



An improved penalty immersed boundary method for fluid–flexible body interaction

Wei-Xi Huang^{a,b}, Cheong Bong Chang^a, Hyung Jin Sung^{a,*}

^a Department of Mechanical Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea

^b Department of Engineering Mechanics, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:

Received 7 September 2010

Received in revised form 29 January 2011

Accepted 14 March 2011

Available online 21 March 2011

Keywords:

Penalty immersed boundary method

Fluid–structure interaction

Flexible body

Elastic circular disk

Spherical capsule

ABSTRACT

An improved penalty immersed boundary (pIB) method has been proposed for simulation of fluid–flexible body interaction problems. In the proposed method, the fluid motion is defined on the Eulerian domain, while the solid motion is described by the Lagrangian variables. To account for the interaction, the flexible body is assumed to be composed of two parts: massive material points and massless material points, which are assumed to be linked closely by a stiff spring with damping. The massive material points are subjected to the elastic force of solid deformation but do not interact with the fluid directly, while the massless material points interact with the fluid by moving with the local fluid velocity. The flow solver and the solid solver are coupled in this framework and are developed separately by different methods. The fractional step method is adopted to solve the incompressible fluid motion on a staggered Cartesian grid, while the finite element method is developed to simulate the solid motion using an unstructured triangular mesh. The interaction force is just the restoring force of the stiff spring with damping, and is spread from the Lagrangian coordinates to the Eulerian grids by a smoothed approximation of the Dirac delta function. In the numerical simulations, we first validate the solid solver by using a vibrating circular ring in vacuum, and a second-order spatial accuracy is observed. Then both two- and three-dimensional simulations of fluid–flexible body interaction are carried out, including a circular disk in a linear shear flow, an elastic circular disk moving through a constricted channel, a spherical capsule in a linear shear flow, and a windsock in a uniform flow. The spatial accuracy is shown to be between first-order and second-order for both the fluid velocities and the solid positions. Comparisons between the numerical results and the theoretical solutions are also presented.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Since the 70s of the last century, the immersed boundary (IB) method has gained more and more popularity due to its ability in simulating complex flows and has been applied to various engineering flow problems [1]. The basic idea of the IB method is to avoid the necessity of adopting body-fitted meshes by adding a momentum forcing in the equations of motion to mimic complex boundaries. In this way, a flow over fixed or moving boundaries can be simulated on fixed Eulerian meshes, thus significantly simplifying the solution process. More attractively, including the momentum forcing requires little modification of the original numerical scheme for flow computation, thus the IB method can be implemented conveniently based on the existing flow solver. Moreover, the computation efficiency can also be improved significantly as compared with the body-fitted mesh method for simulating complex flows.

* Corresponding author. Tel.: +82 42 350 3027; fax: +82 42 350 5027.

E-mail address: hjsung@kaist.ac.kr (H.J. Sung).

Recently, various versions of the IB method have been proposed according to the boundary configurations of fluid motions. To enforce the no-slip condition on fixed boundaries, the momentum forcing is formulated on the discretized grids directly or is first calculated on the Lagrangian boundary points and is then transformed to the Eulerian coordinates. The former is referred to as the discrete forcing approach, where specified schemes were introduced for velocity interpolation or extrapolation on the forcing points [2–4]. The latter is referred to as the continuous forcing approach, where a convenient way to bring the fluid to rest on the boundaries is to adopt a feedback law and transformation between the Lagrangian and Eulerian coordinates is realized by a smoothed approximation of the Dirac delta functions or similar methods [5–7]. The momentum forcing was also interpreted as a Lagrange multiplier and was determined implicitly by using a projection method to solve the whole system [8]. Implicit schemes can also be found in [9,10] where a linear system was formed from the overall numerical schemes to solve the momentum forcing. For moving boundary problems, more elaborate treatments were developed in the discrete forcing approach for dealing with the forcing point which is in the solid domain at the current time step but is moved into the fluid domain at the next time step, such as the field-extension method [11] and the fresh-cell method [12]. Another methodology for single arbitrarily moving rigid body in unbounded flows is to fix the reference frame with the moving body and thus to solve the governing equations in the non-inertial frame [13]. On the other hand, in the continuous forcing approach, there is no need of special treatment for dealing with the moving boundary problems due to the fact that the smoothed delta function has a smooth effect on the numerical solutions [8,9,14,15]. However, the boundary is smeared in a width of several grids which limits the continuous forcing approach to low and moderate Reynolds number flows. Moreover, for the two-way coupling fluid–structure interaction problems, the structure motion is determined by its dynamic equations under certain coupling schemes [13,14,16,17].

