, 1:2), `mof`(1:*nf*, 1:2), `fom`(1:*nm*, 1:2), and `vom`(1:*nm*, 1:2). To be more specific, `vof`(*i*, 1:2) stores the two vertices of the *i*th face; `mof`(*i*, 1) and `mof`(*i*, 2) store the first mortar and the total number of mortars, respectively, of the *i*th face; `fom`(*i*, 1) and `fom`(*i*, 2) are the left and right faces, respectively, of the *i*th mortar; and `vom`(*i*, 1:2) stores the two vertices of the *i*th mortar. Algorithm 1 lists the detailed steps for updating these connectivities.

We already know that a cell face is mapped to a unit line segment, for example, $0 \le \xi \le 1$, when the cell is mapped to a standard computational element. Similarly, we also map each mortar element to a unit line segment $0 \le z \le 1$, which we call the mortar space. The mortar space and the computational space are then related as

$$\xi = o(t) + s(t)z, \tag{44}$$

where $o(t)$ is the offset of the mortar with respect to the start point of the cell face, and $s(t)$ is the relative scaling. If we use Ω to denote a cell face and Ξ to denote a mortar, then, as shown in Figure 7B, we have $o_1 = 0$ and $s_1 = L^{\Xi_1}/L^{\Omega}$ for Ξ_1, $o_i = (L^{\Xi_1} + \cdots + L^{\Xi_{i-1}})/L^{\Omega}$ and $s_i = L^{\Xi_i}/L^{\Omega}$ for Ξ_i, where L denotes the physical length of a cell face or a mortar.

According to the 2D solution polynomial, the solution on a cell face Ω is represented by the following one-dimensional polynomial:

$$\mathbf{Q}_f^{\Omega} = \sum_{i=1}^{N} \left(\mathbf{Q}_f^{\Omega}\right)_i h_i(\xi). \tag{45}$$

If we define the same set of SPs on a mortar Ξ, then the solution polynomial on Ξ has a similar form, ie,

$$\mathbf{Q}_f^{\Xi} = \sum_{i=1}^{N} \left(\mathbf{Q}_f^{\Xi}\right)_i h_i(z), \tag{46}$$

where $(\mathbf{Q}_f^{\Xi})_i$ is the discrete solution at the *i*th SP on mortar Ξ.

Algorithm 1 Algorithm for updating mortar and face connectivities

```
mof=0; fom=0; vom=0                                   # initialize all arrays with zeros

ifl = 1
ifr = nfl + 1
im = 1                                                # partial connectivities of the first mortar
mof(ifl, 1) ← im
mof(ifr, 1) ← im
vom(im, 1) ← vof(ifl, 1)

for im = 1 to nm − 1 do                               # update connectivities of mortars 1 to nm
    if vof(ifl, 2) lies between vof(ifr, 1) and vof(ifr, 2) then
        mof(ifl + 1, 1) ← im + 1
        mof(ifl, 2) ← mof(ifl, 2) + 1
        mof(ifr, 2) ← mof(ifr, 2) + 1
        fom(im, 1) ← ifl
        fom(im, 2) ← ifr
        vom(im, 2) ← vof(ifl, 2)
        vom(im + 1, 1) ← vof(ifl, 2)
        ifl ← ifl + 1
    else
        mof(ifr + 1, 1) ← im + 1
        mof(ifl, 2) ← mof(ifl, 2) + 1
        mof(ifr, 2) ← mof(ifr, 2) + 1
        fom(im, 1) ← ifl
        fom(im, 2) ← ifr
        vom(im, 2) ← vof(ifr, 2)
        vom(im + 1, 1) ← vof(ifr, 2)
        ifr ← ifr + 1
    end if
end for

mof((nfl), 2) ← mof(nfl, 2) + 1                       # finish connectivities of mortar nm
mof((nf), 2) ← mof(nf, 2) + 1
fom(nm, 1) ← nfl
fom(nm, 2) ← nf
vom(nm, 2) ← vof(nf, 2)
```

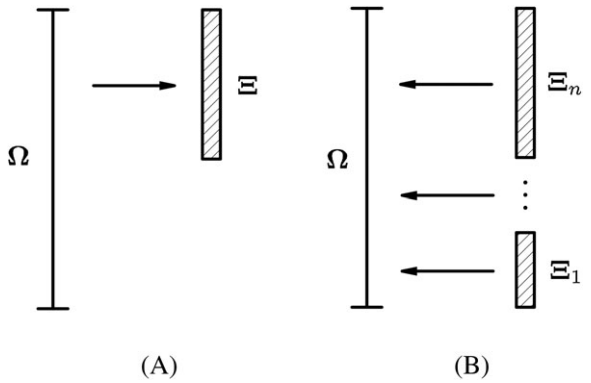

FIGURE 7 Projection between a cell face and mortars. A, From a cell face to the left side of a mortar; B, From *n* mortars back to the left cell face

To get the discrete solutions on a mortar, for example, those on the left side of mortar Ξ in Figure 7A, we require

$$\int_0^1 \left(\mathbf{Q}_f^{\Xi,L}(z) - \mathbf{Q}_f^{\Omega}(\xi)\right) h_j(z)\mathrm{d}z = 0, \quad \forall j = 1, 2, \ldots, N. \tag{47}$$

The solutions of this equation system when written in matrix form are

$$\left(\mathbf{Q}_f^{\Xi,L}\right)_{1:N} = \mathbf{M}^{-1}\mathbf{S}^{\Omega\to\Xi}\left(\mathbf{Q}_f^{\Omega}\right)_{1:N}, \tag{48}$$

where the matrices $\mathbf{M}$ and $\mathbf{S}^{\Omega\to\Xi}$ have the following elements:

$$M_{i,j} = \int_0^1 h_i(z)h_j(z)\mathrm{d}z, \quad S_{i,j}^{\Omega\to\Xi} = \int_0^1 h_i(o+sz)h_j(z)\mathrm{d}z, \quad i, j = 1, 2, \ldots, N, \tag{49}$$

where o and s are the offset and the scaling, respectively, of mortar Ξ, with respect to face Ω.

Similarly, we can get solutions on the right side of a mortar, ie, $(\mathbf{Q}_f^{\Xi,R})_{1:N}$. The common solution is then computed as

$$\left(\mathbf{Q}_f^{\Xi}\right)_{1:N} = \frac{1}{2}\left(\left(\mathbf{Q}_f^{\Xi,L}\right)_{1:N} + \left(\mathbf{Q}_f^{\Xi,R}\right)_{1:N}\right). \tag{50}$$

The common inviscid fluxes at the SPs of the mortar, denoted by $(\widetilde{\mathbf{F}}_{\mathrm{inv}}^{\Xi})_{1:N}$, are computed using a Riemann solver, for example, the Rusanov solver.[22]

The flux polynomials on a cell face and a mortar element have the same forms as those for the solution polynomials in Equations (45) and (46). As shown in Figure 7B, to project the common inviscid fluxes from n mortars back to a cell face Ω, we require

$$\sum_{i=1}^{n}\int_{o_i}^{o_i+s_i} \left(\widetilde{\mathbf{F}}_{\mathrm{inv}}^{\Omega}(\xi) - \widetilde{\mathbf{F}}_{\mathrm{inv}}^{\Xi_i}(z)\right) h_j(\xi)d\xi = 0, \ \forall j = 1, 2, \ldots, N, \tag{51}$$

where $\widetilde{\mathbf{F}}_{\mathrm{inv}}^{\Omega}(\xi)$ and $\widetilde{\mathbf{F}}_{\mathrm{inv}}^{\Xi_i}(z)$ are the inviscid flux polynomials on face Ω and mortar Ξ_i, respectively. The solutions of the above equation system when written in matrix form are

$$\left(\widetilde{\mathbf{F}}_{\mathrm{inv}}^{\Omega}\right)_{1:N} = \sum_{i=1}^{n} s_i\mathbf{M}^{-1}\mathbf{S}^{\Xi_i\to\Omega}\left(\widetilde{\mathbf{F}}_{\mathrm{inv}}^{\Xi_i}\right)_{1:N}, \tag{52}$$

where matrix $\mathbf{M}$ is identical to that in Equation (49), and $\mathbf{S}^{\Xi_i\to\Omega}$ is simply the transpose of $\mathbf{S}^{\Omega\to\Xi_i}$.

