Computers and Fluids 196 (2020) 104340

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Efficient FSI solvers for multiple-degrees-of-freedom flow-induced vibration of a rigid body

Methma M. Rajamuni*, Mark C. Thompson, Kerry Hourigan

Fluids Laboratory for Aeronautical and Industrial Research (FLAIR), Department of Mechanical and Aerospace Engineering, Monash University, Clayton, Victoria 3800, Australia

ARTICLE INFO

Article history:
Received 30 November 2018
Revised 15 July 2019
Accepted 19 October 2019
Available online 24 October 2019

Keywords:
Fully-coupled
FSI solver
OpenFOAM
Flow-induced vibration
An elastically mounted bluff body
A tethered body

ABSTRACT

Flow-induced vibration (FIV) of a bluff body is a complex fluid-structure interaction (FSI) problem, for which analytical solutions generally don't exist. Therefore, such problems generally need to be examined with experimental methods or computational simulations. Researchers have developed various numerical methodologies to solve FSI problems using either conforming or non-conforming grid methods. However, these methods are not always ideal for flow-induced vibration problems, as they can be computationally expensive or generate low accuracy solutions, especially for the cases with large body displacements. FIV problems often require long simulation times because, for a given parameter set, the transient flow solution time can be long and generally at least 10 oscillation cycles are required to simulate the representative long term behaviour. Thus, to gain a clear understanding and enhance knowledge of FIV of a bluff body, it is advantageous to have an efficient numerical methodology. We have developed two efficient fully-coupled FSI solvers to accurately predict the FIV of an elastically mounted bluff body and a tethered bluff body. These FSI solvers were developed based on the widely used open-source CFD package *OpenFOAM*. In these solvers, the fluid flow was modelled in a reference frame attached to the centre of mass of the solid body, so that a non-deforming grid can be employed. A predictor-corrector iterative method was used to enable strong coupling between the solid motion and the fluid flow. Each of the FSI solvers was validated against previously reported investigations. While efficient, the limitation of these FSI solvers is that they can only be used to examine the nature of FIV of a single, rigid bluff body.

1. Introduction

Flow-induced vibration (FIV) problems of bluff bodies are confronted in a wide range of engineering fields: fluid mechanics, structural mechanics, vibrations, computational fluid dynamics (CFD) and acoustics. When a fluid flows past a bluff solid structure, a large amplitude fluctuating pressure force can develop near the rear of the structure, leading to an unsteady wake. FIV is primarily excited by this unsteadiness of the wake. Vortex-induced vibration, or (VIV), is a category of FIV, occurring through the synchronisation of structural vibration with the vortex shedding. This self-limited vibration state can be expected to be excited when the vortex shedding frequency is sufficiently close to the natural frequency of the solid system. FIV can cause fatigue damage or even a failure of a structure, and therefore it is a crucial consideration for the design of many engineering systems. Some examples are bridges, chimney stacks, aircraft, ground vehicles, submarines, tethered structures, buoyancy and spar hulls, pipelines, cable-laying and offshore structures.

Due to the complex nature of FIV problems concerning bluff body flows, it is difficult to obtain an analytical solution. Experiments or computational simulations are the only feasible options for investigating FIV. To reveal the fundamentals of the field, a huge volume of experimental studies has been conducted on FIV on generic shapes. The major findings of these studies can be found from the comprehensive reviews of [1,2,28,29,39,47,48,50] and [9].

In the literature, phenomenological models have been developed following the idea of wake oscillators, to predict FIV. These models were based on the van der Pol or Rayleigh equation, which models a self-sustained, stable and nearly harmonic oscillation of finite amplitude [10,51]. These models still need to be improved to accurately predict FIV. Moreover, these models are always required to be fine tuned based on the experimental observations and have a limited contribution toward the understanding of FIV problems. The experimental studies have their own limitations, for example, difficulties in the visualization of the wake, especially for the case of complex and three-dimensional geometries, and diffi-

* Corresponding author.
E-mail address: methma.mm@gmail.com (M.M. Rajamuni).

https://doi.org/10.1016/j.compfluid.2019.104340
0045-7930/

culties of obtaining accurate force measurements, as a result of external noise. Therefore, computational simulations are essential to advance the knowledge of FIV. Compared to experimental studies, fewer computational studies have been reported on FIV, as it is difficult to implement an accurate, stable and efficient numerical methodology.

In FIV problems of bluff bodies, the flow of the fluid and the motion of the solid need to be solved in a coupled manner, as both the fluid and the solid body have their own motion. There exist two main fluid-structure interaction approaches, which have been used to solve FIV problems of bluff bodies for many years. One is the *conforming grid* method and the other is the *non-conforming grid* method [15]. The conforming grid method is the conventional method, which treats the interface between the fluid and the solid as a physical boundary, and deforms the grid of the fluid domain according to the motion/or deformation of the solid structure. This method is also known as the arbitrary Lagrangian–Eulerian (ALE) method [16,42] or adaptive-mesh method. This method consists of three different solver algorithms; a fluid flow solver, a solid deformation solver (if the solid is a flexible body), and an internal grid deformation solver. At each time step, the method needs to be proceed through all three solvers. The limitations of this method arise from the grid deformation step. This step can be quite challenging and time-consuming, especially for 3*D* problems. Moreover, for the scenarios involving large translational or rotational structural motion, the grid can be severely distorted, leading to a less accurate solution. The studies of [8,14,49] are some examples of using a conforming method for FIV problems. Although conforming methods have been widely used for many fluid-structure interaction problems, it is not an efficient method for FIV problems.

For FSI problems, widely used non-conforming grid methods are the immersed boundary (IB) methods. These methods treat the boundary between the fluid and the solid as a virtual interface, and the related interface conditions are imposed on the model equations as constraints so that a non-conforming grid can be employed. The classical IB method was first introduced by [30], to simulate blood flow in the human heart. As re-meshing is not required in IB methods, and it can be applied to a wide variety of problems including complex geometries and large structural deformations. Some examples are: FSI problems in heart values [5,13,20]; bio-film processes [7]; particulate flows [45]; flapping wings [27]; flow-induced vibration problems [21,40,41]. In this method, the structural grid is arbitrarily immersed in the background fluid grid with a moving boundary. The motion of the immersed boundary is usually taken into account by adding a fictitious body force in the governing equations. The major disadvantage of these methods is the reduced accuracy of the solution near the fluid-structure interface, due to IB methods typically smearing out the sharp interface to a layer of the thickness of the order of the mesh width. This limits the IB method's applicability to high Reynolds number flows. Beside thick immersed boundaries, IB methods can result in less accurate predictions, as the Dirichlet boundary condition is not able to be applied strongly on the interface.

As a remedy to the low accuracy due to the thick-interface of IB methods, a set of sharp-interface IB methods has been recently developed [5,21,25]. In these methods, the resolution near the fluid-solid interface is increased enabling a sharp interface. Cut-cell methods, immersed interface methods, and hybrid Cartesian immersed boundary methods are some examples of these methods, which have increased the accuracy of the solution, but the complexity of the solution process has also increased. Even for these improved methods that have higher-order accuracy at a boundary, because the boundary translates through the grid, it still restricts the resolution achievable at a boundary to the size of the cell size.

Compared to the adaptive mesh methods and immersed boundary methods, a flow-induced vibration problem of a rigid bluff body can be solved accurately and efficiently when the fluid flow is modelled in a body-fixed reference frame. This is a non-conforming grid method, which solves the fluid flow and the solid body motion in a coupled manner without deforming the grid. This method has been previously used in the studies of [3,4,22–24] to investigate the forced vibration and the free vibration of an elastically mounted cylinder. They have implemented this numerical methodology based on the spectral element method, which has higher order accuracy. As the spectral element methods have mainly been developed for direct numerical simulations, applications were limited to low Reynolds number flows. This article develops a fully coupled FSI solver in the OpenFOAM environment to accurately and efficiently predict the flow-induced vibration problems of both elastically mounted and tethered bodies. OpenFOAM is an open source CFD package which comes with a range of inbuilt turbulence models. Therefore, this solver can be easily adapted for high Reynolds number flows. Although this method is efficient and accurate, it can only be applied for the investigations of FIV problems of a single, rigid bluff body. As the fluid flow is modelled in a reference frame attached to the centre of the mass of the solid body, it cannot be used for FIV problems of multiple bodies or other FSI problems involving flexible bodies.

Our previous articles [34–37] provide a general outline of this method. This article presents a detailed description of the numerical methodology, including the implementation process in OpenFOAM. The structure of present article is as follows: Section 2 provides a brief overview of OpenFOAM including discussions on the algorithm of the *icoFoam* flow solver for the laminar flows and validation studies for non-VIV studies; following this, Sections 3 and 4 discuss the numerical approaches used for the flow-induced vibration problems of an elastically mounted body and a tethered body, respectively, providing detailed descriptions of the FSI systems, FSI solver algorithms, implementations of the solvers in OpenFOAM, and validation studies; finally, the article ends with conclusions in Section 5.

2. Simulations in openfoam

The widely used open-source CFD package, OpenFOAM, is made available by OpenCFD Ltd and distributed freely via https://www.openfoam.com for Linux operating systems. This CFD package is developed based on the finite-volume method, and it has the capability of performing simulations on multiple processors in parallel, achieving good parallel efficiencies. OpenFOAM is a framework for developing application executables that use packaged functionality contained within a collection of approximately 100 C+ libraries [43]. OpenFOAM contains approximately 250 pre-built applications that fall into *solver* and *utility* categories. Solvers are designed to solve a specific problem in continuum mechanics while *utilities* are designed to perform tasks that involve data manipulation. OpenFOAM solvers are capable of handling a wide range of problems in fluid dynamics. Users can develop new solvers, utilities and libraries with some pre-requisite knowledge of the underlying method, physics and programming techniques involved.

OpenFOAM comes with both pre- and post-processing environments. The interface to the pre- and post-processing environments are themselves OpenFOAM utilities, thereby ensuring consistent data handling across all environments. OpenFOAM has a limited graphical interface. Therefore, the utility *paraFoam* provides visualisation capabilities for the grid and the predictions by connecting to the data analysis and visualization application ParaView.

Typically, an OpenFOAM case contains mainly three types of directories, namely the *constant* directory, the *system* directory and the *time* directories. Only a brief description of the file system is

given below, and readers are referred to the OpenFOAM user guide [43] for more details. The constant directory contains the full description of the case grid in a subdirectory called *polyMesh*. It also contains the files specifying physical and turbulent properties for the application concerned, *e.g.* the file *transportProperties*. The system directory contains the files associated with the control parameters and solution procedures. It should contain at least the *controlDict, fvSchemes* and *fvSolution* files. The *controlDict* file specifies the control parameters including the start/end time, the time step and parameters for data output. The *fvSchemes* file determines the discretization schemes used in the solution, while the *fvSolution* file determines the equation solver algorithms, tolerances and other algorithm controls. A case directory contains individual files for each time instance. The name of each time directory is based on the simulated time at which the data is written. A time directory contains individual files of data for particular fields, *e.g.* velocity and pressure fields.

2.1. Governing equations of non-FIV simulations

The fluid is assumed Newtonian and incompressible and the Navier–Stokes and incompressibility equations given in Eqs. (1) and (2) describe the motion of the fluid.