Another prevalent way is to consider the elastic body motion in fluid, where the fluid–solid interface is subjected to not only the fluid shear stress and pressure, but also the elastic force due to the solid deformation. Thus, the field equations should be solved for the solid motion as well as the fluid motion, which makes the whole system more complicated than the fluid–rigid body interaction problems. The original IB method proposed by Peskin [18] was designed toward this end. Starting from the continuum mechanics point of view, Peskin [19] wrote the momentum equations of an incompressible elastic material in the same form as those of the incompressible fluid, i.e. the Navier–Stokes (N–S) equations. The elastic force was calculated at the Lagrangian points of solid, which move with the local fluid velocity, and was then spread to the nearby Eulerian grids by the smoothed delta function as the additional momentum forcing in the N–S equations. This is especially efficient for dealing with a massless (or neutrally buoyant) elastic body immersed in fluid. In order to simulate a massive elastic body moving in fluid, Zhu and Peskin [20] spread the solid mass to the near Eulerian grid points in the same manner as the elastic force. The multigrid method was used to solve the discretized N–S equations with a variable density coefficient, instead of the efficient FFT method. Furthermore, Kim and Peskin [21] proposed an improved method referred to as the penalty IB (pIB) method, where the solid mass is added by introducing a virtual boundary with the needed mass that does not directly interact with the fluid but rather is connected by a stiff spring to its twin massless boundary that moves with the local fluid velocity. In this way, they were able to solve the N–S equations with a constant density coefficient, and thus retain the use of the FFT method. Mori and Peskin [22] interpreted the solid mass as the d'Alembert force and proposed an implicit scheme for the momentum forcing, including the elastic force and the d'Alembert force. However, iterations are needed to solve the whole system at each time step.

For most biofluid problems, such as blood flows in the heart and blood cells moving in a vessel, solution of the solid motion poses another challenge and usually requires an independent solid solver. Luo et al. [23] developed a sharp-interface IB method for solving the motion equations of a linear viscoelastic solid on the Cartesian grid, which was coupled with a previously developed IB method for the flow motion in an iterative manner. The method was then applied to three-dimensional simulations of the human phonation phenomena. Gilmanov and Acharya [24] adopted the material point method to calculate the solid motion on a background Cartesian grid, while the fluid motion with solid boundaries was solved in a curvilinear coordinate by the hybrid immersed boundary method [25]. Even for a uniform set of equations used when both the fluid and the solid are incompressible as in Peskin [19], representation of the complex solid geometries usually required the unstructured grid and the elastic force due to the solid deformation was evaluated by the finite element method [26–28]. To couple with the fluid motion, Zhao et al. [26] proposed a force-projection method to transfer the body and surface forces, and Le and coworkers [27,28] developed an implicit forcing scheme where a Jacobian-free approach was adopted for the advancement of solid positions.

In the present study, our objective is to establish an efficient numerical algorithm for simulating fluid–flexible body interaction. For general consideration, the fluid solver and the solid solver are developed separately. The incompressible fluid motion is defined on the Eulerian domain and is solved by the fractional step method on a staggered Cartesian grid system. The solid motion is described by the Lagrangian variables and is solved by the finite element method. To couple the fluid and solid motions, we adopt the penalty IB method by virtually separating the solid body into the massive material and the massless material (referred to as the twin boundaries [21]). Remember that in Kim and Peskin [21] the elastic force was exerted on the massless boundary which interacts with the fluid so that a single set of equations were solved for the whole system. Here we simply move the elastic force to the massive boundary. In this way, we have the fluid motion equations and the solid motion equations to be solved independently, with the momentum forcing just the restoring force of a stiff spring with damping which links the twin material points. More importantly, the elastic force in the solid motion equations can be treated implicitly, which makes the present method applicable to more complicated solid motions. In the next section the prob-

lem formulation is derived, and in Section 3 the numerical method is introduced. Then in Section 4 we present several numerical examples, including both two- and three-dimensional cases. Finally, conclusions are drawn in Section 5.

2. Problem formulation

2.1. Governing equations

The fluid motion is governed by the incompressible N–S equations and the continuity equation, i.e.

$$\rho_0 \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where ρ_0 is the fluid density, \mathbf{u} is the velocity vector, p is the pressure, μ is the dynamic viscosity, and \mathbf{f} is the momentum forcing applied to enforce the no-slip condition along the IB. The formulation of \mathbf{f} will be given later. Here the fluid motion is defined in the Eulerian domain using a Cartesian coordinate system $Oxyz$ (Fig. 1). The Cartesian axes are denoted by x , y and z , or equivalently x_1 , x_2 and x_3 , and the corresponding velocity components are denoted by u , v and w , or equivalently u_1 , u_2 and u_3 .

In the present study, we consider a two-dimensional (2D) flexible body immersed in 2D fluid and a 2D membrane in three-dimensional (3D) fluid. Unlike the Eulerian fluid motion, the solid motion is described by the Lagrangian variables defined on a moving curvilinear coordinate system (s_1, s_2) , as shown in Fig. 1. The initial undeformed configuration is regarded as the reference frame of solid motion, i.e. $\mathbf{X}^0 = \mathbf{X}(s_1, s_2, t = 0)$. At the subsequent time, the position of the flexible body is denoted by $\mathbf{X}(s_1, s_2, t)$, which obeys

$$\rho \frac{\partial^2 \mathbf{X}}{\partial t^2} + \lambda \frac{\partial \mathbf{X}}{\partial t} = \frac{\partial}{\partial s_i} \left(S_{ij} \frac{\partial \mathbf{X}}{\partial s_j} \right) - \mathbf{F}, \quad (3)$$

where ρ denotes the density difference between solid and fluid, λ denotes the damping coefficient, S_{ij} denotes the second Piola–Kirchhoff stress tensor, and \mathbf{F} denotes the Lagrangian forcing exerted on the flexible body by the fluid. Here the summation convention is applied for i and j , which take the values 1 and 2. Note that \mathbf{F} is the Lagrangian counterpart of \mathbf{f} in Eq. (1), and a minus sign in front of \mathbf{F} is on account of the mutual forces of action and reaction between fluid and solid.