For viscous flow, the common solutions $(\mathbf{Q}_f^{\Xi})_{1:N}$ are projected back to cell faces to compute local viscous fluxes $(\widetilde{\mathbf{F}}_{\mathrm{vis}}^{\Omega})_{1:N}$, which are then projected to mortars following Equation (47). The common viscous fluxes, ie, $(\widetilde{\mathbf{F}}_{\mathrm{vis}}^{\Xi})_{1:N}$, are computed as the average of the left and right values and are finally projected back to cell faces following Equation (51).

3.5 | Temporal scheme and GCL discretization

The coupled–fluid-and-solid system (ie, Equation (22)) is a first-order partial differential equation system in time. We march this system in time using an explicit five-stage fourth-order strong stability–preserving Runge-Kutta scheme.[25]

To numerically satisfy the GCL equations (ie, Equations (11)-(13)), we apply the same spatial and temporal schemes as for the governing equations to them. Because our spatial discretization scheme is actually a direct differentiation and mesh movement is controlled by blending polynomials, the first two GCL equations are therefore satisfied automatically. For the last GCL equation, we treat $|\mathcal{J}|$ as an unknown and solve for it. This numerical $|\mathcal{J}|$ is then used to update the physical flow solutions. In this way, all GCL equations are numerically satisfied.

3.6 | Parallelization

Each subdomain mesh is generated separately. During preprocessing, we combine these meshes into a single file and number the elements consecutively. The Metis library[26] is then called to partition this single mesh file. After that, each processor reads in its own part of the mesh. For example, Figure 8 shows the partitions of a simple mesh for six processors. We notice that the grid partition for processor P0 (also for P5) is discontinuous; this is because of the discontinuous sliding

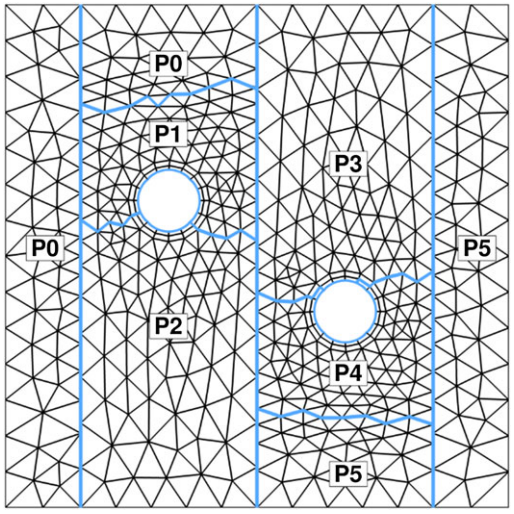

FIGURE 8 Schematic of partitions of a simple mesh for six processors (blue lines represent processor interfaces)

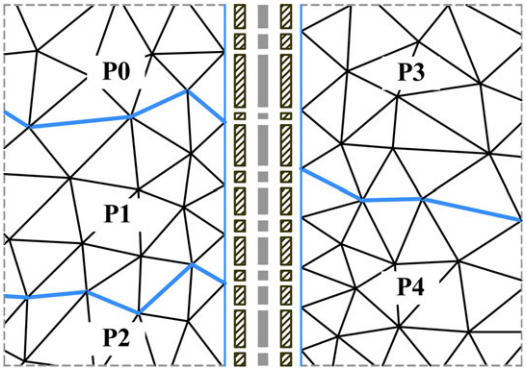

FIGURE 9 Schematic of the distribution of mortars between two subdomains for the parallel solver (hatched lines are local mortars; thick gray lines are global mortars)

interfaces. However, since the sliding mesh introduces only a very small amount of computational cost,[19] the overall load balancing is not a problem, especially for real simulations where the computational cost on the sliding-mesh interface is negligible.

After mesh partitioning, each processor only contains local information that is incomplete about the overall mesh. To exchange information between sliding interfaces, we introduce the concept of local mortar and global mortar. For example, in Figure 9, a local mortar (hatched) on processor P1 is only connected to a cell face on its left, and only the left information is locally available, whereas the right information needs to be sent from its pairing mortar on processor P3 or P4. Similarly, a mortar on processor P3 is only connected to a cell face on its right, and only the right information is locally available, whereas the left information needs to be sent from its pairing mortar on processor P0 or P1. A local mortar finds its pairing mortar through a local-mortar-to-global-mortar connectivity: two local mortars that share the same global mortar form a pair. Global mortars (denoted by thick gray lines) are equivalent to the mortars in a serial solver (see Figure 6) and are generated in the same way as in a serial solver. The exchange of information is done by using the message passing interface library.

4 | NUMERICAL TESTS

In this section, we first verify the spatial accuracies of the solver on an inviscid flow and a viscous flow. Following that, we simulate VIVs of a single cylinder and compare the results with previously published ones. Finally, we apply the solver to simulate VIVs of two tandem cylinders and test its parallel efficiency by simulating VIVs of a row of three cylinders.

4.1 | Euler vortex flow

In a Euler vortex flow,[27] an isentropic vortex is superimposed to and convected by a uniform mean flow. The analytical solution for the Euler vortex flow in an infinite domain is

$$u = U_\infty \left\{ \cos\theta - \frac{\epsilon y_r}{r_c} \exp\left(\frac{1 - x_r^2 - y_r^2}{2r_c^2} \right) \right\}, \tag{53}$$

$$v = U_\infty \left\{ \sin\theta + \frac{\epsilon x_r}{r_c} \exp\left(\frac{1 - x_r^2 - y_r^2}{2r_c^2} \right) \right\}, \tag{54}$$

$$\rho = \rho_\infty \left\{ 1 - \frac{(\gamma - 1)(\epsilon M_\infty)^2}{2} \exp\left(\frac{1 - x_r^2 - y_r^2}{r_c^2} \right) \right\}^{\frac{1}{\gamma - 1}}, \tag{55}$$

$$p = p_\infty \left\{ 1 - \frac{(\gamma - 1)(\epsilon M_\infty)^2}{2} \exp\left(\frac{1 - x_r^2 - y_r^2}{r_c^2} \right) \right\}^{\frac{\gamma}{\gamma - 1}}, \tag{56}$$

where U_∞, ρ_∞, p_∞, and M_∞ are the mean flow speed, density, pressure, and Mach number, respectively; θ is the direction of the mean flow; ϵ and r_c denote the vortex strength and size, respectively; and the relative coordinates (x_r, y_r) are defined as

$$x_r = x - x_0 - \bar{u}t, \tag{57}$$

$$y_r = y - y_0 - \bar{v}t, \tag{58}$$

where $(\bar{u}, \bar{v}) = (U_\infty \cos\theta, U_\infty \sin\theta)$ are the mean velocity components, and (x_0, y_0) represent the initial position of the vortex.

In the present simulation setup, the mean flow is $(U_\infty, \rho_\infty) = (1, 1)$, with a Mach number $M_\infty = 0.3$ and a direction $\theta = \pi/6$. The size of the computational domain is $0 \le x, y \le 10$. A vortex with $\epsilon = 1$, $r_c = 1$ is initially placed at the center of the domain, ie, at $(x_0, y_0) = (5, 5)$. The computational domain is divided into three subdomains by $x = 3$ and $x = 7$. The two side subdomains are meshes into triangular cells, whereas the middle subdomain is meshed into quadrilateral cells. Four meshes with 44/15, 156/50, 568/200, and 2216/800 hybrid triangular/quadrilateral cells are employed for accuracy tests. Figure 10 shows a mesh with 156/50 triangular/quadrilateral cells. The horizontal centerline of the middle subdomain (ie, $y_c = 5, 3 \le x \le 7$) has a motion $\Delta y_c = \sin(0.5t)$, and the deforming region is $|y_0 - 5| \le 4$. The two side subdomains are kept stationary.

In Figure 11, we compare the density contours from the exact solution and the fourth-order scheme on the mesh from Figure 10. The time is $t = 2.3$ when the center of the vortex travels onto one of the sliding interfaces. As we can see, the solver resolves the vortex very well, and the nonconforming and nonuniform sliding interfaces do not contaminate the flow at all.