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - \frac{1}{\rho}\nabla P + \frac{\mu}{\rho}\nabla^2\mathbf{u}. \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{2}$$

Here, $\mathbf{u} = \mathbf{u}(x, y, z, t)$ is the velocity vector of the fluid at a given location and at a given time, P is the pressure, ρ is the fluid density, assumed constant. Given constant density, it is usual to eliminate explicit reference to the density by introducing the kinematic pressure $p = P/\rho$, and the kinematic viscosity $\nu = \mu/\rho$. OpenFOAM facilitates simulations in the dimensional form. For the present direct numerical simulations, the only fluid property that is required to be specified is the kinematic viscosity, ν, which can be obtained by $\nu = UD/Re$, where U is the freestream velocity, D is the diameter of the sphere/cylinder, and Re is the Reynolds number.

2.2. The icoFOAM flow solver for non-VIV simulations

OpenFOAM has a wide range of standard solvers designed for applications in different categories of continuum mechanics. Flow in the laminar regime has been considered so far. The pre-built *icoFoam* solver is considered to be appropriate for the present simulations, as it is a transient solver for the incompressible, laminar flow of Newtonian fluids. The *icoFoam* solver was implemented according to the PISO (Pressure Implicit with Splitting of Operator) algorithm introduced by [17]. This algorithm approximates the spatially and temporally discretized fluid equations with an order of accuracy $O(\delta t^2)$, with δt the time step. The widely used PISO algorithm is generally stable and relatively easy to implement.

2.2.1. Outline of the PISO algorithm

The PISO algorithm integrates the Navier–Stokes equations forward in time using a predictor step followed by several corrector steps. In the predictor step, the discretised momentum equation (we will refer to this as *the velocity equation*) is solved implicitly for a new velocity field with the previous pressure and velocity fields. An equation is derived combining the discretised continuity equation and momentum equations (we will refer to this as *the pressure equation*). In the corrector step, a velocity field and a pressure field are found that are able to satisfy both the continuity and the momentum equations. This is done by first solving the above-mentioned pressure equation for the pressure (with the velocity found in the predictor step or the previous corrector step), and then solving the momentum equation for the velocity. At the end of the time step, the velocity and the pressure fields found from the last corrector step are taken as the new velocity and pressure fields. The next section describes the PISO algorithm used in the icoFoam solver in more detail.

[17] shows that this algorithm needs, at least, two corrector steps to achieve the desired accuracy. He showed that with two corrector steps, the velocity and pressure can be approximated to order of accuracy $O(\delta t^4)$ and $O(\delta t^3)$, respectively. Adding another corrector step would increase the accuracy of the approximations, but it will unnecessarily increase the computational time because the order of accuracy of the algorithm is only $O(\delta t^2)$. Therefore, we used only two corrector steps in all of our simulations.

2.2.2. PISO algorithm used in the icoFoam solver

This algorithm states the steps of solving Navier–Stokes equations for the time step $(n+1)$ with the values of $\mathbf{u}$ and p at previous time step (n) for two corrector steps. Let the superscripts *, **, *** denote intermediate field values obtained during the splitting process. A semi-discrete form of the momentum Eq. (1) can be given as

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -A'\mathbf{u}^* + H'(\mathbf{u}^*) - \nabla p^n, \tag{3}$$

where $-A'\mathbf{u}^*$ and $H'(\mathbf{u}^*)$ represent the diagonal and non-diagonal elements of the discretized convection and diffusion terms, respectively. This equation can be rearranged as:

$$A\mathbf{u}^* = H(\mathbf{u}^*) - \nabla p^n, \tag{4}$$

with $A = 1/\Delta t + A'$ and $H(\mathbf{u}^*) = H'(\mathbf{u}^*) + \mathbf{u}^n/\Delta t$.

Predictor step: Solve the momentum equation given in Eq. (4) for the first intermediate value of the velocity field $(\mathbf{u}^*)$ with previous values of pressure (p^n) and velocity $(\mathbf{u}^n)$. This $\mathbf{u}^*$ in general will not satisfy the zero divergence condition (given in Eq. (2)).

First corrector step: A new velocity field $(\mathbf{u}^{**})$ together with a corresponding new pressure field (p^*) are now considered such that the zero-divergence condition

$$\nabla\cdot\mathbf{u}^{**} = 0 \tag{5}$$

is met. For this, the momentum Eq. (4) is taken as

$$A\mathbf{u}^{**} = H(\mathbf{u}^*) - \nabla p^*. \tag{6}$$

Here, non-diagonal terms of the convection and diffusion terms have been treated explicitly $(H(\mathbf{u}^*))$; we will see the reason shortly. Eq. (6) and (5) are used to derive the pressure equation

$$\nabla\left(\frac{\nabla p^*}{A}\right) = \nabla\cdot\left(\frac{H(\mathbf{u}^*)}{A}\right). \tag{7}$$

Then the pressure equation is solved for p^* with velocity field found in the predictor step $(\mathbf{u}^*)$, and afterwards Eq. (6) is solved for $\mathbf{u}^{**}$.

Second corrector step: A new velocity field $(\mathbf{u}^{***})$ together with its corresponding new pressure field (p^{**}) are formulated to meet the zero-divergence condition

$$\nabla\cdot\mathbf{u}^{***} = 0. \tag{8}$$

The momentum Eq. (4) is taken in a semi explicit form as

$$A\mathbf{u}^{***} = H(\mathbf{u}^{**}) - \nabla p^{**}, \tag{9}$$

the corresponding pressure equation is therefore

$$\nabla\left(\frac{\nabla p^{**}}{A}\right) = \nabla\cdot\left(\frac{H(\mathbf{u}^{**})}{A}\right). \tag{10}$$

Solving Eq. (10), p^{**} can be found. Then $\mathbf{u}^{***}$ can be found from Eq. (9). Finally, set $\mathbf{u}^{***}$ as $\mathbf{u}^{n+1}$ and p^{**} as p^{n+1}.

Table 1
A comparison of time-averaged drag coefficient, $\overline{C}_d$, time-averaged lift coefficient, $\overline{C}_l$, and Strouhal number, *St*, of a rigidly mounted sphere, at $Re = 300$, 500 and 1000. Values calculated for $\overline{C}_d$, $\overline{C}_l$, and *St* closely match values of other studies.

	$Re = 300$			$Re = 500$			$Re = 1000$		
Study	$\overline{C}_d$	$\overline{C}_l$	*St*	$\overline{C}_d$	$\overline{C}_l$	*St*	$\overline{C}_d$	St_1	St_2
Present study	0.665	0.070	0.137	0.57	0.06	0.18	0.49	0.19	0.33
[6]	0.665	0.065	0.136	–	–	–	–	–	–
[11]	0.658	0.067	0.134	–	–	–	–	–	–
[19]	0.656	0.069	0.137	–	–	–	–	–	–
[26]	–	–	–	0.55	–	–	0.46	–	–
[31]	–	–	–	0.56	0.05	0.15	0.46	0.185	0.33
[38]	–	–	–	–	–	0.18	–	0.2	–
[44]	–	–	–	–	–	0.167	–	0.2	0.35

2.3. Numerical discretization and solver algorithms

OpenFOAM was developed based on the finite-volume method (FVM), similar to many computational fluid dynamics packages. In general, the finite-volume approach is based on the conservation of some quantity, *i.e.*, what goes into the control volume through the sides accumulates in the control volume. In this method, the governing equations are integrated over all finite volumes of the computational domain. The finite-volume method requires a spatial domain to discretize into contiguous cells. Dependent variables are principally stored at cell centroids, although they may be stored in cell faces or vertices. OpenFOAM provides considerable freedom in grid generation and manipulation, especially when the geometry is complex or changes over time.

OpenFOAM offers the freedom of choosing appropriate discretization schemes from a wide selection, for each and every term in the governing equations. This is done through the *fvSchemes* file in the *system* directory. The time-derivative term, $\partial \boldsymbol{u}/\partial t$, was discretized based on the backward Euler approach, which is implicit and second-order accurate. The gradient (∇) and Laplacian (∇^2) operators were discretized by the second-order Gauss scheme using linear interpolation. The divergence ($\nabla \cdot$) operator was also discretized similarly but with the $\Gamma = 0.5$ interpolation scheme. To calculate the surface normals for the Laplacian, a blend of the corrected (which is unbounded, second-order and conservative) and uncorrected (bounded, first-order and non-conservative) schemes were used with a blend factor of 0.5.

The velocity equation mentioned in the predictor step was solved using a Preconditioned (Bi-) Conjugate Gradient (PBiCG) iterative method preconditioned with a Diagonal Incomplete-Lower-Upper (DILU) decomposition. The pressure equation defined in a corrector step was solved using a Preconditioned Conjugate Gradient (PCG) iterative method preconditioned with the Diagonal Incomplete Cholesky (DIC) decomposition. It is possible to use the simpler Conjugate Gradient method with the DIC preconditioner for the pressure equation (see the Eq. (7)), since the Laplacian operator yields a symmetric and positive definite matrix when discretised. However, the discretized velocity equation will not be symmetric due to the nonlinear convection term, thus dictating the use of the Bi-conjugate gradient method with the DILU preconditioner.

2.4. Validation studies of flow past a bluff body

The flow past a rigidly mounted sphere was investigated at $Re = 300$, 500 and 1000. The computed values for the time-averaged drag and lift coefficients, $\overline{C}_d$ and $\overline{C}_l$, respectively, and the Strouhal number, *St*, are listed in Table 1. As the lift coefficient is negligible at $Re = 1000$, a secondary Strouhal number is calculated instead of $\overline{C}_l$. As can be seen, the present results closely match values calculated in other studies [6,11,19,26,31,38,44].

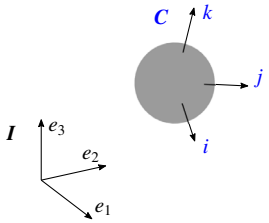


Fig. 1. The fixed frame $I(e_1, e_2, e_3)$ and the moving frame $C(i, j, k)$. The fluid body was modelled in the moving reference frame *C*, which is attached to the centre of mass of the solid body, rather than the absolute reference frame *I*.

3. FIV solver of an elastically mounted body

3.1. Governing equations

FIV of an elastically mounted rigid body placed at the centre of a huge fluid domain was considered for the present study. The fluid flow was modelled in the moving reference frame fixed to the centre of the mass of the solid body (see Fig. 1) so that a non-deforming grid can be used for the fluid domain. Since this frame accelerates according to the body vibration, it is a non-inertial reference frame. Therefore, the (momentum) Navier–Stokes equations given in Eq. (1) need to be adjusted by adding the acceleration of the frame, which is indeed the acceleration of the sphere, to the momentum equations, as a fictitious source term. For the case of an elastically-mounted rigid-body, the motion of the solid body was assumed to behave as a spring-mass-damper system, while the fluid was assumed incompressible and viscous.

The coupled fluid-solid system can be described by the Navier–Stokes equations given by Eq. (11), and the continuity equation given by (12), together with the governing equation for the motion of the solid by Eq. (13):

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u} - \ddot{\boldsymbol{x}}_{\boldsymbol{c}}, \tag{11}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{12}$$

$$m \ddot{\boldsymbol{x}}_{\boldsymbol{c}} + c \dot{\boldsymbol{x}}_{\boldsymbol{c}} + k \boldsymbol{x}_{\boldsymbol{c}} = \boldsymbol{f}_{\boldsymbol{l}}. \tag{13}$$

Here, $\boldsymbol{x}_{\boldsymbol{c}}$, $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$, and $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}$ are the solid displacement, velocity, and acceleration vectors, respectively. In addition, *m* is the mass of the body, *c* is the damping constant, *k* is the structural spring constant, and $\boldsymbol{f}_{\boldsymbol{l}}$ is the flow-induced integrated vector force acting on the solid body due to kinematic pressure and viscous shear forces acting on the body surface.