To take into account large deformation of the flexible body, a Green strain tensor is introduced [29]

$$D_{ij} = \frac{1}{2} \left(\frac{\partial \mathbf{X}}{\partial s_i} \cdot \frac{\partial \mathbf{X}}{\partial s_j} - \frac{\partial \mathbf{X}^0}{\partial s_i} \cdot \frac{\partial \mathbf{X}^0}{\partial s_j} \right) \quad (i, j = 1, 2), \quad (4)$$

which can also be written in a matrix form

$$\mathbf{D} = \frac{1}{2} (\mathbf{G}^T \mathbf{G} - \mathbf{G}^{0T} \mathbf{G}^0), \quad (5)$$

where $\mathbf{G} = \partial \mathbf{X} / \partial \mathbf{s}$, or equivalently $G_{ij} = \partial X_i / \partial s_j$, denotes the deformation gradient tensor, and $\mathbf{G}^0 = \partial \mathbf{X}^0 / \partial \mathbf{s}$ is its initial status. It should be pointed out the classical formulation for the deformation gradient tensor rather than \mathbf{G} is $\tilde{\mathbf{F}} = \partial \mathbf{X} / \partial \mathbf{X}^0$. In general, the base-vectors can be selected independently for \mathbf{X}^0 and \mathbf{X} , so \mathbf{G} is actually a special case of $\tilde{\mathbf{F}}$. But here we use the same rectangular Cartesian axes as the base vectors for both \mathbf{X}^0 and \mathbf{X} , so $\tilde{\mathbf{F}}$ is related to \mathbf{G} by $\mathbf{G} = \tilde{\mathbf{F}} \mathbf{G}^0$. Therefore, the current Green strain tensor \mathbf{D} is related to that based on the rectangular Cartesian coordinates by $\mathbf{D} = \mathbf{G}^{0T} \mathbf{E} \mathbf{G}^0$, where $\mathbf{E} = \frac{1}{2} (\tilde{\mathbf{F}}^T \tilde{\mathbf{F}} - \mathbf{I})$. Since \mathbf{G}^0 is only dependent on the initial configuration, \mathbf{D} is invariant with the base-vectors for \mathbf{X} and is thus a Lagrangian tensor.

Given a strain energy function W , the second Piola–Kirchhoff stress tensor can be determined by [29],

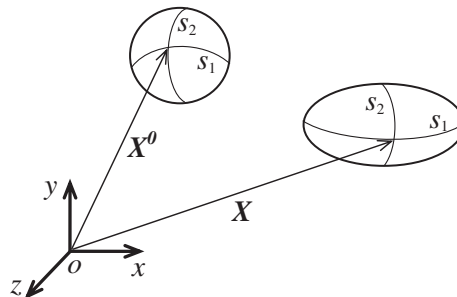


Fig. 1. Schematic of the fluid–flexible body interaction system.

$$\mathbf{S} = \frac{\partial W}{\partial \mathbf{D}} \quad \text{or} \quad S_{ij} = \frac{\partial W}{\partial D_{ij}} \quad (i, j = 1, 2). \quad (6)$$

A simple formulation for W adopted in the present study is

$$W = \varphi \mathbf{D} : \mathbf{D} = \varphi D_{ij} D_{ij}, \quad (7)$$

where φ is a material constant. Using the invariants of \mathbf{D} , W can be expressed equivalently by

$$W = \frac{\varphi}{2} \left(\frac{1}{2} I_1^2 + I_1 - I_2 \right), \quad (8)$$

where $I_1 = 2(D_{11} + D_{22})$, $I_2 = 2(D_{11} + D_{22}) + 4(D_{11}D_{22} - D_{12}^2)$ [30], and the symmetry of \mathbf{D} is used. It is noted that Eq. (8) is actually the Skalak law [30] by neglecting the area dilation effect. By substituting Eqs. (7) or (8) into Eq. (6), we have

$$S_{ij} = 2\varphi D_{ij} \quad (i, j = 1, 2). \quad (9)$$

It is interesting to see that Eq. (9) holds for large deformation as well as small deformation. In the limit of small deformation, the second Piola–Kirchhoff stress converges to the Cauchy stress, and Eq. (9) indicates that φ is actually the shear modulus and the Poisson's ratio is $\nu_s = 0$ by comparing with the 2D Hooke's law.

For comparison, the neo-Hookean law is also considered [31],

$$W^{NH} = \frac{\varphi}{2} \left(I_1 + 2 + \frac{1}{I_2 + 1} \right). \quad (10)$$

Using Eqs. (6) and (10), the second Piola–Kirchhoff stress for the neo-Hookean solid is then given by

$$S_{ij}^{NH} = \varphi \left(\delta_{ij} - (-1)^{i+j} \frac{\delta_{ij} + 2D_{3-i,3-j}}{(I_2 + 1)^2} \right) \quad (i, j = 1, 2), \quad (11)$$

where δ_{ij} denotes the Kronecker delta function. In the limit of small deformation, Eq. (11) indicates that φ is just the shear modulus and the Poisson's ratio is $\nu_s = 0.5$. In the following computations, it is default that Eq. (9) is used while a notation will be given if Eq. (11) is adopted.