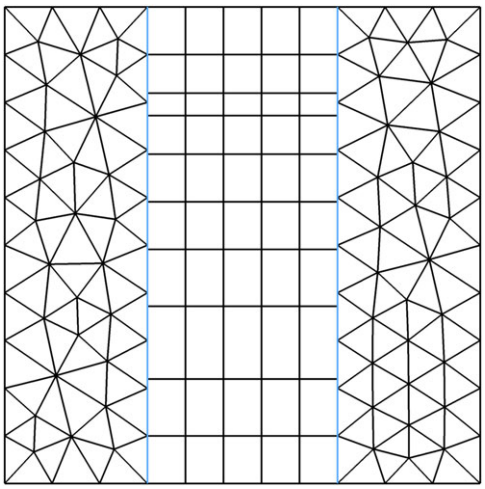

FIGURE 10 Mesh with 156/50 triangular/quadrilateral cells for Euler vortex flow simulation (blue lines are sliding interfaces)

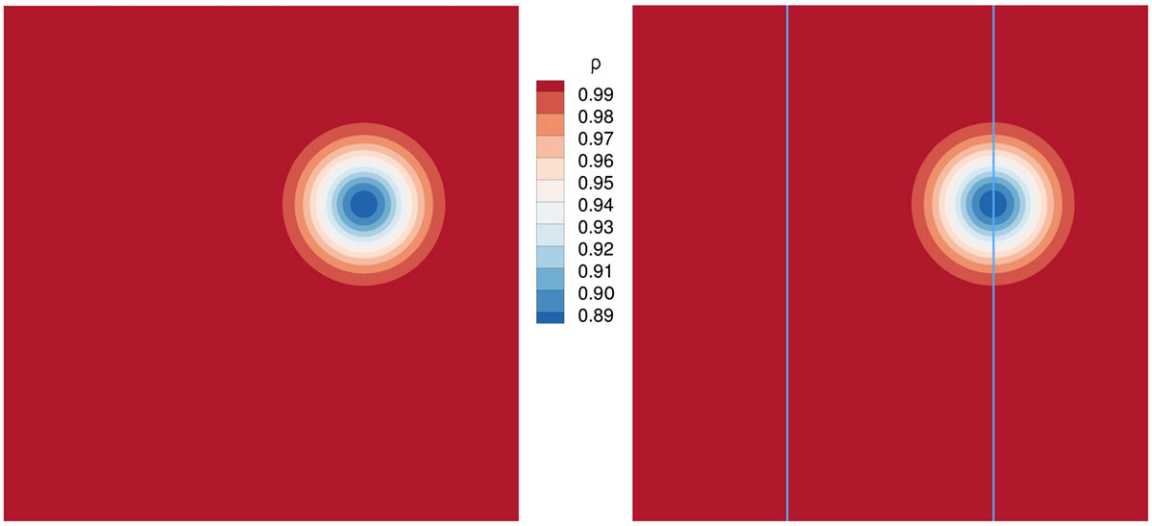

FIGURE 11 Density contours for the Euler vortex flow. (Left) Exact solution; (Right) Numerical solution (blue lines are sliding interfaces)

TABLE 1 Errors and orders of accuracy (based on density) for the Euler vortex flow simulation

Scheme	Cells	L_1 Error	Order	L_2 Error	Order
Third-order	44/15	1.58E−03	–	2.97E−03	–
	156/50	3.08E−04	2.62	5.69E−04	2.65
	568/200	4.32E−05	2.97	9.17E−05	2.76
	2216/800	6.58E−06	2.75	1.47E−05	2.67
Fourth-order	44/15	6.95E−04	–	1.37E−03	–
	156/50	5.76E−05	4.00	1.14E−04	4.00
	568/200	3.44E−06	4.25	7.43E−06	4.12
	2216/800	2.00E−07	4.15	4.34E−07	4.14

Furthermore, we compute the orders of accuracy from the L_1 and L_2 errors of density for this flow. The errors are defined as

$$L_1 \text{ error} = \frac{\sum_{i=1}^{\text{DOF}} \left|\rho_i - \rho_i^{\text{exact}}\right|}{\text{DOF}}, \; L_2 \text{ error} = \sqrt{\frac{\sum_{i=1}^{\text{DOF}} \left(\rho_i - \rho_i^{\text{exact}}\right)^2}{\text{DOF}}}, \tag{59}$$

where ρ_i and ρ_i^{exact} are the numerical and exact solutions, respectively, at the ith DOF, and DOF is the total number of DOFs. For an Nth-order scheme on a mesh with N_{quad} quadrilateral cells and N_{tri} triangular cells, we have

$$\text{DOF} = N_{\text{quad}} \cdot N^2 + N_{\text{tri}} \cdot \frac{N(N+1)}{2}. \tag{60}$$

The computed errors and accuracies are shown in Table 1 for the third- and fourth-order schemes on four meshes at $t = 2.3$. It is obvious that despite of the presence of nonconforming sliding interfaces, the sliding-mesh method retains the high-order accuracy of the SD and SDRT methods on this inviscid flow.

4.2 | Planar Couette flow

For the viscous planar Couette flow between two infinite plates that are separated by a distance of H, if the top plate moves at speed U and has a temperature T_t and the bottom plate is stationary with a temperature T_b, then the steady-state flow field is

$$u = \frac{y}{H}U, \; v = 0, \tag{61}$$

$$e = e_b + \frac{y}{H}(e_t - e_b) + \frac{P_r U^2}{2\gamma}\left(\frac{y}{H} - \left(\frac{y}{H}\right)^2\right), \tag{62}$$

$$p = \text{constant}, \tag{63}$$

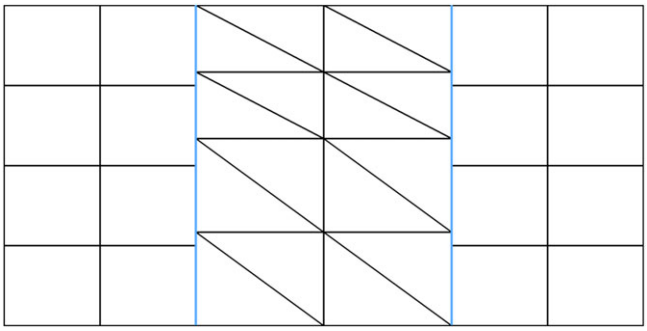

FIGURE 12 Mesh with 16/16 triangular/quadrilateral cells for planar Couette flow simulation (blue lines are sliding interfaces)

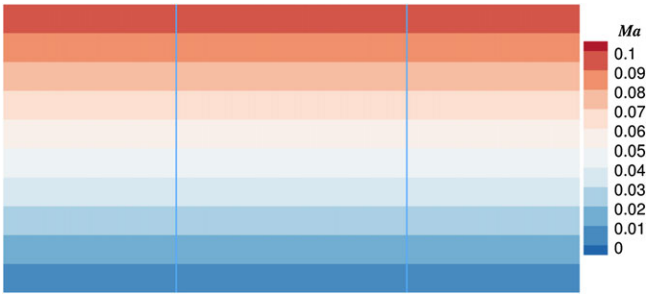

FIGURE 13 Mach number contours of the planar Couette flow using the fourth-order scheme (blue lines are sliding interfaces)

where $e = C_V T$ is the internal energy; $C_V = R/(\gamma - 1)$ is the specific heat at constant volume; the subscripts "b" and "t" denote the bottom and top plates, respectively; and Pr is the Prandtl number, which is set to 0.72 in this work.

In the present simulation, we have set $H = 1, U = 1.0$, and $T_b = T_t = 1.0$. The Mach number on the top plate is $Ma = 0.1$. The Reynolds number based on H, U, and the fluid viscosity (which is assumed constant) is $Re = 100$. The overall computational domain is bounded by $0 \le x \le 2$ and $0 \le y \le H$. It is further divided into three subdomains by $x = 0.6$ and $x = 1.4$. The middle subdomain again has a motion $\Delta y_c = 0.1 \sin(0.5t)$ about its centerline, and meshes within $0.1 \le y_0 \le 0.9$ are deformed. The other two subdomains are fixed. The top and bottom boundaries are treated as no-slip isothermal walls. The left and right boundaries are set as periodic boundaries. Three meshes with 4/4, 16/16, and 64/64 hybrid triangular/quadrilateral cells are used for the tests. Figure 12 shows the mesh with 16/16 triangular/quadrilateral cells.