3.2. The fluid-structure solver

To solve the fully-coupled fluid-structure system defined by the Eqs. (11)–(13), an FSI solver was created (named *vivIcoFoam*). This solver is developed based on the OpenFOAM flow solver, icoFoam, that we discussed in Section 2.2. A predictor-corrector iterative method is used in the vivIcoFoam solver to handle the coupling between the solid displacement and the fluid flow. At the beginning of an iteration, the solid motion is obtained by explicitly predicting or implicitly correcting. Then, the fluid equations given in Eqs. (11) and (12) are solved with the predicted or subsequently corrected solid acceleration, and the fluid forces induced on the solid are calculated. Details of the predictor and corrector iterations at the $(n+1)$th time step can be elaborated as follows:

The predictor iteration

Initially, the sphere acceleration, $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}$, is predicted explicitly using the third-order polynomial extrapolation

$$\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,p_1)} = 3\,\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n)} - 3\,\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n-1)} + \ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n-2)}. \tag{14}$$

Then, the sphere velocity, $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$, and displacement, $\boldsymbol{x}_{\boldsymbol{c}}$, are estimated by integrating the predicted $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}$ and estimated $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$ by a third-order Adams–Moulton method by

$$\dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,p_1)} = \dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n)} + \frac{\delta t}{12}\left(5\,\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,p_1)} + 8\,\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n)} - \ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n-1)}\right) \tag{15}$$

and

$$\boldsymbol{x}_{\boldsymbol{c}}^{(n+1,\,p_1)} = \boldsymbol{x}_{\boldsymbol{c}}^{(n)} + \frac{\delta t}{12}\left(5\,\dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,p_1)} + 8\,\dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n)} - \dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n-1)}\right), \tag{16}$$

respectively, where δt is the time step. Finally, the fluid equations are solved with $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,p_1)}$, and the fluid force exerted on the sphere, $\boldsymbol{f}_{\boldsymbol{l}}^{(n+1,\,p_1)}$, is calculated for the coming corrector iteration.

***i*th corrector iteration**

Initially, the corrected value of $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}$ is calculated by solving the solid motion Eq. (13) with the values of $\boldsymbol{x}_{\boldsymbol{c}}$, $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$, and $\boldsymbol{f}_{\boldsymbol{l}}$ calculated in the predictor iteration or the previous corrector iteration by

$$\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_i)} = -\frac{c}{m}\,\dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,k_i)} - \frac{k}{m}\,\boldsymbol{x}_{\boldsymbol{c}}^{(n+1,\,k_i)} + \frac{1}{m}\,\boldsymbol{f}_{\boldsymbol{l}}^{(n+1,\,k_i)}, \tag{17}$$

where, $k_1 = p_1$ at the first corrector iteration, and $k_i = c_{i-1}$ for $i > 1$. For $i \geq 2$, the solid acceleration is relaxed to improve the convergence by

$$\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_i)'} = \ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_{i-1})} + \gamma\left(\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_i)} - \ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_{i-1})}\right), \tag{18}$$

where γ is the relaxation parameter. Note that relaxing the solution is only required for small mass ratio cases ($m^* < 1$), and otherwise γ is set to be 1. Then, the corrected values of $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$ and $\boldsymbol{x}_{\boldsymbol{c}}$ are updated by

$$\dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_i)} = \dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n)} + \frac{\delta t}{12}\left(5\,\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_i)} + 8\,\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n)} - \ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n-1)}\right) \tag{19}$$

and

$$\boldsymbol{x}_{\boldsymbol{c}}^{(n+1,\,c_i)} = \boldsymbol{x}_{\boldsymbol{c}}^{(n)} + \frac{\delta t}{12}\left(5\,\dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_i)} + 8\,\dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n)} - \dot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n-1)}\right), \tag{20}$$

respectively. Finally, the fluid equations are solved with $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,k_i)}$, and the fluid force exerted on the sphere, $\boldsymbol{f}_{\boldsymbol{l}}^{(n+1,\,c_i)}$, is calculated for the coming corrector iteration.

3.2.1. Termination of the iterative algorithm

The predictor-corrector iterative process is terminated once the convergence criteria are met. After the predictor iteration, several corrector iterations are performed until the relative error of the solid acceleration, e_a, and the relative error of the fluid force act on the solid, e_f, converge within a prescribed error bound, ϵ, as given in Eq. (21):

$$e_a = \frac{||\,\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_{i+1})} - \ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_i)}\,||_2}{||\,\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_{i+1})}\,||_2} < \epsilon \qquad \&$$
$$e_f = \frac{||\,\boldsymbol{f}_{\boldsymbol{l}}^{(n+1,\,c_{i+1})} - \boldsymbol{f}_{\boldsymbol{l}}^{(n+1,\,c_i)}\,||_2}{||\,\boldsymbol{f}_{\boldsymbol{l}}^{(n+1,\,c_{i+1})}\,||_2} < \epsilon, \tag{21}$$

where $||.||_2$ represents the Euclidean norm. The value of ϵ was chosen as 0.001 for simulations since it was found that further decreasing ϵ does not increase the accuracy of the solution. Typically, the FSI solver required 3 corrector steps. Fig. 2 displays histograms of number of corrector iterations consumed in the VIV simulations conducted at $Re = 2000$, $m^* = 3$, and $U^* = 5$, 6, 7, 8, 9, 10, 12, and 14. As can be seen, in most cases the number of corrector steps was less than 6, with the upper limit set to 15.

The temporal accuracy of the overall FSI solver is second-order, although the solution process for the solid motion is third-order accurate. This is because the PISO algorithm itself is of second-order accuracy. It is recalled that the fluid domain was modelled in a moving frame-of-reference. This motion is acknowledged through the outer domain velocity boundary conditions (except the outlet boundary). In this article, all the outer boundaries, except the outlet where a pressure condition is enforced, have a velocity condition prescribed on them. Once the predictor-corrector iterative process has completed, the velocity at the inlet boundaries is updated according to the velocity of the solid, $\dot{\boldsymbol{x}}_{\boldsymbol{s}}$, before proceeding to the next time step.

3.3. Implementation of the vivIcoFoam solver in OpenFOAM

The web page, https://openfoamwiki.net/index.php, describes how to develop a new OpenFOAM solver, by providing an example of 'how to add temperature to *icoFoam*'. The *vivIcoFoam* solver was implemented in OpenFOAM following those steps.

This solver was developed such that it can be used for both 2*D* simulations (such as cylindrical bodies) and 3*D* simulations. In addition, the solid motion can be restricted to the lift direction or can be allowed to move in all three directions. The solid motion parameters and other algorithm control settings should be prescribed in the *solidMotionData* file, which is in the *system* directory of a case, (see Appendix A for a sample of this file). The solver calculates the mass of the solid, $m = \frac{4}{3}\rho\pi\,(D/2)^3 m^*$, the damping constant, $c = 4\pi m\zeta/U^*$, and the spring constant, $k = 4\pi^2 m/(U^*)^2$, according to the values given for the non-dimensional parameters: mass ratio m^*, damping ratio ζ, and reduced velocity U^*.

This solver is designed to be used for a rectangular fluid domain. It reads the names of the boundary patches from the *boundaryToUpdate* file in the *constant* directory. A sample of this file can be found in Appendix A. This file also declares the type of the simulation, *i.e.* 2*D* or 3*D*, to recognize the inlet patches that need to be updated. The solver updates the velocity of the inlet boundaries and the pressure gradient of the solid boundary at the end of each time step. The solver writes the solid motion data (displacement, velocity and acceleration of the solid) together with the force coefficients to a csv file in the case directory.

3.3.1. vivIcoFoam algorithm

The steps of 'vivIcoFoam' algorithm, which solves fluid-solid coupled system Eqs. 11 – (13) are given below for the time step $(n+1)$ with the values of $\mathbf{u}$, p, $\boldsymbol{x}_{\boldsymbol{c}}$, $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$, and $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}$ at the previous time steps, n, $n-1$ and $n-2$ (Fig. 3 illustrates the flow chart of this algorithm.)

Step 1 [Initialization] Read the *solidMotionData* file in the *system* directory and the *boundarytToUpdate* file in the *constant* directory

Table 2
Computational time of flow-induced vibration of a cylinder.

Grid near the cylinder (circumferential × radial)	A^*_{rms}	$\overline{C}_d$	f^*	No. of shedding cycles	Computational time (h)
Fine (112 × 45)	0.41	1.67	1.00	10	***1.20***
Medium (168 × 80)	0.41	1.66	1.00	10	***3.41***
Extra fine (224 × 111)	0.41	1.66	1.00	10	***6.69***

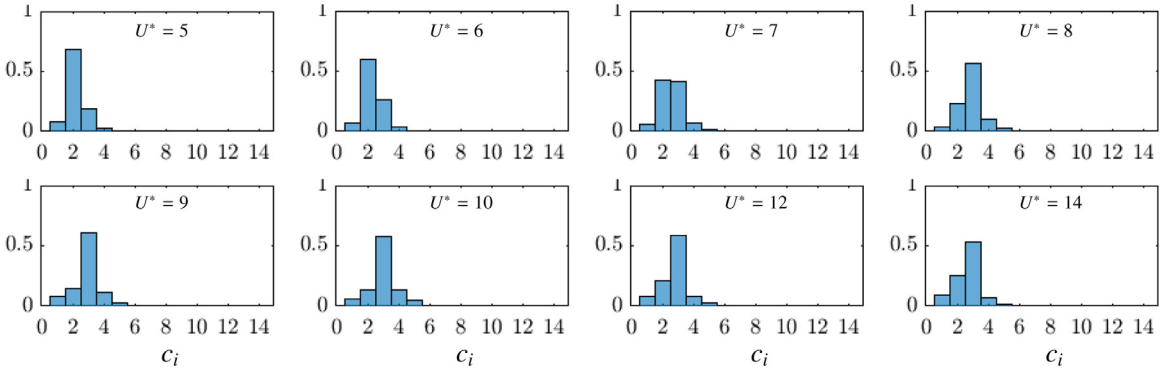


Fig. 2. Normalized histograms of number of corrector iterations, c_i used for the simulations of the vortex-induced vibration of a sphere at $Re = 2000$, $m^* = 3$, and reduced velocity $U^* = 5$, 6, 7, 8, 9, 10, 12, and 14.

of the case. Declare variables, and perform initializations required to update the boundary conditions and to write the output data. Calculate the dimensional parameters of the solid motion equation given in Eq. (13) according to the user defined non-dimensional parameters.

Step 2a [The predictor iteration] Predict the solid acceleration, $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}{}^{(n+1,\ p_1)}$, using a polynomial extrapolation as shown in Eq. (14). Then, estimate the solid velocity, $\dot{\boldsymbol{x}}_{\boldsymbol{c}}{}^{(n+1,\ p_1)}$, and displacement, $\boldsymbol{x}_{\boldsymbol{c}}{}^{(n+1,\ p_1)}$, using Eqs. (15) and (16), respectively.

Step 2b [A corrector iteration] Correct the value of $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}{}^{(n+1,\ c_i)}$ by solving the Eq. (17). Then, update the values of $\dot{\boldsymbol{x}}_{\boldsymbol{c}}{}^{(n+1,\ c_i))}$ and $\boldsymbol{x}_{\boldsymbol{c}}{}^{(n+1,\ c_i)}$ using Eqs. (19) and (20), respectively.