Finally, we need to make non-dimensionalization for the governing equations, i.e. Eqs. (1)–(3). Note that ρ is the volume density for 2D flexible body in 2D fluid, but it becomes the surface density for 2D membrane in 3D fluid. Such difference exists for all the terms in the equations of solid motion for these two cases, and thus different characteristic scales should be applied. Here we use the fluid density ρ_0 as the characteristic density and choose certain characteristic velocity and length (denoted by U_0 and L_0 , respectively), which depend on concrete problems. The characteristic time is then given by L_0/U_0 , and the other characteristic scales are listed in Table 1. As a result, Eqs. (1)–(3) are non-dimensionalized as, respectively,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{f}, \quad (12)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (13)$$

$$\rho \frac{\partial^2 \mathbf{X}}{\partial t^2} + \lambda \frac{\partial \mathbf{X}}{\partial t} = \frac{\partial}{\partial s_i} \left(S_{ij} \frac{\partial \mathbf{X}}{\partial s_j} \right) - \mathbf{F}, \quad (14)$$

where $Re = \rho_0 U_0 L_0 / \mu$ denotes the Reynolds number. For convenience, the dimensionless quantities in Eqs. (12)–(14) are written in the same forms as their dimensional counterparts. It is seen that the forms of the continuity equation (Eq. (13)) and the solid motion equation (Eq. (14)) are unchanged after non-dimensionalization.

2.2. Penalty immersed boundary method

The pIB method is an effective method for handling the interaction between the massive flexible body and the fluid [21]. The basic idea of the pIB method is to use two sets of the material points: one set is massive but does not interact with the fluid directly; the other set is massless and interacts with the fluid. The two sets of points are restrained to keep close enough by using a stiff spring so that they behave like a single massive structure interacting with fluid. The massless material points

Table 1
Characteristic scales for variables in the equations of fluid and solid motions.

Fluid		Solid		
		2D		Membrane
p	$\rho_0 U_0^2$	S_{ij}	$\rho_0 U_0^2$	$\rho_0 U_0^2 L_0$
\mathbf{f}	$\rho_0 U_0^2 / L_0$	\mathbf{F}	$\rho_0 U_0^2 / L_0$	$\rho_0 U_0^2$
		ρ	ρ_0	$\rho_0 L_0$
		λ	$\rho_0 U_0 / L_0$	$\rho_0 U_0$

are moved according to the Eulerian fluid velocity, while the positions of the massive material points are calculated in the Lagrangian coordinates. By noting the fact that the fluid and solid motion equations can be written in a unified form in the Eulerian domain, i.e. the N–S equations, Kim and Peskin [21] spread the elastic force of solid directly to the nearby Eulerian points, which implies that the massless material points are subjected to the elastic force of solid as well as the restoring force of the stiff spring. On the other hand, the massive material points are driven only by the restoring force.

In the present study, we made a revision on the original pIB method. The main difference lies on the treatment of the elastic force of solid. Here we remain the elastic force with massive material points, instead of spreading it to the Eulerian fluid points. In other words, the massive material points are subjected to the elastic force due to solid deformation, while the massless material points move following the local Eulerian fluid velocity. In this way, we can make use of the governing equations for fluid and solid motions, i.e. Eqs. (12)–(14), which are linked through the Lagrangian and Eulerian momentum forcings \mathbf{F} and \mathbf{f} . In detail, each massive point and its massless counterpart are connected closely by a stiff spring with damping, and the restoring force acting on the massive material points is denoted by $-\mathbf{F}$ in Eq. (14), which can be expressed by

$$\mathbf{F}(s_1, s_2, t) = -\kappa[(\mathbf{X}_{ib} - \mathbf{X}) + \Delta t(\mathbf{U}_{ib} - \mathbf{U})], \quad (15)$$

where κ is a large constant, Δt denotes the computational time step, \mathbf{X}_{ib} and \mathbf{U}_{ib} denote respectively the position and velocity of the massless material points, and \mathbf{X} and \mathbf{U} denote respectively the position and velocity of the massive material points. From the above discussion, we can see that \mathbf{X} is solved from the solid motion equation (Eq. (14)) and $\mathbf{U} = \partial \mathbf{X} / \partial t$. On the other hand, \mathbf{X}_{ib} and \mathbf{U}_{ib} are determined by the local Eulerian fluid velocity, as expressed by respectively,

$$\mathbf{X}_{ib} = \mathbf{X}_{ib}^0 + \int_0^t \mathbf{U}_{ib} dt, \quad (16)$$

$$\mathbf{U}_{ib}(s_1, s_2, t) = \int_{\Omega_f} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{X}(s_1, s_2, t) - \mathbf{x}) d\mathbf{x}, \quad (17)$$

where $\delta(\cdot)$ denotes the Dirac delta function. Eq. (17) is a formulation of interpolation of the Eulerian fluid velocity at the Lagrangian points, where we should note that the massive material points \mathbf{X} are selected for velocity interpolation, instead of the massless material points \mathbf{X}_{ib} . Since \mathbf{X}_{ib} is allowed to deviate slightly from \mathbf{X} , Shin et al. [15] showed that the selection of \mathbf{X} for velocity interpolation rather than \mathbf{X}_{ib} would require less restriction on the computational time step. Actually, by substituting Eqs. (16) and (17) into the Lagrangian forcing formulation (Eq. (15)), we can see that the present forcing scheme is equivalent to the feedback forcing scheme as proposed by Goldstein et al. [5].