Figure 13 shows the steady-state Mach number contours from the fourth-order scheme on the mesh with 16/16 triangular/quadrilateral cells. It is evident that the grid motion and the nonconforming sliding interfaces do not introduce any visible disturbances to the flow field. The Mach contours align very well along the y-direction, and they are seen invariant along the x-direction, which is consistent with the analytical solutions.

The errors and orders of accuracy for this flow are computed in the same way as for the Euler vortex flow. Table 2 shows the results for the u velocity. It is observed that for this viscous flow, the sliding-mesh method and the hybrid SD method have also achieved the optimum orders of accuracy.

TABLE 2 Errors and orders of accuracy (based on the u velocity) for the planar Couette flow simulation

Scheme	Cells	L_1 Error	Order	L_2 Error	Order
Third-order	4/4	2.03E−05	–	2.36E−05	–
	16/16	2.54E−06	3.00	2.89E−06	3.03
	64/64	3.05E−07	3.06	3.47E−07	3.06
Fourth-order	4/4	2.46E−07	–	3.91E−07	–
	16/16	1.77E−08	3.79	2.71E−08	3.85
	64/64	1.18E−09	3.90	1.66E−09	4.03

4.3 | VIVs of an isolated cylinder

The motion of an elastically mounted cylinder placed in a uniform free-stream flow is affected by several nondimensional variables: the mass ratio (m^*), the speed ratio (U^*), the damping ratio (ζ), the lift coefficient (C_L), and the Reynolds number (Re), and these parameters are defined as

$$m^* = \frac{m}{\rho_\infty \pi \left(\frac{1}{2}d\right)^2 l}, \tag{64}$$

$$U^* = \frac{u_\infty}{f_N d}, \tag{65}$$

$$\zeta = \frac{c}{2\sqrt{km}}, \tag{66}$$

$$C_L = \frac{F_L}{\frac{1}{2}\rho_\infty u_\infty^2 l d}, \tag{67}$$

$$Re = \frac{\rho_\infty u_\infty d}{\mu_\infty}, \tag{68}$$

where the physical meanings of m, k, c, and F_L have already been explained in Equation (14); ρ_∞, u_∞, and μ_∞ are the free-stream density, velocity, and viscosity, respectively; d is the diameter of the cylinder; l is the spanwise length of the cylinder, which is treated as 1 for 2D simulations; and f_N is the natural frequency for undamped vibration, whose expression is

$$f_N = \frac{1}{2\pi}\sqrt{\frac{k}{m}}. \tag{69}$$

Following the simulation in the work of Prasanth et al,[8] in this test, we set $m^* = 10$, $\zeta = 0$, and $U^* = 0.06Re$, where Re takes several different values. The free-stream flow has a Mach number of 0.1. The overall domain has a size of $60d \times 100d$ or $60d \times 20d$, resulting in a blockage ratio of 1% or 5%. The middle subdomain has a width of $6d$ and is bounded by two sliding interfaces. The boundary conditions are set as follows: Dirichlet for the inlet, pressure outflow for the outlet, symmetric for the top and the bottom, and no-slip adiabatic wall for the cylinder.

Figure 14 shows a global view and a local view of the computational mesh with a blockage ratio of 5%, where the mesh has a total number of 6696 cells, with 1494 in the middle subdomain and 5202 in the two static subdomains altogether. The mesh with a blockage ratio of 1% has a total number of 8818 cells, with 1854 in the middle subdomain and 6964 in the two static subdomains altogether. The mesh around the cylinder and that in the wake region are refined. A quadrilateral boundary-layer mesh has been applied on the cylinder surface, where the minimum mesh spacing normal to the surface is $0.01d$. The mesh is designed such that the solutions are mesh independent for both the third- and fourth-order schemes. A computational time step size of $\Delta t u_\infty/d = 1.0 \times 10^{-3}$ is used for all simulations.

The maximum vibration amplitude ($y_{\max}/d$) and the maximum lift coefficients ($(C_L)_{\max}$), both from the fourth-order scheme, are plotted in Figures 15 and 16 against the Reynolds number. From both figures, it is seen that both the vibration amplitude and the lift coefficient first increase and then decrease with the Reynolds number. The maximum values of $y_{\max}/d$ and $(C_L)_{\max}$ are achieved when the vortex shedding frequency approaches the natural frequency (Equation (69)),

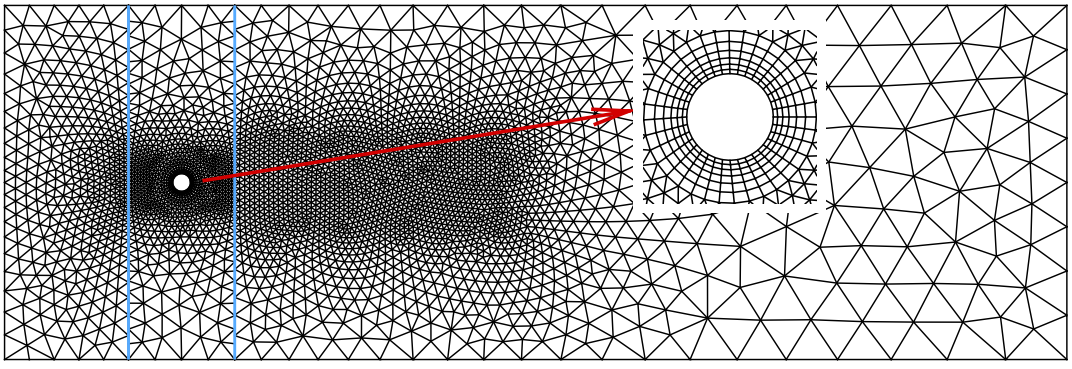

FIGURE 14 Global and local views of the mesh with a blockage ratio of 5% for simulating vortex-induced vibrations of a single cylinder [Colour figure can be viewed at wileyonlinelibrary.com]

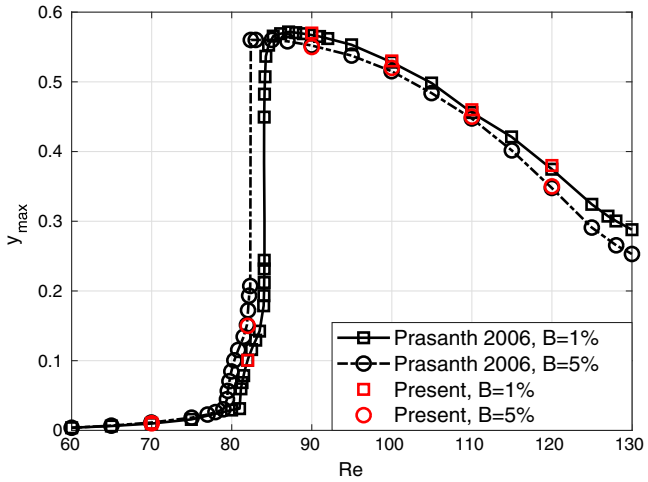

FIGURE 15 The maximum normalized vibration amplitude for vortex-induced vibrations of a single cylinder [Colour figure can be viewed at wileyonlinelibrary.com]

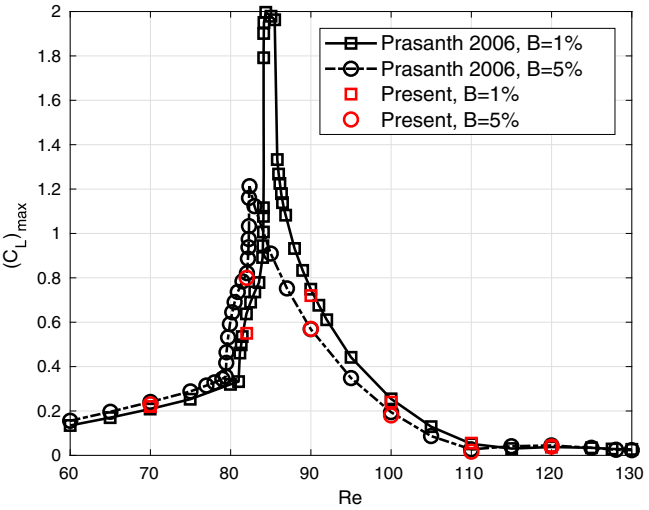

FIGURE 16 The maximum lift coefficient for vortex-induced vibrations of a single cylinder [Colour figure can be viewed at wileyonlinelibrary.com]

which are reflected as a big jump on each curve. It is seen that the present results agree well with the previous published ones, which verifies the solver.