Step 3 [Solve fluid equations] PISO algorithm discussed in Section 2.2.2 was modified to solve Navier–Stokes Eqs. (11) and (12). Each form of momentum equations in PISO algorithm in Section 2.2.2, needs to include the solid acceleration term in the right. Thus, Eqs. (4), (6), and (9) become;

$$A\mathbf{u}^* = H(\mathbf{u}^*) - \nabla p^n + \ddot{\boldsymbol{x}}_{\boldsymbol{c}}{}^{(n+1)}, \tag{22}$$

$$A\mathbf{u}^{**} = H(\mathbf{u}^*) - \nabla p^* + \ddot{\boldsymbol{x}}_{\boldsymbol{c}}{}^{(n+1)}, \tag{23}$$

and

$$A\mathbf{u}^{***} = H(\mathbf{u}^{**}) - \nabla p^{**} + \ddot{\boldsymbol{x}}_{\boldsymbol{c}}{}^{(n+1)}, \tag{24}$$

respectively. Due to the new term in the momentum equation, pressure equation also needs to be modified accordingly. Thus, Eqs. (7) and (10) become

$$\nabla\left(\frac{\nabla p^*}{A}\right) = \nabla\cdot\left(\frac{H(\mathbf{u}^*) + \ddot{\boldsymbol{x}}_{\boldsymbol{c}}{}^{(n+1)}}{A}\right) \tag{25}$$

and

$$\nabla\left(\frac{\nabla p^{**}}{A}\right) = \nabla\cdot\left(\frac{H(\mathbf{u}^{**}) + \ddot{\boldsymbol{x}}_{\boldsymbol{c}}{}^{(n+1)}}{A}\right). \tag{26}$$

Step 04 [Calculate the fluid forces] With $\mathbf{u}^{n+1}$ and p^{n+1} found from the PISO algorithm, calculate fluid forces exerted on the sphere, $\boldsymbol{f}_{\boldsymbol{l}}{}^{(n+1)}$.

Step 05 [Loop from step 02b to step 04]. Proceed to the next iteration (to step 02b) with the newly calculated fluid forces, $\boldsymbol{f}_{\boldsymbol{l}}{}^{(n+1)}$. When it completes step 04, calculate the relative errors of fluid forces, e_f, and solid acceleration, e_a. Break the iterative loop if both e_f and e_a are less than the desired error tolerance, as shown in Eq. (21). If none of the convergence criteria is met, then continue until it reaches the maximum number of iterations.

Step 06 [Update boundary conditions and write motion data] Once the iterative process is complete, update the boundary conditions and write the solid motion data ($\boldsymbol{x}_{\boldsymbol{c}}$, $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$ and $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}$) and force coefficients in all three direction to a csv file. Then, proceed to the next time step.

Only the asymptotic state of a simulation was used for the analyses presented in this article. Sometimes, simulations need a long time to reach the asymptotic state. However, there is a run-time limitation of a maximum of 1 day on the supercomputer system used for the FIV cases. Therefore, in such a case, it is required to restart from the previously stopped time. In these circumstances, the *vivIcoFoam* solver writes the necessary solid-motion data of the last three time steps to an OpenFOAM file called *lastMotionData* in the *system* directory (a sample of this file also given in Appendix A). The solver updates this file at the time it writes the results for the fluid to a *time* directory.

3.4. Computational time

Simulations of flow-induced vibration of a bluff-body, in general, take longer than for a fixed body. Usually, a simulation takes a considerable amount of time to fully develop the vibration response and reach the asymptotic state. In addition, to accurately predict the time-average parameters, such as response amplitude, drag and lift coefficients, and the frequency of the vibration, the simulation was required to be extended for a number of vibration cycles sufficient for statistical averaging, once an asymptotic state had been reached. For highly periodic response, 10 oscillation cycles are found to be sufficient. However, if the response is not fully periodic or aperiodic, simulations are required to run for a larger number of oscillation cycles. The time taken for FIV simulations, depends on many parameters, for example, mass ratio, reduced velocity, Reynolds number, and the nature of the vibration.

To provide an idea of the computational time taken for 2*D* simulations discussed in Section 3.5.1, simulations were carried out on three levels of grids with a non-dimensional time step of 0.005, at $U^* = 5.5$. At this U^*, the cylinder vibrated sinusoidally with a large vibration amplitude. Table 2 tabulates the results obtained from each grid and the computational time utilized to run for 10

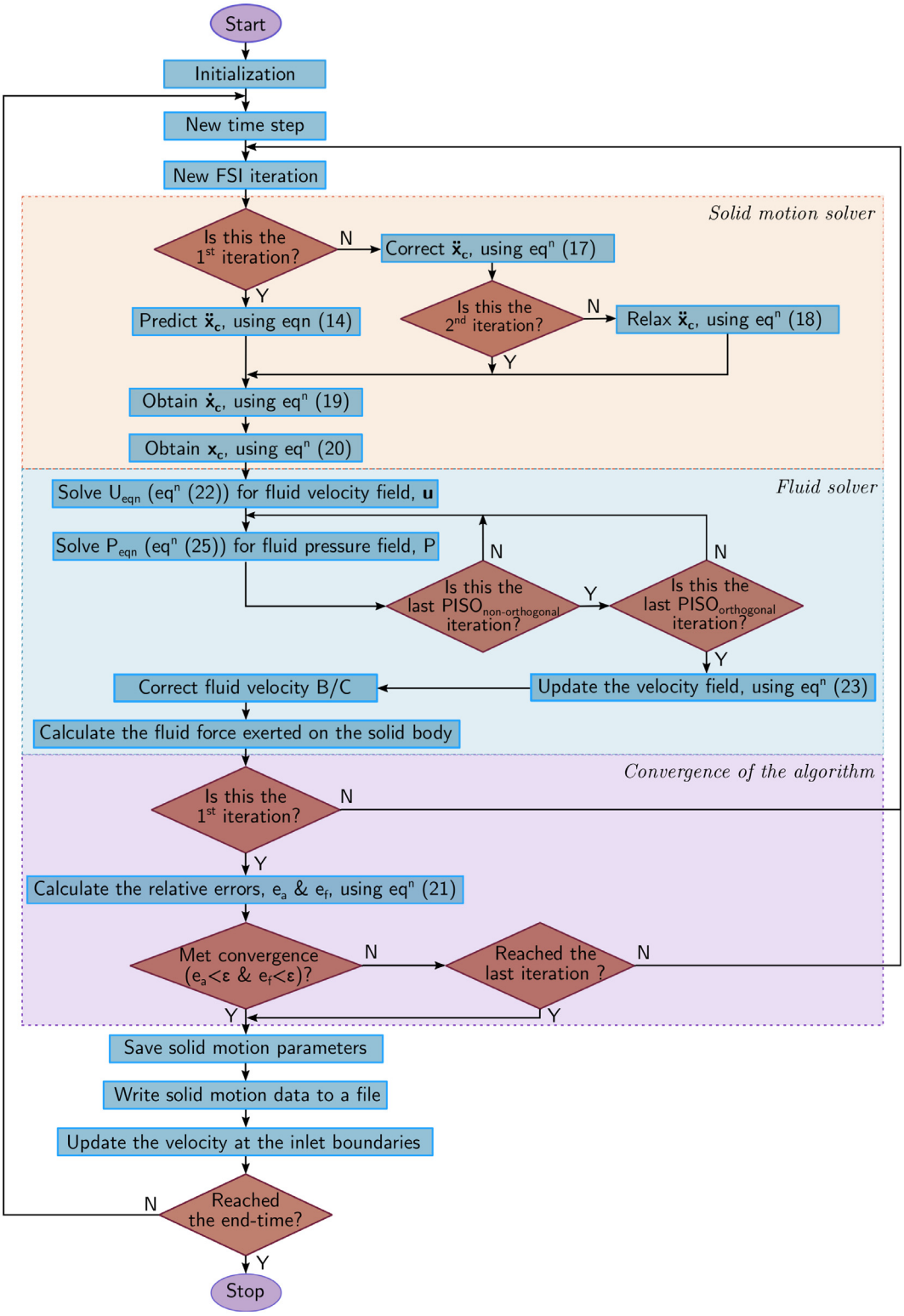


Fig. 3. Flow-chart of the FIV solver of an elastically mounted body. The number of non-orthogonal and orthogonal PISO iterations were set to 2 while the number of FSI iterations was set to 15.

shedding cycles (10 cycles are sufficient as the response is sinusoidal). The medium grid (29,296 cells) is fine enough for the simulations, as the results obtained with each grid have a good agreement. In the medium grid, it took only 3.41 h in real time to run for 10 shedding cycles, which is approximately equivalent to a non-dimensional simulation time of 55.

The 3*D* simulations conduced with a sphere (1.23 million cells) were run parallel on the Pawsey centre supercomputer Magnus using 48 computer cores. A simulation of an elastically mounted sphere of $m^* = 3$ took around 1–2 days, while a simulation of a tethered sphere of $m^* = 0.8$ took on average 1–3 days. The longer computational time of the tethered sphere case is mainly due to small mass ratio of the sphere. When the body is light, the system is intrinsically unstable and relaxation of the solution is required to stabilize the algorithm. Also, the FSI algorithm requires to cycle through a large number of corrector steps.

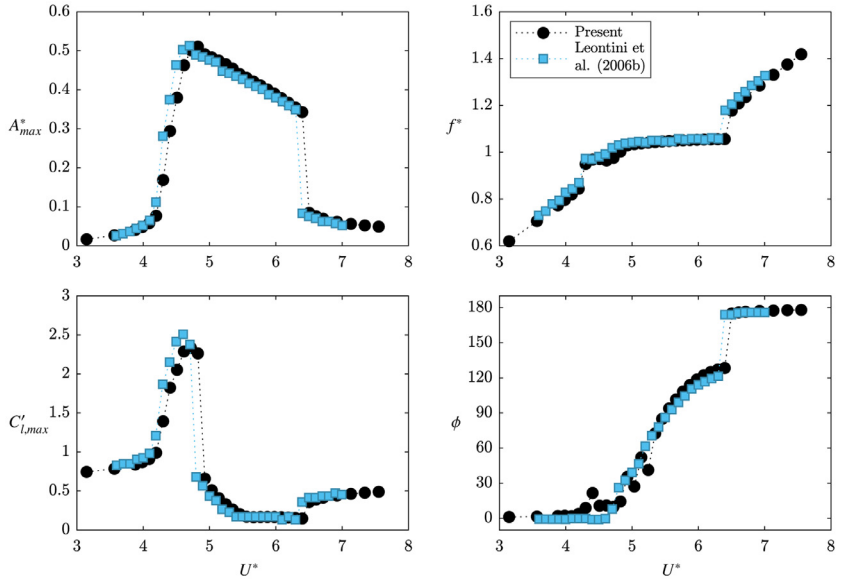


Fig. 4. Vortex-induced vibration response of a circular cylinder. Comparison of the present results, •, with the results of [24].

3.5. Results and validations

3.5.1. Vortex-induced vibration of a circular cylinder

A set of 2*D* simulations were conducted with an elastically mounted circular cylinder of mass ratio, $m^* = 10$ over the reduced velocity range $U^* = [3, 8]$, by fixing the Reynolds number and the damping ratio at $Re = 200$ and $\zeta = 0.01$, respectively. These parameters were chosen to compare the results with the observations of [24] and validate the vivIcoFoam solver. Fig. 4 compares present predictions of maximum vibration amplitude, A^*_m, the maximum fluctuation amplitude of the lift coefficient, $C'_{l,m}$, frequency ratio, $f^* = f/f_n$, and the average phase angle between the lift force and the cylinder displacement, ϕ, with the results of [24]. As the figure shows, the current predictions are almost identical with their findings with very minor differences (the percentage errors of A^*_m, $C'_{l,m}$, f^*, and ϕ are -8%, -8%, 1.8%, and 3.6%, respectively).