After obtaining the Lagrangian forcing \mathbf{F} , we transform it to the Eulerian form using the Dirac delta function

$$\mathbf{f}(\mathbf{x}, t) = \int_{\Omega_s} \mathbf{F}(s_1, s_2, t) \delta(\mathbf{x} - \mathbf{X}(s_1, s_2, t)) ds_1 ds_2. \quad (18)$$

It should be noted that in the above equation the dimension of the Dirac delta function depends on the fluid flow, i.e. 2D for 2D fluid but 3D for 3D fluid. Hence, for the problem of 2D membrane motion in 3D fluid, the Eulerian momentum forcing \mathbf{f} is singular and may cause a pressure jump in fluid across the membrane. In computation, \mathbf{f} is actually distributed over several grids in width according to the support of the smoothed approximation of the Dirac delta function.

In summary, the pIB method treats the flexible body as an assembly of the massive material points in vacuum and the massless material points interacting with the fluid. In the original pIB method the massless material points are considered as an elastic boundary [21], while in the present method the massive material points are treated as an elastic boundary. Then in both methods each pair of massive and massless points is kept close enough by a stiff spring with damping, so that as an overall effect the interaction between the massive flexible body and the fluid is realized. It should be noted that the massless material points are actually not massless, but neutrally buoyant in the fluid, and the massive material points contain the extra mass. In other words, the fluid in the domain occupied by the flexible body is forced to move with the solid velocity, and thus should be regarded as neutrally buoyant (massless) part of solid. The main advantage of the present pIB method is that the flow solver and the solid solver can be developed separately. To solve the fluid–flexible body interaction problems, they are combined simply by including the Eulerian and Lagrangian momentum forcing terms as discussed above, which can be added to the original solvers as an additional subroutine and requires essentially no modifications on the original solvers. In this sense, our method is general and effective for fluid–flexible body interaction by providing the flow solver and the solid solver. A natural selection is that the fluid flow is solved by the finite difference/volume approach on the Cartesian grid, while the solid motion is solved by the finite element approach on the unstructured grid because the solid geometry is usually complex. Now the two different approaches can be integrated in the framework of the present pIB method aiming at simulating the fluid–flexible body interaction.

3. Numerical method

3.1. Flow solver

The N–S equations (Eqs. (12) and (13)) are discretized on a staggered Cartesian grid system, which can be written as

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + N\mathbf{u}^{n+1} = -Gp^{n+1/2} + \frac{1}{2Re}(\mathbf{L}\mathbf{u}^{n+1} + \mathbf{L}\mathbf{u}^n) + \mathbf{f}^n, \quad (19)$$

$$D\mathbf{u}^{n+1} = 0, \quad (20)$$

where N , G , L and D are the linearized discrete convective operator, the discrete gradient operator, the discrete Laplacian operator, and the discrete divergence operator, respectively. Note that the numerical operators D and G are different with the strain tensor and the deformation gradient tensor \mathbf{D} and \mathbf{G} as introduced in the previous section. Here the velocity components and momentum forcing are defined on the staggered grid, whereas the pressure is applied at the cell center. To solve Eqs. (19) and (20), the fractional step method is adopted, which can be expressed by

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + N\mathbf{u}^* = -Gp^{n-1/2} + \frac{1}{2Re}(\mathbf{L}\mathbf{u}^* + \mathbf{L}\mathbf{u}^n) + \mathbf{f}^n, \quad (21)$$

$$\Delta t DG\delta p = D\mathbf{u}^*, \quad (22)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t G\delta p, \quad (23)$$

$$p^{n+1/2} = p^{n-1/2} - \delta p. \quad (24)$$

where \mathbf{u}^* denotes the intermediate velocity. In the above equations, decoupling of the velocity and pressure is achieved by block LU decomposition in conjunction with approximate factorization. In the momentum equation (Eq. (21)), fully implicit time advancement is employed, with the Crank–Nicholson scheme being used for the discretization of the diffusion and convection terms. Due to the implicit treatment of the nonlinear convection terms, further decoupling of the intermediate velocity components is implemented, and finally a system of tridiagonal matrices is formed instead of a large sparse matrix. Thus, Eq. (21) can be solved directly at a high computational efficiency. The pressure Poisson equation (Eq. (22)) is solved by a direct method using FFT or a multigrid method. The pressure is then used to correct the velocity field to satisfy the continuity equation. More details of the fractional step method for the N–S equations can be found in Kim et al. [32].

3.2. Solid solver

In order to simulate the motion of the flexible body, the finite element method is developed to solve Eq. (14) in the present study. We first divide the physical domain of the flexible body into a set of pieces of elements, which are approximated by an unstructured triangular mesh. Then the weak form of Eq. (14) is derived by the variational method and is discretized on the triangular elements. Finally we can obtain a linear system of algebraic equations from which the positions of the flexible body at vertices (nodes) of the triangular elements are solved.