4.4 | VIVs of two tandem cylinders

In this test, we apply the solver to two cylinders in a tandem arrangement. The two cylinders are separated by a gap of $\Delta L/d = 0.1$. The aim of this test is to demonstrate the capability of the solver for dealing with very closely placed objects with relative motions. The incoming free-stream flow has a Mach number of $Ma = 0.1$ and a Reynolds number of $Re = 200$ (based on free-stream properties and cylinder diameter). The two cylinders are identical, with a mass ratio of $m^* = 5.0$, a speed ratio of $U^* = 9$, and a damping ratio of $\zeta = 0$.

Figure 17 shows a global view and a local view of the mesh for this case. The overall domain has a size of $66d \times 40d$. The front cylinder locates $10d$ away from the inlet. The domain is divided into four subdomains by three sliding interfaces (each locates $0.05d$ away from the cylinder). The overall domain is discretized into 22 220 triangular cells, with 3200 cells in each moving subdomain. The mesh has been refined around the cylinders and in the wake region to provide good

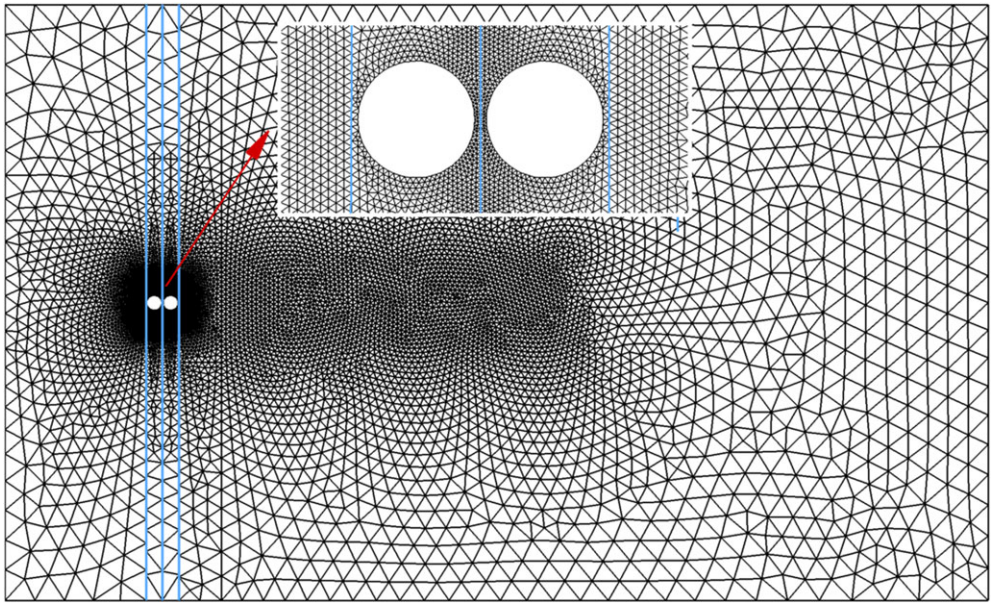

FIGURE 17 Global and local views of the mesh for vortex-induced vibrations of two tandem cylinders [Colour figure can be viewed at wileyonlinelibrary.com]

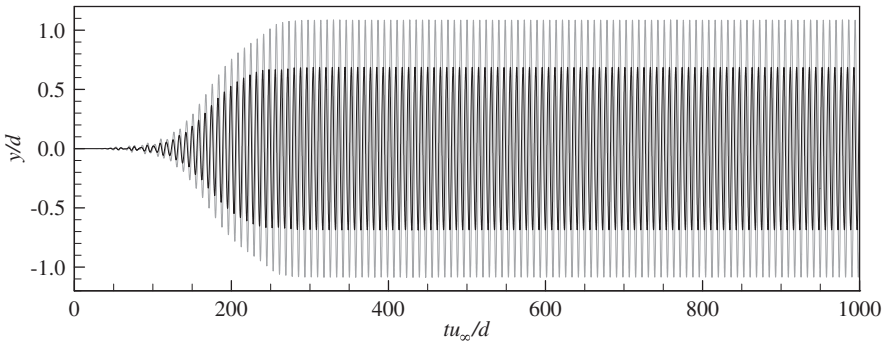

FIGURE 18 Overview of the vortex-induced vibration displacement histories of two tandem cylinders. (Black) Front cylinder; (Gray) Rear cylinder

resolution. The boundary conditions are set the same as those for the previous isolated cylinder case. The fourth-order scheme with a time step size of $\Delta tu_\infty/d = 2.0 \times 10^{-4}$ is used for this simulation.

The simulation was started from a uniform flow field and continued for 1000 time units. Figure 18 shows the displacement history of the cylinders for the complete simulation. We see that the displacement magnitudes for both cylinders gradually increase and finally converge at about $tu_\infty/d = 300$. For the converged flow, the front cylinder (black curve) has a vibration magnitude of about $y_{\max}/d = 0.69$, and the rear cylinder (gray curve) has a magnitude of $y_{\max}/d = 1.09$, which is obviously larger than that of the front cylinder.

To see the details of the curves, we have plotted a close view from $tu_\infty/d = 940$ to 1000 in Figure 19. The vibrations for both cylinders are seen to have the same period, which is approximately $Tu_\infty/d = 8.10$. The front cylinder leads the rear one for about $\Delta tu_\infty/d = 1.5$, which corresponds to a phase angle of about 66.7°.

The flow field is visualized using vorticity contours in Figure 20 for approximately one period. We see that as the cylinders move downward, a negative vortex is shed off from the top; as they move upward, a positive vortex is shed off from the bottom. As a result, a vortex street consisting of an upper row of negative vortices and a lower row of positive vortices is formed in the wake region. Overall, the rear cylinder behaves like a tail of the front one, where the lag on motion is reflected as the phase difference that we have observed in the VIV magnitude curves. We also see interactions between the two cylinders. For example, in Figure 20A, a negative vortex is formed on top of the front cylinder, and then, in Figure 20B, this vortex is squeezed and pushed through the gap between the cylinders. For this reason, vortex shedding from the front cylinder is greatly suppressed, resulting in a smaller vibration magnitude than the rear cylinder.

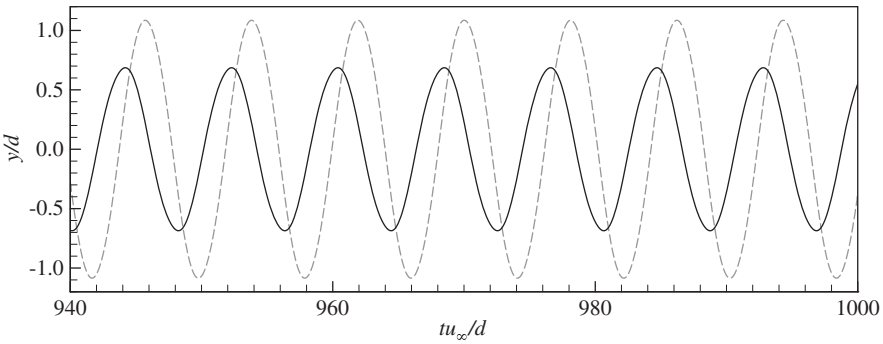

FIGURE 19 Close view of the vortex-induced vibration displacement histories of two tandem cylinders. (Solid) Front cylinder; (Dashed) Rear cylinder