At this low Reynolds number, soon after the *initial vibration branch* ($4.2 \leq U^* \leq 4.7$), the *lower branch* appeared without a trace of the *upper branch*. Vibrations of the cylinder in the lower branch were highly periodic with a large vibration amplitude. Furthermore, the frequency ratio, f^*, at this branch was close to 1, indicating the resonance behaviour (see Fig. 4(b). To strengthen this validation study, the wake behind the cylinder was visualized using the vorticity field. Fig. 5 provides a comparison of our results with the vortical structures observed by [24] at the beginning of the lower branch for the reduced velocities $U^* = 5.0$, 5.2, 5.4, and 5.6. At $U^* = 5.0$, the shed vortices took a stable double-row configuration close to the rear of the cylinder, as [24] observed. The formation of this double-row configuration was delayed at $U^* = 5.2$. It was pushed further downstream at $U^* = 5.4$, and at $U^* = 5.6$, vortex loops were formed in a single-row configuration near the cylinder. This variation of the wake pattern matches closely with the findings of [24].

At this point, it is worth comparing the current predictions with the results of [32], who investigated the vortex-induced vibration of a circular cylinder by varying the Reynolds number from $Re = 60$ to 200. Although this study was conducted by varying the Reynolds number, the amplitude response curve they observed is identical to that of ours. In their study, the lower branch was from $Re = 86$ to 140. Similar to the current predictions, Prasanth and Mittal reported a double-row configuration in the vorticity field at the beginning of the lower branch and a gradual transition to a single-row configuration, as the Reynolds number increased [32], Fig. 9. The strong resemblance of our findings of other studies, the validity of the vivIcoFoam solver is proven.

3.5.2. Vortex-induced vibration of a sphere

Vortex-induced vibration of an elastically and transversely mounted sphere was examined at the Reynolds number of $Re = 2000$. The mass ratio of the sphere and the damping ratio were set to $m^* = 3$ and $\zeta = 0$, respectively, while the reduced velocity was varied in the range $3.5 \leq U^* \leq 10$. This mass ratio was chosen to minimize the cost of computations since one order of magnitude higher or lower mass ratio increases the computational cost substantially. This is because when the body is heaving, it takes a longer time to reach the asymptotic state, and when the body is light, the system is intrinsically unstable and needs to cycle through a large number of corrector steps of the FSI algorithm. The damping ratio of $\zeta = 0$ was chosen to enable the highest possible sphere response amplitude. Simulations were run for at least 10 vibration cycles after the asymptotic state was attained, to enhance the accuracy of the results, and the latter part of the signal was used for the statistical calculations.

In the literature of flow-induced vibration of a sphere, four different vibration modes, namely modes I–IV, have been identified. The first two modes of vibration occur as a result of the synchronization of the vortex shedding behind the body and areknown as vortex-induced vibration. These two modes usually appear in the range of $5 \lesssim U^* \lesssim 14$ with a smooth mode transition. [12] and [35] discuss differences of these two modes in detail. In short, mode I is the resonance response which is excited when the normalized velocity, $(U^*/f^*)St = 1$, and mode II can be observed for $1 < (U^*/f^*)St \lesssim 2$. Fig. 6(a) compares the present predictions of the sphere response amplitude, $A^*_{rms} = \sqrt{2}A_{rms}/D$, plotted against the normalized velocity, with the experimental results of [12] obtained with a transversely vibrating sphere of $m^* = 7$. As can be seen, the amplitude response curve strongly resembles that of [12]. In the synchronization regime, the time-averaged drag coefficient increased roughly by 90% from the value for a stationary sphere (see Fig. 6(b)), as reported by [46] using a tethered sphere.

[12] were the first to visualize the wake behind vibrating spheres. Using the digital particle image velocimetry (DPIV) technique, they found that the wake consists of a chain of two-sided hairpin loops for both modes I and II. Consistently, in our previous studies [34–36], we also observed two-streets of interlaced hairpin

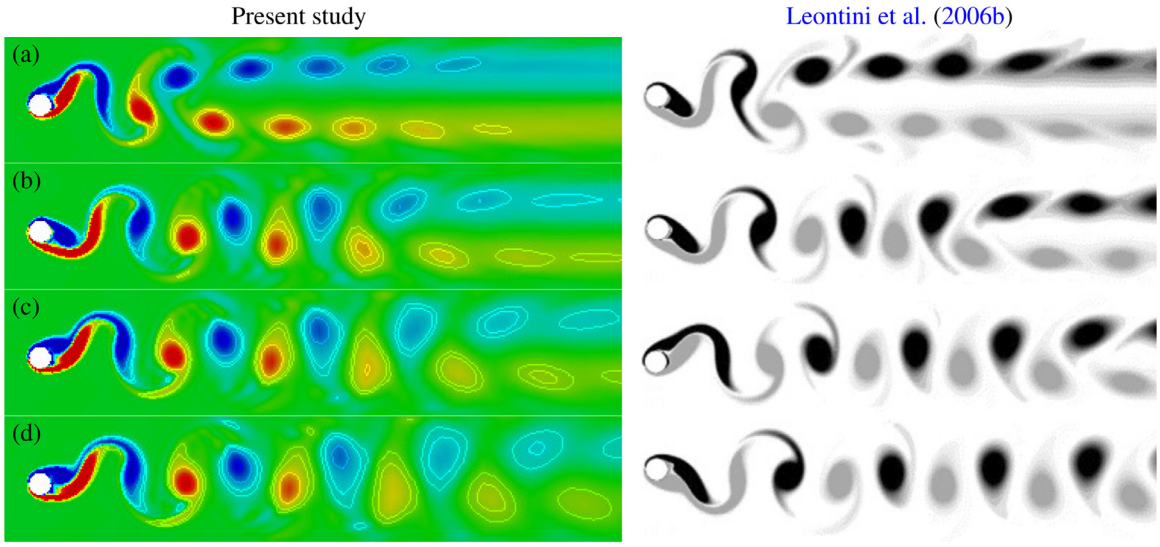


Fig. 5. Comparison of the vortical structures of the wake of vortex-induced vibration of a cylinder at $U^* = 5.0$, 5.2, 5.4, and 5.6 shown in (a), (b), (c), and (d), respectively, with the observations of [24]. Flow is from left to right. Formation of the double-row configuration is delayed, as the reduced velocity increased.

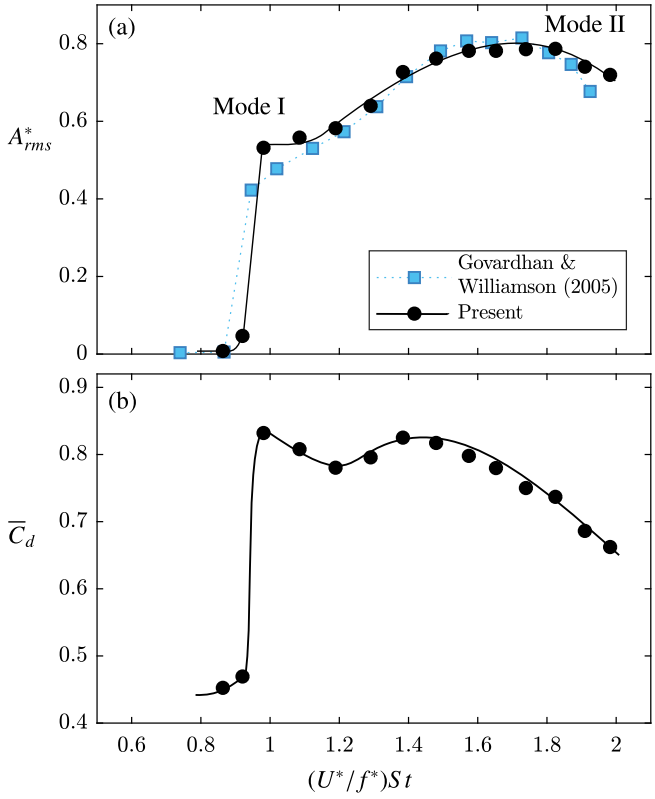


Fig. 6. Vortex-induced vibration response of a sphere ($m^* = 3$ and $Re = 2000$) as a function of the normalized velocity, $(U^*/f^*)St$: (a) comparison of the sphere rms amplitude with the results of [12] with a sphere of $m^* = 7$ and (b) the time-averaged drag coefficient, $\overline{C}_d$.

vortex loops behind the sphere. To analyse further the effects of the Reynolds number on the characteristics of the wake, a number of simulations were conducted by varying the Reynolds number from $Re = 300$ to 2000 and at the reduced velocity $U^* = 9$.

Fig. 7 compares the wake structures visualised with iso-surfaces of the Q criterion, at $Re = 300$, 800, 1200, and 2000. At $Re = 300$, two-streets of regular hairpin loops form the wake. Only large scale loops were visible, and each loop was accompanying with a tail. As the hairpin loops move downstream, they gradually con-

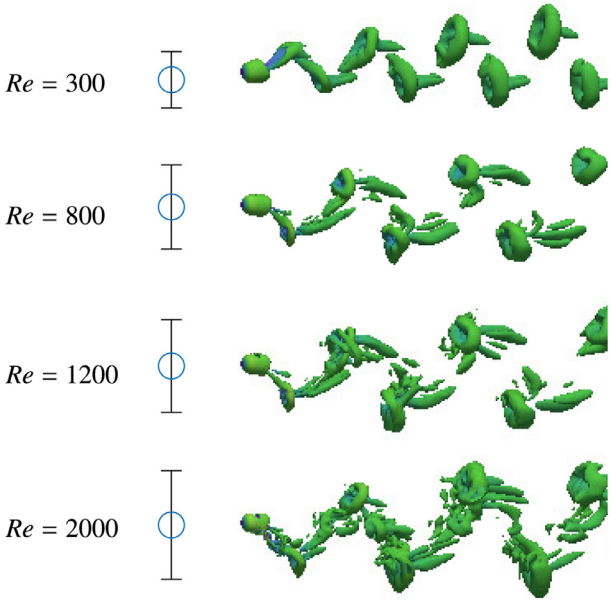
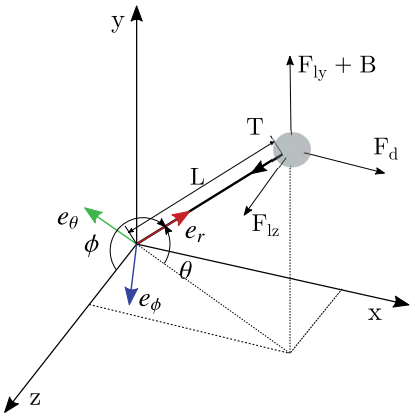


Fig. 7. Effect of the Reynolds number on the wake behind the sphere when it has synchronized vibrations at $U^* = 9$. Flow is from left to right and the wakes were visualised with iso-surfaces at $Q = 0.003$. The second column shows the peak to peak vibration amplitude for each case.

verted into rings. The wake was symmetric through the plane made by the direction of sphere vibration and the streamwise direction. This clear wake was modified in several ways, as the Reynolds number increased. First, the distance between the two vortex streets increased with increasing Re, indicating an increment in the vibration amplitude. Consistently, the sphere vibration amplitude gradually increased from $A^*_{rms} = 0.3$ to 0.8, as Re increased from 300 to 2000. Second, the hairpin loops converted into rings more quickly, with increasing Reynolds number. Third, as the Reynolds number increased, the large scale vortex loops in the wake were slightly distorted; in addition, small scale structures were visible in the wake. This was more prominent at $Re = 2000$, as expected at this higher Reynolds number. Finally, the vortex loops in the wake were slightly twisted about the streamwise directions, for $Re \geq 800$ case.



x - freestream direction
θ - tether angle from xz plan
ϕ - tether angle from z direction
F_d - drag force
F_{ly} - lift force in the y direction
F_{lz} - lift force in the z direction
B - buoyancy force
T - tension of the tether
L - tether length

Fig. 8. Schematic of the tethered sphere. Two coordinate systems were used to model the system; Cartesian coordinate, $<i, j, k>$, and spherical coordinates, $<e_r, e_\theta, e_\phi>$.