The unstructured triangular mesh is generated by the quadrissection method as demonstrated in Fig. 2, where several geometries are shown including a circular disk, a circular ring, a circular cylinder and a sphere. Initially, we set up manually a coarse mesh for sketching the desired geometry, as shown in the column of level 0 in Fig. 2. For instance, for a circular disk, the vertices of a regular polygon (e.g. hexagon) and its center are selected as the initial nodes. At a time each triangle is then quadrisected into four triangles by connecting the midpoints of the three edges. Starting from the initial configuration (level 0), the mesh is gradually approaching to the real geometries by quadrissections (level 1–3), as shown in Fig. 2. We can also see that by giving a regular initial grid, the quadrisected grid is always regular, which greatly facilitates the grid refinement test. Here we use the cylindrical and spherical coordinates for mesh generation in cases of a circular cylinder and a sphere (Fig. 2(c) and (d)), respectively, and after quadrissection we calculate the nodal positions in the Cartesian coordinates.

To derive the weak form of the solid motion equations, a weighted average of Eq. (14) is expressed by

$$\iint_{\Omega_s} \delta X \left(\rho \frac{\partial^2 X}{\partial t^2} + \lambda \frac{\partial X}{\partial t} \right) ds_1 ds_2 = \iint_{\Omega_s} \delta X \frac{\partial}{\partial s_i} \left(S_{ij} \frac{\partial X}{\partial s_j} \right) ds_1 ds_2 - \iint_{\Omega_s} \delta X F ds_1 ds_2, \quad (25)$$

where X is a component of the position variable \mathbf{X} , F is the corresponding component of the Lagrangian forcing \mathbf{F} , and δX is the variation of X . The above equation applies equivalently for the other components. Using integration by parts, the elastic force term in Eq. (25) becomes

$$\iint_{\Omega_s} \delta X \frac{\partial}{\partial s_i} \left(S_{ij} \frac{\partial X}{\partial s_j} \right) ds_1 ds_2 = - \iint_{\Omega_s} \frac{\partial \delta X}{\partial s_i} S_{ij} \frac{\partial X}{\partial s_j} ds_1 ds_2 + \oint_{\Gamma} \delta X S_{ij} \frac{\partial X}{\partial s_j} n_i d\Gamma, \quad (26)$$

where Γ denotes the boundary of the solid domain Ω_s and n_i is the component of its outer normal unit vector along the s_i direction. Gauss' theorem is used to derive the surface integral term, i.e. the second term on the right hand side of Eq. (26). In the present study, we set the solid boundary to be either fixed or traction free, so the surface integral term in Eq. (26) vanishes. Thus Eq. (25) can be rewritten as

$$\iint_{\Omega_s} \delta X \left(\rho \frac{\partial^2 X}{\partial t^2} + \lambda \frac{\partial X}{\partial t} + F \right) ds_1 ds_2 = - \iint_{\Omega_s} \frac{\partial \delta X}{\partial s_i} S_{ij} \frac{\partial X}{\partial s_j} ds_1 ds_2 - \iint_{\Omega_s} \delta X F ds_1 ds_2. \quad (27)$$

Next, we make finite element approximations to Eq. (27) on the triangular mesh,

$$\sum_{m=1}^M \iint_{e_m} \delta X \left(\rho \frac{\partial^2 X}{\partial t^2} + \lambda \frac{\partial X}{\partial t} + F \right) ds_1 ds_2 = - \sum_{m=1}^M \iint_{e_m} \frac{\partial \delta X}{\partial s_i} S_{ij} \frac{\partial X}{\partial s_j} ds_1 ds_2, \quad (28)$$