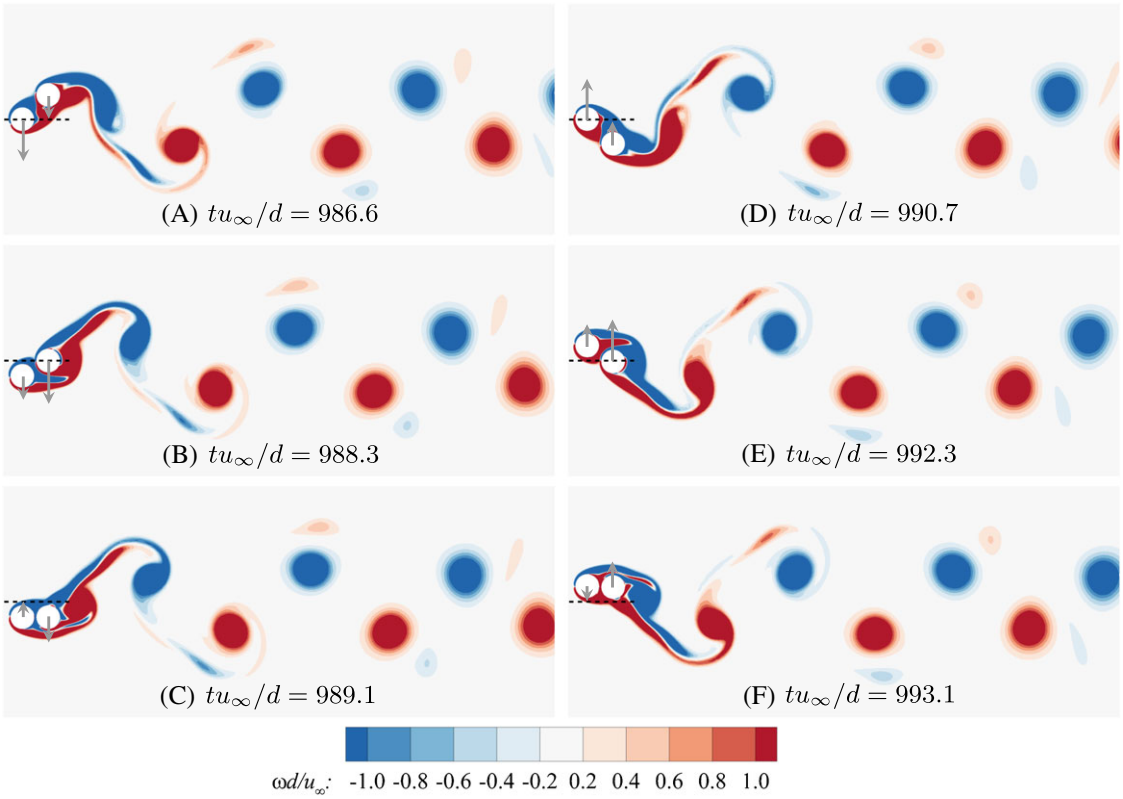

FIGURE 20 Vorticity contours in a period for vortex-induced vibrations of two tandem cylinders (dashed line represents mean position) [Colour figure can be viewed at wileyonlinelibrary.com]

4.5 | VIVs of three cylinders in a row

In this test, we simulate the VIV of a row of three cylinders. The aim of this test is to demonstrate the solver's capability in dealing with multiple objects with large relative displacements and to test the speedup of parallelization. Each cylinder is separated by a gap of $\Delta L/d = 1.0$ from its neighbor(s). Other parameters for the simulation setup are similar to those of the previous case. For example, the incoming free-stream flow has a Mach number of $Ma = 0.1$ and a Reynolds number of $Re = 200$, and each cylinder has a mass ratio of $m^* = 5.0$, a speed ratio of $U^* = 9$, and a damping ratio of $\zeta = 0$.

The mesh for this case is shown in Figure 21, and the partitions for 128 processors are shown in Figure 22. The overall domain has a size of $74d \times 50d$. The front cylinder locates $20d$ away from the inlet. The domain is divided into five subdomains by four sliding interfaces. These subdomains are discretized into a total number of 51 320 hybrid grids, with 7116 cells in each moving subdomain, 8708 in the front subdomain, and 21 264 in the last subdomain. Once again, the

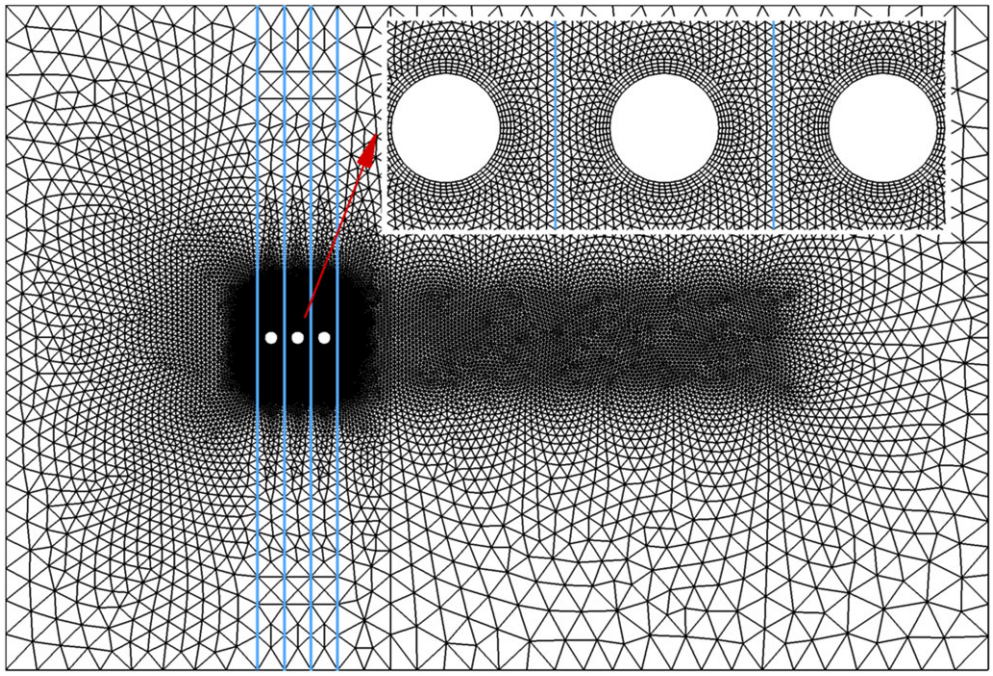

FIGURE 21 Global and local views of the mesh for vortex-induced vibrations of three cylinders [Colour figure can be viewed at wileyonlinelibrary.com]

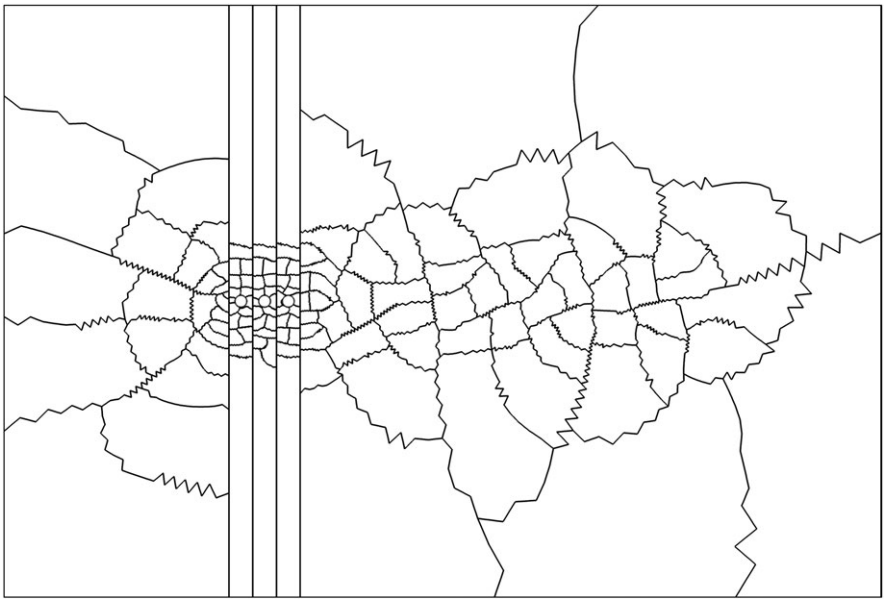

FIGURE 22 Partitions of a mesh for vortex-induced vibrations of three cylinders on 128 processors

mesh has been refined around the cylinders and in the wake region to provide good resolution. The boundary conditions are set the same as those for the previous isolated cylinder case. The fourth-order scheme with a time step size of $\Delta t u_\infty / d = 2.0 \times 10^{-4}$ is used for the simulation.