4. FIV solver of a tethered body

As the dynamic nature of the solid body attached to the end of a tether is different from an elastically mounted body, another FSI solver was developed to study the flow-induced vibration of a tethered body. The set-up is simply a tethered body in a uniform flow field, as the schematic shown in Fig. 8. The tether was assumed to be massless, rigid and inextensible. The last two assumptions restrict the motion of the solid body to lie in a spherical surface whose radius is the tether length. As the tether is assumed to be rigid, the orientation of the solid body stays fixed. Moreover, with this holonomic constraint, the number of equations required to describe the sphere dynamics reduces to two, even though the sphere has three degrees of freedom. Moreover, the tethered sphere undergoes pure rotation around the base point of the tether. Therefore, a 3D Rotation Group SO(3) can also be used to obtain the equations of motion of the sphere, as used by [33]. However, for simplicity, Newtonian Mechanics principles are used here, as described below.

4.1. Problem formulation

A spherical coordinate system, $S(\boldsymbol{e_r}, \boldsymbol{e_\theta}, \boldsymbol{e_\phi})$, was employed to obtain the dynamic equation of motion of the solid body, as shown in Fig. 8. However, fluid equations were derived with a Cartesian coordinate system $C(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$, attached to the centre of mass of the solid body. The unit vectors $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ are aligned with x, y, and z directions, respectively. The mapping between these two coordinate systems is bijective and can be elaborated with parameters, $\theta \in [0, 2\pi)$ and $\phi \in [0, \pi]$ as:

$$M : \begin{pmatrix} \boldsymbol{e_r} \\ \boldsymbol{e_\theta} \\ \boldsymbol{e_\phi} \end{pmatrix} = \begin{pmatrix} \cos\theta\sin\phi & \sin\theta\sin\phi & \cos\phi \\ -\sin\theta & \cos\theta & 0 \\ \cos\theta\cos\phi & \sin\theta\cos\phi & -\sin\phi \end{pmatrix} \begin{pmatrix} \boldsymbol{i} \\ \boldsymbol{j} \\ \boldsymbol{k} \end{pmatrix}, \tag{27}$$

$$M^{-1} : \begin{pmatrix} \boldsymbol{i} \\ \boldsymbol{j} \\ \boldsymbol{k} \end{pmatrix} = \begin{pmatrix} \cos\theta\sin\phi & -\sin\theta & \cos\theta\cos\phi \\ \sin\theta\sin\phi & \cos\theta & \sin\theta\cos\phi \\ \cos\phi & 0 & -\sin\phi \end{pmatrix} \begin{pmatrix} \boldsymbol{e_r} \\ \boldsymbol{e_\theta} \\ \boldsymbol{e_\phi} \end{pmatrix}, \tag{28}$$

where θ is the angle of tether to the xz plane and ϕ is the angle of the tether to the z direction. Let $\boldsymbol{x_s}$, $\dot{\boldsymbol{x}}_{\boldsymbol{s}}$, and $\ddot{\boldsymbol{x}}_{\boldsymbol{s}}$ represent the position, velocity, and acceleration of the tethered body in spherical coordinates. Then, the position of the body is simply,

$$\boldsymbol{x_s} = L\,\boldsymbol{e_r},$$

where L is the tether length. Then, $\dot{\boldsymbol{x}}_{\boldsymbol{s}}$ and $\ddot{\boldsymbol{x}}_{\boldsymbol{s}}$ can be obtained by differentiating the position and the velocity of the sphere w.r.t. time, as given in Eqs. (29) and (30), respectively.

$$\dot{\boldsymbol{x}}_{\boldsymbol{s}} = L\left(\dot\theta\sin\phi\,\boldsymbol{e_\theta} + \dot\phi\,\boldsymbol{e_\phi}\right). \tag{29}$$

$$\ddot{\boldsymbol{x}}_{\boldsymbol{s}} = L\left(-\left(\dot\theta^2\sin^2\phi + \dot\phi^2\right)\boldsymbol{e_r} + \left(\ddot\theta\sin\phi + 2\dot\theta\dot\phi\cos\phi\right)\boldsymbol{e_\theta} + \left(-\dot\theta^2\sin\phi\cos\phi + \ddot\phi\right)\boldsymbol{e_\phi}\right). \tag{30}$$

Three types of forces are acting on the tethered body; a tension force of the tether, T; a buoyancy force, B; and fluid forces F_d, F_{ly} and F_{lz}, which denote the components in the streamwise (x), lateral (y) and transverse (z) directions, respectively; see Fig. 8 for more information. The summation of the forces acting on the tethered body is $T\,\boldsymbol{e_r} + F_d\,\boldsymbol{i} + (F_{ly}+B)\,\boldsymbol{j} + F_{lz}\,\boldsymbol{k}$. Using the map M given in Eq. (27), the summation of forces can be converted in to spherical coordinates as:

$$\begin{aligned}\sum \boldsymbol{F} = &\left(F_d\cos\theta\sin\phi + (F_{ly}+B)\sin\theta\sin\phi + F_{lz}\cos\phi - T\right)\boldsymbol{e_r} \\ &- \left(F_d\sin\theta - (F_{ly}+B)\cos\theta\right)\boldsymbol{e_\theta} \\ &+ \left(F_d\cos\theta\cos\phi + (F_{ly}+B)\sin\theta\cos\phi - F_{lz}\sin\phi\right)\boldsymbol{e_\phi}.\end{aligned} \tag{31}$$

As the tether is assumed to be massless, the tethered body can be considered as an isolated mass. Now, with the information of acceleration of the body and forces acting on it, the dynamic equations of the solid body can be easily obtained by the angular momentum balance, *i.e.* $I\dot{\boldsymbol{\omega}} = \sum \boldsymbol{x_s} \times \boldsymbol{F}$, where I is the inertia of the solid body at the base of the tether and $I = m(D^2/10 + L^2)$ for a solid sphere and $\dot{\boldsymbol{\omega}} = \ddot{\boldsymbol{x}}_{\boldsymbol{s}}/L$ is the angular acceleration of the body. The component equations are

$$I\left(\ddot\theta\sin\phi + 2\dot\theta\dot\phi\cos\phi\right) = -L\left(F_d\sin\theta - (F_{ly}+B)\cos\theta\right) \tag{32}$$

and

$$I\left(\ddot\phi - \dot\theta^2\sin\phi\cos\phi\right) = L\left(F_d\cos\theta\cos\phi + (F_{ly}+B)\sin\theta\cos\phi - F_{lz}\sin\phi\right). \tag{33}$$

The above two dynamics equations can be converted into a matrix form by rearranging the terms as

$$\begin{pmatrix} \ddot\theta \\ \ddot\phi \\ 0 \end{pmatrix} = \frac{L}{I}\begin{pmatrix} -\sin\theta/\sin\phi & \cos\theta/\sin\phi & 0 \\ \cos\theta\cos\phi & \sin\theta\cos\phi & -\sin\phi \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} F_d \\ F_{ly}+B \\ F_{lz} \end{pmatrix} + \begin{pmatrix} -2\dot\theta\dot\phi\cot\phi \\ \dot\theta^2\sin\phi\cos\phi \\ 0 \end{pmatrix}. \tag{34}$$

At this point, it is important to note that there is a singularity associated with $\phi = 0$. However, it is not a problem for the current

simulations, since ϕ can never be 0 because the buoyancy force is much higher than the fluid forces and, therefore, the tether can never be aligned to the transverse direction (z direction).

Similar to the elastically mounted case, the Newtonian fluid is assumed incompressible and viscous, and modelled in a Cartesian coordinate system whose origin is the centre of mass of the solid body. As discussed in Section 3, the acceleration of the solid body should be included in the momentum Navier-Stokes equations, as the frame of reference is a non-inertial reference frame. Since the solid motion is modelled in a spherical coordinate system, the acceleration of the solid, $\ddot{\boldsymbol{x}}_{\boldsymbol{s}}$, given in Eq. (30) first needs to be converted into Cartesian coordinates. This can be easily done by using the mapping M given in Eq. (27) by

$$\ddot{\boldsymbol{x}}_{\boldsymbol{c}} = M\,\ddot{\boldsymbol{x}}_{\boldsymbol{s}}. \tag{35}$$

Finally, the coupled fluid-solid system can be described by the fluid equations given in (36) and (37) and the sphere motion equations given in (34) together with Eqs. (27), (30), and (35):

$$\frac{\partial \boldsymbol{u}}{\partial t} = -(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \nabla p + \nu\,\nabla^2\boldsymbol{u} - \ddot{\boldsymbol{x}}_{\boldsymbol{c}}, \tag{36}$$

$$\nabla\cdot\boldsymbol{u} = 0. \tag{37}$$

As a reminder, here p is the kinematic pressure, i.e., the static pressure divided by the density.

4.2. FSI solver for a tethered sphere

Similar to the vivIcoFoam solver for FIV problems of elastically mounted bodies, a fully coupled FSI solver was developed for FIV problems of tethered bodies, based on the dynamic equations presented in the previous subsection. This solver was named as *tetheredVivIcoFoam* and developed similar to the vivIcoFoam solver using a predictor-corrector iterative method. The only difference between this solver and the vivIcoFoam solver is the way it obtains the solid acceleration. The tetheredVivIcoFoam solver first solves the angular accelerations in spherical coordinates, and then converts them into the linear accelerations, while the vivIcoFoam solver directly solves for the linear accelerations. The iterative process of the tetheredVivIcoFoam solver at the $(n+1)$th time step can be elaborated as follows:

The predictor iteration

Initially, the angular accelerations of the sphere, $(\ddot{\theta}\ \ddot{\phi})^T$, are predicted explicitly using a third-order polynomial interpolation by

$$\begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n+1,\,p)} = 3\begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n)} - 3\begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n-1)} + \begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n-2)}. \tag{38}$$

Then, the angular velocities, $(\dot{\theta}\ \dot{\phi})^T$, and tether angles, $(\theta\ \phi)^T$, are estimated using a third-order Adams–Moulton method, by integrating the angular accelerations and angular velocities to obtain

$$\begin{pmatrix}\dot{\theta}\\ \dot{\phi}\end{pmatrix}^{(n+1,\,p)} = \begin{pmatrix}\dot{\theta}\\ \dot{\phi}\end{pmatrix}^{(n)} + \frac{\delta t}{12}\left(5\begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n+1,\,p)} + 8\begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n)} - \begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n-1)}\right) \tag{39}$$

and

$$\begin{pmatrix}\theta\\ \phi\end{pmatrix}^{(n+1,\,p)} = \begin{pmatrix}\theta\\ \phi\end{pmatrix}^{(n)} + \frac{\delta t}{12}\left(5\begin{pmatrix}\dot{\theta}\\ \dot{\phi}\end{pmatrix}^{(n+1,\,p)} + 8\begin{pmatrix}\dot{\theta}\\ \dot{\phi}\end{pmatrix}^{(n)} - \begin{pmatrix}\dot{\theta}\\ \dot{\phi}\end{pmatrix}^{(n-1)}\right), \tag{40}$$

respectively. Then, the acceleration of the solid, $\ddot{\boldsymbol{x}}_{\boldsymbol{s}}^{(n+1,\,p)}$, is obtained by Eq. (30) with the predicted angles, angular velocities and angular accelerations. After that, $\ddot{\boldsymbol{x}}_{\boldsymbol{s}}^{(n+1,\,p)}$ is converted into Cartesian coordinates, $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,p)}$, using the mapping given in Eq. (27). At the end of the predictor step, the Navier–Stokes equations given in Eqs. (36) and (37) are solved with the predicted $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,p)}$ and the forces exerted on the sphere, $(F_d\ F_{ly}\ F_{lz})^{(n+1,\,p)}$, are calculated.