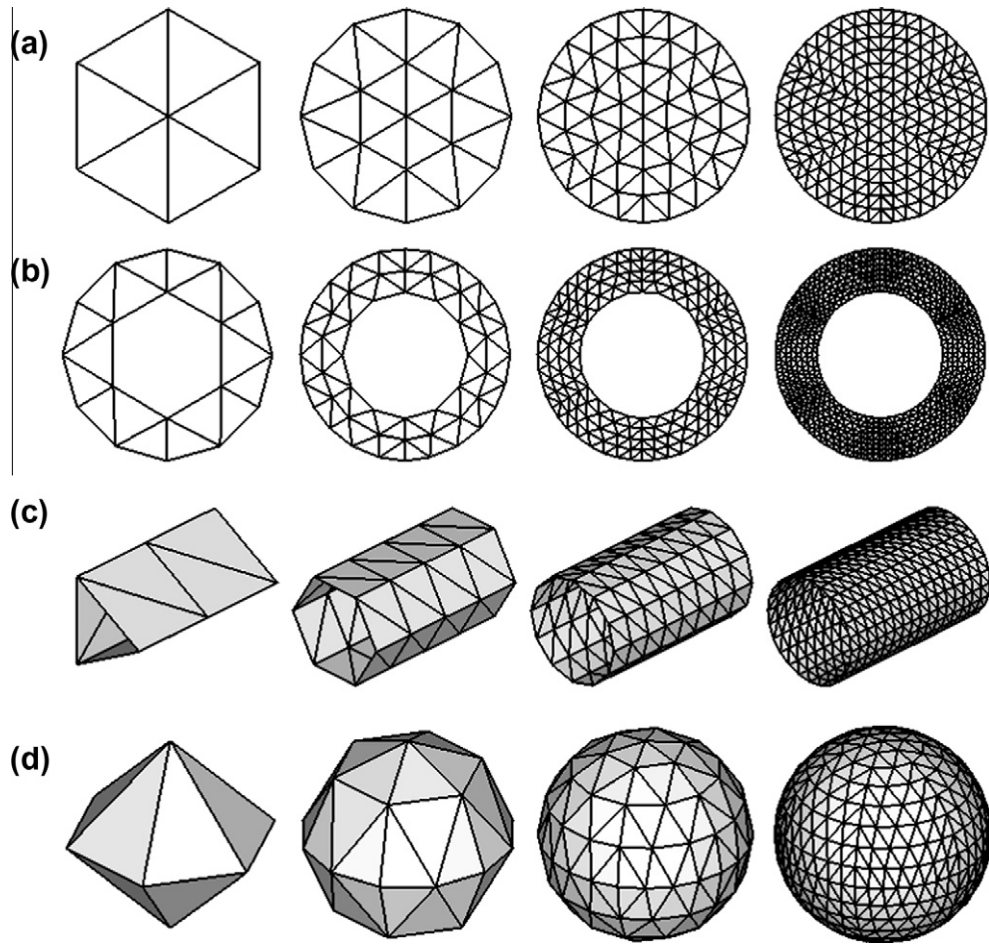


Fig. 2. Unstructured mesh generation for different geometries: (a) circular disk; (b) circular ring; (c) circular cylinder; (d) sphere. Each column represents a level of quadrisection, i.e. level 0 (initial configuration) to level 3 from left to right.

where e_m denotes the m th triangular element and M denotes the total element number. For a specific element e_m , an isoparametric formulation is established by using the shape functions based on the local coordinate system, i.e.

$$s_i = N_1 s_i^1 + N_2 s_i^2 + N_3 s_i^3 \quad (i = 1, 2), \quad (29)$$

where s_i^k ($k = 1, 2, 3$) denotes the value of the curvilinear coordinate s_i^k at the three vertices of the triangular element e_m and N_k ($k = 1, 2, 3$) denotes the shape functions for a three-node triangular element which is expressed in a matrix form as

$$\mathbf{N} = (N_1 \quad N_2 \quad N_3) = (1 \quad \xi_1 \quad \xi_2) \begin{pmatrix} 1 & \xi_1^1 & \xi_2^1 \\ 1 & \xi_1^2 & \xi_2^2 \\ 1 & \xi_1^3 & \xi_2^3 \end{pmatrix}^{-1}, \quad (30)$$

where ξ_i ($i = 1, 2$) denotes the local coordinates and ξ_i^k ($k = 1, 2, 3$) is the value of ξ_i at the three vertices of the isoparametric triangle which can be selected freely. Here we set (ξ_1, ξ_2) at the three vertices to be $(0, 0)$, $(1, 0)$ and $(0, 1)$, respectively, and Eq. (30) is simplified to

$$\mathbf{N} = (N_1 \quad N_2 \quad N_3) = (1 - \xi_1 - \xi_2 \quad \xi_1 \quad \xi_2). \quad (31)$$

So the above transformation (Eqs. (29) and (31)) indicates that all the triangular elements are mapped into an isosceles right angled triangle in the isoparametric space. Similarly, other field variables can also be approximated by the isoparametric formulation, e.g.

$$X = N_1 X^1 + N_2 X^2 + N_3 X^3 = \mathbf{N} \mathbf{X}_{e_m}, \quad (32)$$

$$F = N_1 F^1 + N_2 F^2 + N_3 F^3 = \mathbf{N} \mathbf{F}_{e_m}, \quad (33)$$

where \mathbf{X}_{e_m} and \mathbf{F}_{e_m} denote the arrays of X and F at three vertices of a triangular element e_m , respectively, i.e. $\mathbf{X}_{e_m} = (X^1 \ X^2 \ X^3)^T$ and $\mathbf{F}_{e_m} = (F^1 \ F^2 \ F^3)^T$, and the superscript denotes the local index of triangle vertices. The derivatives of X with respect to s_i are also transformed to the isoparametric space, i.e.

$$\begin{pmatrix} \frac{\partial X}{\partial \xi_1} \\ \frac{\partial X}{\partial \xi_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial s_1}{\partial \xi_1} & \frac{\partial s_2}{\partial \xi_1} \\ \frac{\partial s_1}{\partial \xi_2} & \frac{\partial s_2}{\partial \xi_2} \end{pmatrix} \begin{pmatrix} \frac{\partial X}{\partial s_1} \\ \frac{\partial X}{\partial s_2} \end{pmatrix} = \mathbf{J} \begin{pmatrix} \frac{\partial X}{\partial s_1} \\ \frac{\partial X}{\partial s_2} \end{pmatrix}, \quad (34)$$

where the chain rule is applied and \mathbf{J} denotes the Jacobian matrix. In Eq. (34), the derivatives with respect to the local coordinates can be determined as,