Figure 23 shows the time history of the displacements for 1000 time units. Compared to the previous tandem-cylinder case, the flow in the present case shows a more complex nature as indicated by the displacement curves: the magnitudes first increase, followed by an obvious decrease, and then again increase until reaching the converged states. Convergence happens at around $t u_\infty / d = 250$, where the vibration magnitudes are found to be $y_{\max}/d = 0.30$, 0.90, and 1.07 for

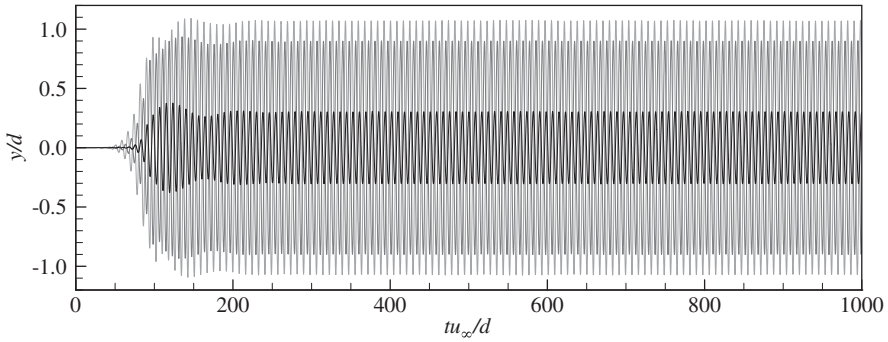

FIGURE 23 Overview of the vortex-induced vibration displacement history of three cylinders in a row. (Black) Front cylinder; (Dark gray) Middle cylinder; (Light gray) Rear cylinder

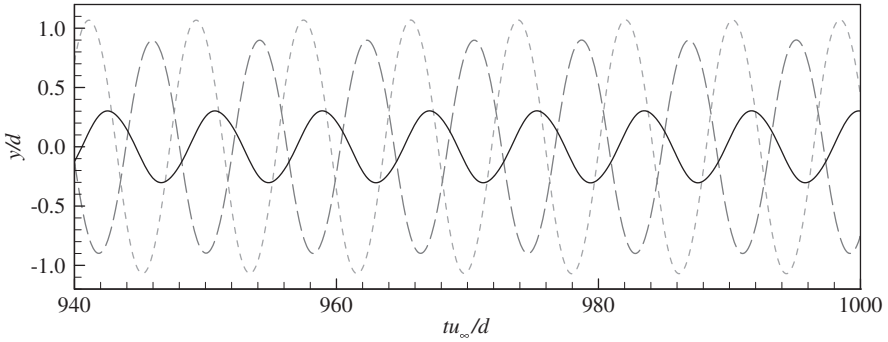

FIGURE 24 Close view of the vortex-induced vibration displacement history of three cylinders in a row. (Solid) Front cylinder; (Long dashed) Middle cylinder; (Short dashed) Rear cylinder

the front, middle, and rear cylinders, respectively. It is interesting to notice that the magnitudes increase from front to rear, which is probably because vortices from the preceding cylinder(s) have enhanced the vibration of the downstream cylinder(s).

A close view of the displacements is plotted in Figure 24 from $tu_\infty/d = 940$ to $tu_\infty/d = 1000$. It is seen that all curves show a period of approximately $Tu_\infty/d = 8.20$, which is very close to that of the two tandem cylinders in the previous test. This is possibly because that the converged flows have approached the vicinity of the lock-on region around the natural frequency (which is 1/9 and corresponds to a period of 9). We also see obvious phase differences on the curves. The front cylinder leads the middle one for about $\Delta tu_\infty/d = 3.42$, which corresponds to a phase difference of 150°. The middle cylinder leads the rear one for about $\Delta tu_\infty/d = 3.34$, which corresponds to a phase difference of 146.6°.

The flow field is visualized using vorticity contours in Figure 25 for one period. Overall, we see that the vibration magnitudes and phase differences are consistent with what were found from the curves. Large vertical distances are observed in Figure 25B,E between the last two cylinders. In fact, a check of the previous displacement curves shows that the maximum relative distance is $(\Delta y)_{\max} = 1.89$ in this simulation. Meanwhile, it is observed that the flow field has two distinct regions: in the near-field region, we see complex vortex interactions; in the far-field region, vortices are well organized, with an upper row of negative vortices and a lower row of positive vortices.

The parallelization speedup is tested on this case using up to 240 processors. Figure 26 shows the scalability curve for different schemes. It is seen that when processor numbers are smaller than 120, the solver obtains an almost linear speedup. As the processor number further increases, the number of cells on each processor decreases, and interprocessor communication takes a larger portion of the overall time and slows down the speedup. It is interesting to notice that the scalability improves as the scheme order (denoted by N) increases. This is consistent with the fact that the computation cost on each processor is of $\mathcal{O}(N^2)$, whereas the communication cost is of $\mathcal{O}(N)$.

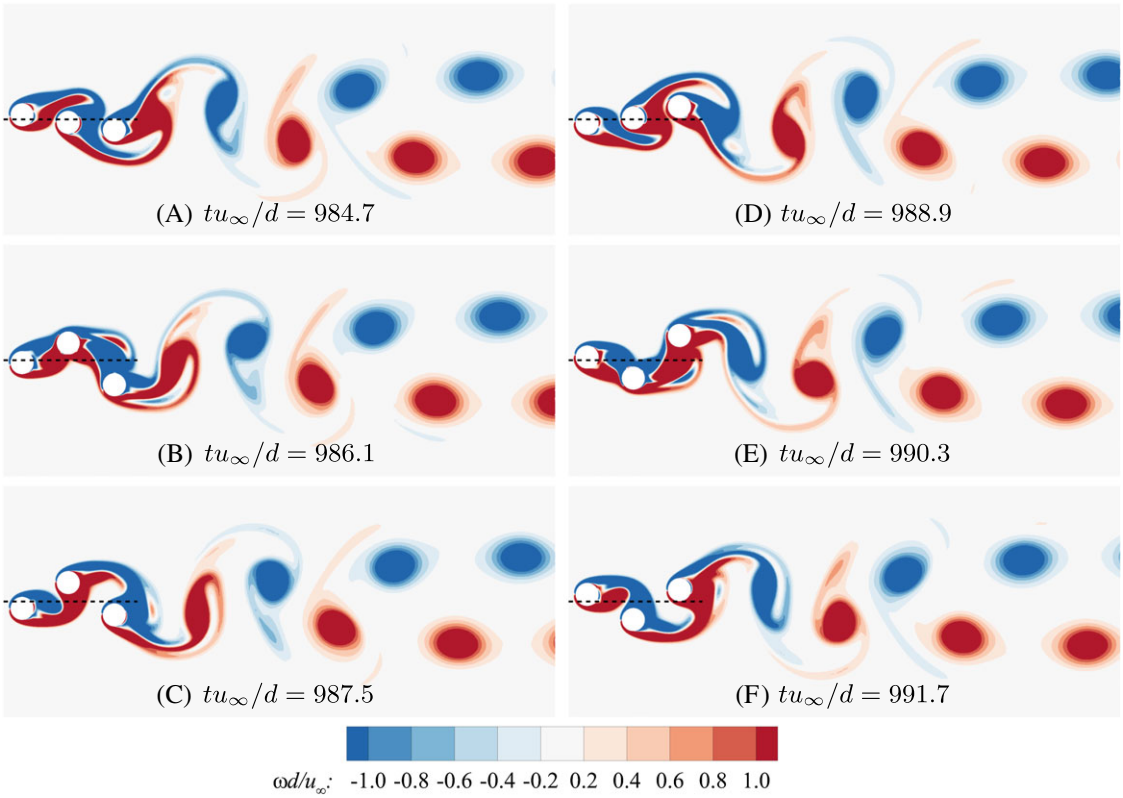

FIGURE 25 Vorticity contours in a period for vortex-induced vibrations of a row of three cylinders (dashed line represents mean position) [Colour figure can be viewed at wileyonlinelibrary.com]