A corrector iteration

Initially, angular accelerations of the sphere, $(\ddot{\theta}\ \ddot{\phi})^T$, are corrected by Eq. (34) with the values obtains for θ, ϕ, $\dot{\theta}$, $\dot{\phi}$, F_d, F_{ly} and F_{lz} at the predictor iteration (or at the last corrector iteration). Then, from the second corrector iteration onwards, the corrected angular accelerations are relaxed to improve the convergence characteristics by

$$\begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n+1,\,c_i)'} = \begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n+1,\,c_{i-1})} + \gamma\left(\begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n+1,\,c_i)} - \begin{pmatrix}\ddot{\theta}\\ \ddot{\phi}\end{pmatrix}^{(n+1,\,c_{i-1})}\right), \tag{41}$$

where γ is the relaxation parameter, and c_i represents the values calculated at the ith corrector iteration. The tetheredVivIcoFoam solver becomes unstable, especially for small mass ratio spheres and low Reynolds number flows in the absence of any relaxation. The convergence of the method can be improved by the choice of γ, depending on the parameter combination.

Once the angular accelerations are obtained and relaxed, the angular velocities and angles can be obtained by integrating the angular accelerations and the angular velocities, similar to the predictor step. Then, the linear acceleration of the sphere in Cartesian coordinates, $\ddot{\boldsymbol{x}}_{\boldsymbol{c}}^{(n+1,\,c_i)}$ is obtained and the Navier–Stokes equations are solved, similar to the predictor step. Finally, the fluid forces exerted on the solid, $(F_d\ F_{ly}\ F_{lz})^{(n+1,\,c_i)}$, are calculated to use in the next corrector iteration. Termination of the iterative process is similar to the vivIcoFoam solver.

Once the predictor-corrector iterative process is completed, the velocity boundary conditions of the inlet patches need to be adjusted according to the solid linear velocity, $\dot{\boldsymbol{x}}_{\boldsymbol{c}} = M\dot{\boldsymbol{x}}_{\boldsymbol{s}}$. This is because the fluid flow is modelled in the moving frame of reference which is attached to the centre of mass of the solid body.

4.3. Validation of the tetheredVivIcoFoam solver

To examine the validity of the newly built tetheredVivIcoFoam solver, a set of simulations were conducted at $Re = 2000$ using a tethered sphere of mass ratio $m^* = 0.8$ and tether length ratio $l^* = 9.5$ by varying the reduced velocity over the range, $U^* \in [3.5, 32]$. Fig. 9(a) shows a comparison of sphere response amplitude with the results of [18] at the same mass ratio and higher and varying Reynolds numbers. The current predictions reasonably agree with the observations of [18]. As shown in the figure, we observed mode I type response around $U^* \backsim 5$ and large amplitude mode II type response just after the mode I response. Both of these responses were highly periodic. Consistent with their observations, the sphere response amplitude transitioned smoothly from mode I to mode II with a local peak at the mode I. Moreover, the peak amplitudes observed at modes I and II were $\approx 0.55D$ and $\approx 0.9D$. However, the response curve we observed was slightly shifted to lower reduced velocities compared to their observations. This may be possibly due to the effect of Reynolds number. In their experimental study, the Reynolds number is varying with the reduced velocity while we fixed the Reynolds number at 2000. If the amplitude response was to be plotted against the normalized velocity, instead of the reduced velocity, the response curves might match better, as seen for the elastically mounted sphere (shown in Fig. 6). However, it is not quite possible to generate this plot, as the f^* and St values of jauvtis et al.'s study are not given.

After the peak of mode II, the current predictions indicate a decreasing trend of response amplitude. At higher reduced velocities ($U^* \gtrsim 15$), the sphere response was aperiodic at this small mass ratio and showed mode IV type intermittent bursts of vibrations (see the time histories shown in Fig. 9 for 10 oscillation cycles at each mode).

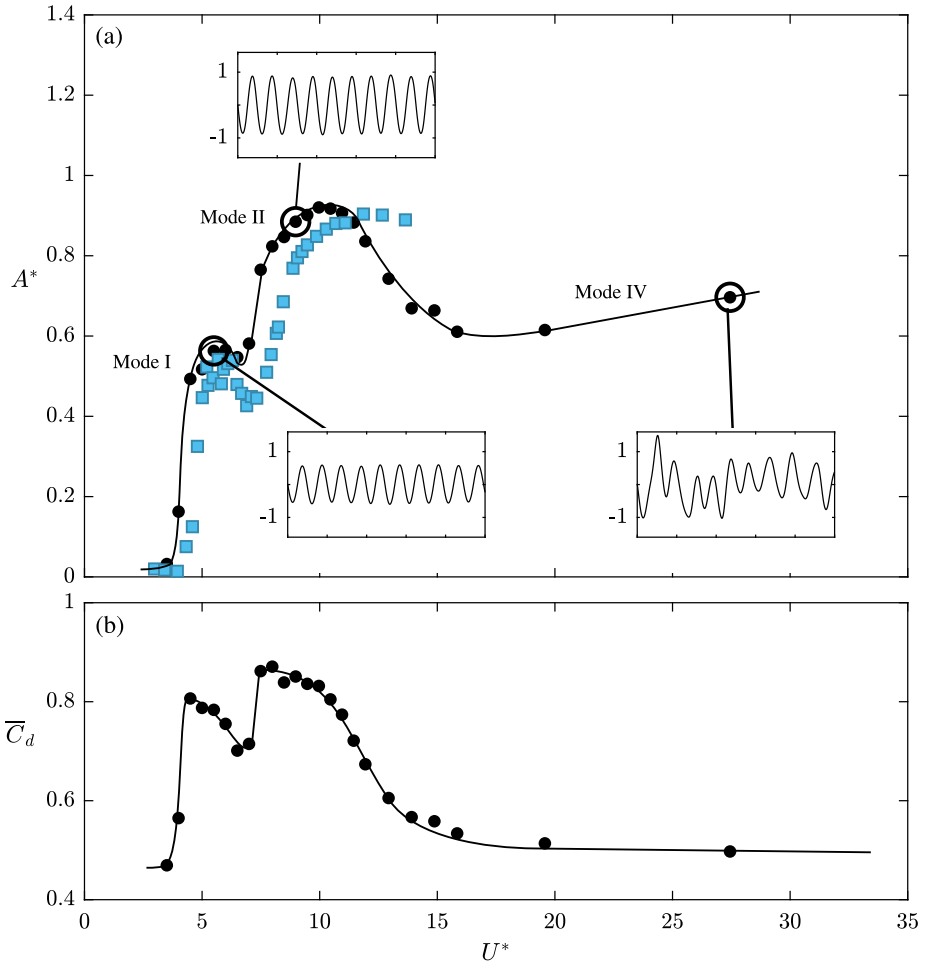


Fig. 9. Flow-induced vibration response of a tethered sphere at $Re = 2000$: (a) comparison of the amplitude response with the experimental results of [18] at higher and varying Reynolds numbers and (b) time-averaged drag coefficient. The mass ratio of the sphere and the tether length ratio are 0.8 and 9.5, respectively.

Fig. 10. Wake structures observed at mode I and II with a tethered sphere at $Re = 2000$. Flow is from left to right.

Fig. 9 (b) plots the time-averaged drag coefficient, $\overline{C}_d$, against U^*. Two peaks were found in the drag coefficient plot, which correspond to mode I and II vibrations. The maximum increment of $\overline{C}_d$ was roughly 90% of the value calculated for a stationary sphere, consistent with the results of an elastically mounted sphere discussed in Section 3.5.2. A comparison of the wake structures visualised using the isosurfaces of the Q criterion ($Q = 0.006$) in modes I and II is given in Fig. 10. In the wake of mode I, two large-scale loops were shed per oscillation cycle, along with small scale loops. As the mode transitioned from mode I to mode II, multiple loops, which were interconnected, were shed per oscillation cycle.

5. Conclusions

Flow-induced vibration of a bluff body is a vibrational phenomenon of a structure induced by forcing from the surrounding fluid flow. Examining the FIV of a bluff body is a complicated process, as it is a fluid-structure interaction (FSI) problem. Finding an analytical solution to a practical FIV problem is nearly impossible. As a result, it should be investigated with physical experiments or computer simulations. Compared to experimental studies, a fewer number of numerical studies have been reported on FIV of a bluff body. This is because developing an efficient and accurate numerical methodology is challenging. Conforming grid methods like arbitrary Lagrangian-Eulerian methods and non-conforming grid methods like immersed boundary methods enable solving FSI problems. However, these FSI algorithms are less suitable for the flow-induced vibration problems, as they are highly time consuming or produce low accuracy predictions, especially for cases involving large body displacements. In this article, we present two fully-coupled FSI solvers to efficiently and accurately investigate FIV problems of bluff bodies mounted in two different ways: mounted with elastic supports and mounted with a tether.

These FSI solvers were developed in the OpenFOAM environment. OpenFOAM is a widely used CFD package based on the finite-volume method. OpenFOAM facilitates the solution for a wide variety of flow problems with pre-built flow solvers and libraries. It also allows a user to develop their own solvers and libraries depending on their requirements. The FSI solvers presented in this paper were initially developed based on the pre-

build *icoFoam* solver for incompressible and laminar flows. However, these solvers can be easily adapted for turbulent flows, as OpenFOAM already has different prebuilt flow solvers that allow different choices of RANS and LES turbulence models.

In each of the two FSI solvers presented, the fluid flow was modelled in a reference frame attached to the centre of mass of the solid body, so that a non-deforming grid can be employed. The fluid equations were coupled with the dynamic equations of the solid body through the acceleration of the frame and the fluid forces acting on the solid body. For the case of an elastically mounted body, the motion of the solid was modelled as a spring-mass-damper system; while for the case of a tethered body, the dynamic equations were obtained using angular momentum balance based in a spherical coordinate system. In each solver, a predictor-corrector iterative procedure was used to solve the coupled equations over each time step. This iterative process is third-order accurate, and typically it required two to three corrector steps for convergence. As a non-deforming grid is used and the iterative process converges quickly, this method is considerably more efficient than arbitrary Lagrangian-Eulerian methods and immersed boundary methods for the class of problems only involving the motion of a single body. Moreover, it produces an accurate answer since the strong Dirichlet boundary condition is applied on the solid boundary, in contrast to immersed boundary methods which are required to use a weak formulation of the Dirichlet boundary condition on the fluid-solid interface.

Although the predictor-corrector iterative method is third-order accurate the entire FSI algorithm is second-order accurate, as the PISO algorithm used in the flow solver is second-order accurate. It should be noted that this method can only be used for flow-induced vibration problems of single, rigid bodies, since the fluid flow is been modelled in a frame attached to the centre of mass of the body. Each of the solvers was validated by comparing predictions obtained against previously published results.

Acknowledgments

The support from Australian Research Council Discovery Grants DP150102879 and DP170100275, and computing time from the National Computational Infrastructure (NCI) and the Pawsey Supercomputing Centre through merit grants N67 and D71 are gratefully acknowledged.

Appendix A. Additional files required for a FIV simulation

system/solidMotionaData

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      solidMotionData;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //


massRatio           3.8197; // Mass ratio, m*
zeta                0.0;    // Damping ratio
reducedVelocity     3.5;    // Reduced velocity, U* w/o added mass
D                   1;      // Diameter of the sphere
CS                  3;      // Cylinder/Sphere, 2- cylinder and 3- sphere
L                   1;      // If a cylinder, Length of the cylinder
Uinf                1;      // Upstream velocity
rho                 1000;   // Fluid density
dispFreedom         1;      // 1- restrict motion on the lift direction.
maxNoOfIterations   50;     // Maximun number of iterations
stopcriterion       0.001;  // Tolarence to stop the iterative process
underRelaxationPara 1;      // Under relaxation, 1- no under relaxation
osillationMode      3;      // 1-elastically mounted, 2-force diven, 3- no
osillation
maximumDispalcement 0.4;    // Maximum dispacement, if forced oscillation
osillationFrequency 0.195;  // Osillation frequency, if forced oscillation
IntegrationMethod   1;      // Integration method used. Defaul- Adams Moulton, 2-
Backward Eular, 3- Trapizoidal rule.

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

system/lastMotionaData

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      lastMotionData;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

s0now           ( 0 0.113035 0 );

s0last          ( 0 0.1131 0 );

s1now           ( 0 -0.0129737 0 );

s1last          ( 0 -0.012904 0 );

s2now           ( 0 -0.0138929 0 );

s2last          ( 0 -0.0139706 0 );

s2last2         ( 0 -0.0140492 0 );


// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

constant/boundaryToUpdate

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      boundaryToUpdate;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

    type        3; // 0- no, 1- inlet only, 2- 2D (inlet, top, bottom), 3- 3D
(first 5 patches)

    patch1      INFLOW;
    patch2      TOP;
    patch3      BOTTOM;
    patch4      FRONT;
    patch5      BACK;
    patch6      OUTFLOW;
    patch7      SPHERE; // The patch7 should be the solid (cylinder/sphere)


// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compfluid.2019.104340.

References

[1] Bearman PW. Vortex shedding from oscillating bluff bodies. Annu Rev Fluid Mech 1984;16(1):195–222.
[2] Bearman PW. Circular cylinder wakes and vortex-induced vibrations. J Fluids Struct 2011;27(5–6):648–58.
[3] Blackburn H, Henderson R. Lock-in behavior in simulated vortex-induced vibration. Exp Thermal Fluid Sci 1996;12(2):184–9.

[4] Blackburn HM, Karniadakis GE, et al. Two-and three-dimensional simulations of vortex-induced vibration or a circular cylinder. In: The third international offshore and polar engineering conference. International Society of Offshore and Polar Engineers; 1993.

[5] Borazjani I. Fluid–structure interaction, immersed boundary-finite element method simulations of bio-prosthetic heart valves. Comput Methods Appl MechEng 2013;257:103–16.

[6] Constantinescu GS, Squires KD. Les and des investigations of turbulent flow over a sphere at Re= 10,000. Flow Turbul Combust 2003;70(1–4):267–98.

[7] Dillon R, Fauci L, Fogelson A, Gaver III D. Modeling biofilm processes using the immersed boundary method. J Comput Phys 1996;129(1):57–73.

[8] Ding L, Bernitsas MM, Kim ES. 2-d URANS vs. experiments of flow induced motions of two circular cylinders in tandem with passive turbulence control for 30,000<Re< 105,000. Ocean Eng 2013;72:429–40.

[9] Ern P, Risso F, Fabre D, Magnaudet J. Wake-induced oscillatory paths of bodies freely rising or falling in fluids. Annu Rev Fluid Mech 2012;44:97–121.

[10] Facchinetti ML, De Langre E, Biolley F. Coupling of structure and wake oscillators in vortex-induced vibrations. J Fluids Struct 2004;19(2):123–40.

[11] Giacobello M, Ooi A, Balachandar S. Wake structure of a transversely rotating sphere at moderate Reynolds numbers. J Fluid Mech 2009;621:103–30.

[12] Govardhan RN, Williamson CHK. Vortex-induced vibrations of a sphere. J Fluid Mech 2005;531:11–47.

[13] Griffith BE, Luo X, McQueen DM, Peskin CS. Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method. Int J Appl Mech 2009;1(01):137–77.

[14] Habchi C, Russeil S, Bougeard D, Harion J, Lemenand T, Ghanem A, Della Valle D, Peerhossaini H. Partitioned solver for strongly coupled fluid–structure interaction. Comput Fluids 2013;71:306–19.

[15] Hou G, Wang J, Layton A. Numerical methods for fluid-structure interaction a review. Commun Comput Phys 2012;12(2):337–77.

[16] Hughes TJR, Liu WK, Zimmermann TK. Lagrangian-eulerian finite element formulation for incompressible viscous flows. Comput Methods Appl MechEng 1981;29(3):329–49.

[17] Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. J Comput Phys 1986;62(1):40–65.

[18] Jauvtis N, Govardhan R, Williamson CHK. Multiple modes of vortex-induced vibration of a sphere. J Fluids Struct 2001;15(3):555–63.

[19] Johnson TA, Patel VC. Flow past a sphere up to a Reynolds number of 300. J Fluid Mech 1999;378:19–70.

[20] Kamensky D, Hsu MC, Evans JA, Schillinger D, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR. An immersogeometric variational framework for fluid–structure interaction: application to bioprosthetic heart valves. Comput Methods Appl MechEng 2015;284:1005–53.

[21] Kim D, Choi H. Immersed boundary method for flow around an arbitrarily moving body. J Comput Phys 2006;212(2):662–80.

[22] Lee H, Hourigan K, Thompson MC. Vortex-induced vibration of a neutrally buoyant tethered sphere. J Fluid Mech 2013;719:97–128.

[23] Leontini JS, Stewart BE, Thompson MC, Hourigan K. Predicting vortex-induced vibration from driven oscillation results. Appl Math Model 2006;30(10):1096–102. Special issue of the 12th Biennial Computational Techniques and Applications Conference and Workshops (CTAC-2004) held at The University of Melbourne, Australia, from 27th September to 1st October 2004.

[24] Leontini JS, Thompson MC, Hourigan K. The beginning of branching behaviour of vortex-induced vibration during two-dimensional flow. J Fluids Struct 2006;22(6):857–64.

[25] Mittal R, Iaccarino G. Immersed boundary methods. Annu Rev Fluid Mech 2005;37:239–61.

[26] Morsi SA, Alexander AJ. An investigation of particle trajectories in two-phase flow systems. J Fluid Mech 1972;55:193–208.

[27] Ota K, Suzuki K, Inamuro T. Lift generation by a two-dimensional symmetric flapping wing: immersed boundary-lattice Boltzmann simulations. Fluid Dyn Res 2012;44(4):045504.

[28] Pantazopoulos MS. Vortex-induced vibration parameters: critical review. Technical Report. American Society of Mechanical Engineers, New York, NY (United States); 1994.

[29] Parkinson G. Phenomena and modelling of flow-induced vibrations of bluff bodies. Progr Aerosp Sci 1989;26(2):169–224.

[30] Peskin CS. Flow patterns around heart valves: a numerical method. J Comput Phys 1972;10(2):252–71.

[31] Poon EKW, Ooi ASH, Giacobello M, Iaccarino G, Chung D. Flow past a transversely rotating sphere at Reynolds numbers above the laminar regime. J Fluid Mech 2014;759:751–81.

[32] Prasanth TK, Mittal S. Vortex-induced vibrations of a circular cylinder at low Reynolds numbers. J Fluid Mech 2008;594:463–91.

[33] Rajamuni MM, Aulisa E, Ghosh BK. Optimal control problems in binocular vision. IFAC Proc Volumes 2014;47(3):5283–9. 19th IFAC World Congress.

[34] Rajamuni MM, Thompson MC, Hourigan K. Transverse flow-induced vibrations of a sphere. J Fluid Mech 2018;837:931–66.

[35] Rajamuni MM, Thompson MC, Hourigan K. Vortex dynamics and vibration modes of a tethered sphere. J Fluid Mech 2018. Under review.

[36] Rajamuni MM, Thompson MC, Hourigan K. Vortex-induced vibration of a transversely rotating sphere. J Fluid Mech 2018;847:786–820.

[37] Rajamuni MM, Thompson MC, Hourigan KH. Vortex-induced vibration of elastically-mounted spheres: a comparison of the response of three degrees of freedom and one degree of freedom systems. J Fluids Struct 2019.

[38] Sakamoto H, Haniu H. A study on votex shedding from sphere in a uniform flow. Trans ASME 1990;112:386–92.

[39] Sarpkaya T. A critical review of the intrinsic nature of vortex-induced vibrations. J Fluids Struct 2004;19(4):389–447.

[40] Silva ALFLE, Silveira-Neto A, Damasceno JJR. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. J Comput Phys 2003;189(2):351–70.

[41] Soti AK, Bhardwaj R, Sheridan J. Flow-induced deformation of a flexible thin structure as manifestation of heat transfer enhancement. Int J Heat Mass Transfer 2015;84:1070–81.

[42] Tezduyar TE, Behr M, Mittal S, Liou J. A new strategy for finite element computations involving moving boundaries and interfacesthe deforming-spatial–domain/space-time procedure: ii. computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. Comput Methods Appl MechEng 1992;94(3):353–71.

[43] The-OpenFOAM-Foundation, Openfoam v6 user guide, 2018, (https://cfd.direct/openfoam/user-guide).

[44] Tomboulides AG, Orszag SA. Numerical investigation of transitional and weak turbulent flow past a sphere. J Fluid Mech 2000;416:45–73.

[45] Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. J Comput Phys 2005;209(2):448–76.

[46] Williamson CHK, Govardhan R. Dynamics and forcing of a tethered sphere in a fluid flow. J Fluids Struct 1997;11(3):293–305. doi:10.1006/jfls.1996.0078.

[47] Williamson CHK, Govardhan R. Vortex-induced vibrations. Annu Rev Fluid Mech 2004;36:413–55.

[48] Williamson CHK, Govardhan R. A brief review of recent results in vortex-induced vibrations. J Wind Eng IndAerodyn 2008;96(6):713–35.

[49] Wu W, Bernitsas MM, Maki K. RANS simulation versus experiments of flow induced motion of circular cylinder with passive turbulence control at 35,000<Re<130,000. J Offshore Mech Arctic Eng 2014;136(4):041802.

[50] Wu X, Ge F, Hong Y. A review of recent studies on vortex-induced vibrations of long slender cylinders. J Fluids Struct 2012;28:292–308.

[51] Xu WH, Wu YX, Zhong JX, Zeng JH, Yu JX. A new wake oscillator model for predicting vortex induced vibration of a circular cylinder. J Hydrodyn 2010;22(3):381–6.