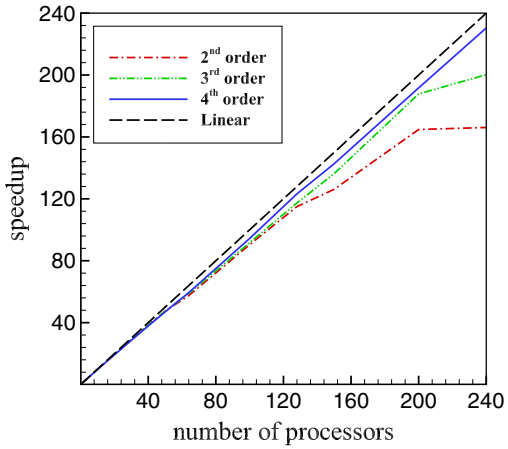

FIGURE 26 Parallelization speedup on vortex-induced vibrations of three cylinders in a row [Colour figure can be viewed at wileyonlinelibrary.com]

5 | CONCLUSIONS

We have successfully developed a new high-order solver for simulating VIVs. This solver is based on the SD method for unstructured grids with mixed elements, a new nonuniform sliding-mesh method for overcoming the deformation limitation of traditional conforming meshes, and a monolithic approach for coupling fluid and structure motions seamlessly. The solver is also successfully parallelized using the message passing interface. Through numerical tests, we have shown that this solver is high-order accurate for both inviscid and viscous flows. Meanwhile, by simulating the VIV of a single cylinder, we have demonstrated that this solver is able to accurately capture all the flow features of VIV, and the present results are found to agree very well with previously published ones. The advantages of this solver have been further

exploited by simulating VIVs of very closely placed tandem cylinders: good mesh quality is ensured across sliding interfaces where a traditional conforming mesh would most likely fail in such challenging situation. The simulation results also reveal that the trailing cylinder acts like a tail of the leading cylinder and, therefore, has larger oscillation magnitudes. Finally, a simulation of a row of three cylinders has been performed, which successfully demonstrates the solver's capability in dealing with multiple closely placed vibrating objects with large relative displacements. It is also interesting to learn from the simulation results that along the flow direction, the vibration magnitude of each cylinder increases consistently. Finally, from this test, the solver is shown to have a good parallel speedup for VIV simulation. These features altogether make the solver ideal for studying VIV problems at very challenging conditions.

ACKNOWLEDGEMENTS

Zihua Qiu would like to acknowledge the China Scholarship Council for supporting him as a visiting student at George Washington University. Chunlei Liang is thankful for support from the Office of Naval Research (ONR) through the Young Investigator Program (YIP) monitored by Dr Ki-Han Kim.

ORCID

Zihua Qiu https://orcid.org/0000-0002-5045-0064

REFERENCES

1. Blevins RD. *Flow-Induced Vibration*. New York, NY: Van Nostrand Reinhold; 1990.
2. Feng CC. *The Measurement of Vortex Induced Effects in Flow Past Stationary and Oscillating Circular and D-Section Cylinders* [master's thesis]. Vancouver, Canada: The University of British Columbia; 1968.
3. Khalak A, Williamson CHK. Dynamics of a hydroelastic cylinder with very low mass and damping. *J Fluids Struct*. 1996;10(5):455-472.
4. Khalak A, Williamson CHK. Motions, forces and mode transitions in vortex-induced vibrations at low mass-damping. *J Fluids Struct*. 1999;13(7-8):813-851.
5. Mittal S, Kumar V. Finite element study of vortex-induced cross-flow and in-line oscillations of a circular cylinder at low Reynolds numbers. *Int J Numer Meth Fluids*. 1999;31(7):1087-1120.
6. Jeon D, Gharib M. On circular cylinders undergoing two-degree-of-freedom forced motions. *J Fluids Struct*. 2001;15(3-4):533-541.
7. Jauvtis N, Williamson CHK. The effect of two degrees of freedom on vortex-induced vibration at low mass and damping. *J Fluid Mech*. 2004;509:23-62.
8. Prasanth TK, Behara S, Singh SP, Kumar R, Mittal S. Effect of blockage on vortex-induced vibrations at low Reynolds numbers. *J Fluids Struct*. 2006;22(6):865-876.
9. Bernitsas MM, Raghavan K, Ben-Simon Y, Garcia EM. VIVACE (Vortex Induced Vibration Aquatic Clean Energy): a new concept in generation of clean and renewable energy from fluid flow. *J Offshore Mech Arctic Eng*. 2008;130(4):041101.
10. Bernitsas MM, Ben-Simon Y, Raghavan K, Garcia EM. The VIVACE converter: model tests at high damping and Reynolds number around 105. *J Offshore Mech Arctic Eng*. 2009;131(1):011102.
11. Williamson CHK, Govardhan R. Vortex-induced vibrations. *Annu Rev Fluid Mech*. 2004;36:413-455.
12. Williamson CHK, Govardhan R. A brief review of recent results in vortex-induced vibrations. *J Wind Eng Ind Aerodyn*. 2008;96(6):713-735.
13. Kopriva DA, Kolias JH. A conservative staggered-grid Chebyshev multidomain method for compressible flows. *J Comput Phys*. 1996;125:244-261.
14. Kopriva DA. A conservative staggered-grid Chebyshev multidomain method for compressible flows. II. A semi-structured method. *J Comput Phys*. 1996;128:475-488.
15. Kopriva DA. A staggered-grid multidomain spectral method for the compressible Navier-Stokes equations. *J Comput Phys*. 1998;143:125-158.
16. Liu Y, Vinokur M, Wang ZJ. Spectral difference method for unstructured grids I: basic formulation. *J Comput Phys*. 2006;216:780-801.
17. Wang ZJ, Liu Y, May G, Jameson A. Spectral difference method for unstructured grids II: extension to the Euler equations. *J Sci Comput*. 2007;32:45-71.
18. Balan A, May G, Schöberl J. A stable high-order spectral difference method for hyperbolic conservation laws on triangular elements. *J Comput Phys*. 2012;231(5):2359-2375.
19. Zhang B, Liang C. A simple, efficient, and high-order accurate curved sliding-mesh interface approach to spectral difference method on coupled rotating and stationary domains. *J Comput Phys*. 2015;295:147-160.
20. Zhang B, Liang C, Yang J, Rong Y. A 2D parallel high-order sliding and deforming spectral difference method. *Comput Fluids*. 2016;139:184-196.
21. Thomas PD, Lombard CK. Geometric conservation law and its application to flow computations on moving grids. *AIAA J*. 1979;17:1030-1037.

22. Rusanov VV. Calculation of interaction of non-steady shock waves with obstacles. *J Comput Math Physics USSR*. 1961;1:261-279.
23. Persson P-O, Bonet J, Peraire J. Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains. *Comput Methods Appl Mech Eng*. 2009;198:1585-1595.
24. Mavriplis CA. *Nonconforming Discretizations and A Posteriori Error Estimates for Adaptive Spectral Element Techniques* [PhD thesis]. Cambridge, MA: Massachusetts Institute of Technology; 1989.
25. Ruuth S. Global optimization of explicit strong-stability-preserving Runge-Kutta methods. *Math Comput*. 2006;75(253):183-207.
26. Karypis G, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J Sci Comput*. 1999;20:359-392.
27. Erlebacher G, Hussaini MY, Shu C-W. Interaction of a shock with a longitudinal vortex. *J Fluid Mech*. 1997;337:129-153.

How to cite this article: Qiu Z, Zhang B, Liang C, Xu M. A high-order solver for simulating vortex-induced vibrations using the sliding-mesh spectral difference method and hybrid grids. *Int J Numer Meth Fluids*. 2019;90:171–194. https://doi.org/10.1002/fld.4717