$$\begin{pmatrix} \frac{\partial X}{\partial \xi_1} \\ \frac{\partial X}{\partial \xi_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial N_1}{\partial \xi_1} & \frac{\partial N_2}{\partial \xi_1} & \frac{\partial N_3}{\partial \xi_1} \\ \frac{\partial N_1}{\partial \xi_2} & \frac{\partial N_2}{\partial \xi_2} & \frac{\partial N_3}{\partial \xi_2} \end{pmatrix} \begin{pmatrix} X^1 \\ X^2 \\ X^3 \end{pmatrix} = \mathbf{H} \mathbf{X}_{e_m}, \quad (35)$$

$$\mathbf{J} = \begin{pmatrix} \frac{\partial s_1}{\partial \xi_1} & \frac{\partial s_2}{\partial \xi_1} \\ \frac{\partial s_1}{\partial \xi_2} & \frac{\partial s_2}{\partial \xi_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial N_1}{\partial \xi_1} & \frac{\partial N_2}{\partial \xi_1} & \frac{\partial N_3}{\partial \xi_1} \\ \frac{\partial N_1}{\partial \xi_2} & \frac{\partial N_2}{\partial \xi_2} & \frac{\partial N_3}{\partial \xi_2} \end{pmatrix} \begin{pmatrix} s_1^1 & s_2^1 \\ s_1^2 & s_2^2 \\ s_1^3 & s_2^3 \end{pmatrix} = \mathbf{H} \begin{pmatrix} s_1^1 & s_2^1 \\ s_1^2 & s_2^2 \\ s_1^3 & s_2^3 \end{pmatrix}. \quad (36)$$

Specifically, from Eq. (31) the above defined matrix \mathbf{H} is given by

$$\mathbf{H} = \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}. \quad (37)$$

From Eq. (36) we can see that the Jacobian matrix \mathbf{J} is invariant with time, but depends on the Lagrangian curvilinear coordinates of the vertices of the triangular element. Substituting Eq. (35) into Eq. (34) yields

$$\begin{pmatrix} \frac{\partial X}{\partial s_1} \\ \frac{\partial X}{\partial s_2} \end{pmatrix} = \mathbf{J}^{-1} \mathbf{H} \mathbf{X}_{e_m}. \quad (38)$$

By applying Eq. (38) for all the components of \mathbf{X} , we can obtain the deformation gradient tensor $\mathbf{G} = \partial \mathbf{X} / \partial s$, or $G_{ij} = \partial X_i / \partial s_j$, as presented in the previous section. Accordingly, the Green strain tensor \mathbf{D} is obtained by Eqs. (4) or (5), and the second Piola–Kirchhoff stress tensor \mathbf{S} is also obtained by Eqs. (9) or (11).

By substituting Eqs. (32), (33) and (38) into Eq. (28), we have

$$\sum_{m=1}^M \delta \mathbf{X}_{e_m}^T \left(\iint_{e_m} \mathbf{N}^T \mathbf{N} \mathbf{J} |d\xi_1 d\xi_2| \right) \left(\rho \frac{\partial^2 \mathbf{X}_{e_m}}{\partial t^2} + \lambda \frac{\partial \mathbf{X}_{e_m}}{\partial t} + \mathbf{F}_{e_m} \right) = - \sum_{m=1}^M \delta \mathbf{X}_{e_m}^T \left(\iint_{e_m} \mathbf{H}^T \mathbf{J}^{-T} \mathbf{S} \mathbf{J}^{-1} \mathbf{H} \mathbf{J} |d\xi_1 d\xi_2| \right) \mathbf{X}_{e_m}, \quad (39)$$

where $|\mathbf{J}|$ denotes the determinant of \mathbf{J} , and the integrals are transformed to the local coordinate system in the isoparametric space. Note that in Eq. (39) it is better to interpolate the symbol Σ as a matrix assembling operation rather than a mathematical summation. Although the integral parts can be read as 3×3 matrices using the local nodal index, i.e. 1–3, for a specific triangular element, when doing a summation over all the elements we have to transform the local nodal index into the corresponding global index. Then according to the global nodal indices of each triangular element, the integral parts in Eq. (39) are assembled into $N \times N$ matrices where N denotes the total node number. Moreover, the variation term can be removed from Eq. (39) and we have

$$\mathbf{M} \left(\rho \frac{\partial^2 \mathbf{X}_e}{\partial t^2} + \lambda \frac{\partial \mathbf{X}_e}{\partial t} + \mathbf{F}_e \right) + \mathbf{K} \mathbf{X}_e = 0, \quad (40)$$

where

$$\mathbf{M} = \sum_{m=1}^M \iint_{e_m} \mathbf{N}^T \mathbf{N} \mathbf{J} |d\xi_1 d\xi_2|, \quad \mathbf{K} = \sum_{m=1}^M \iint_{e_m} \mathbf{H}^T \mathbf{J}^{-T} \mathbf{S} \mathbf{J}^{-1} \mathbf{H} \mathbf{J} |d\xi_1 d\xi_2|, \quad (41)$$

and \mathbf{X}_e and \mathbf{F}_e denote the array of the X and F at all the nodes with indices ranging from 1 to N . Note the difference between the nodal value arrays \mathbf{X}_e , \mathbf{F}_e and the field variables \mathbf{X} , \mathbf{F} . In Eq. (41), the 2D integrals over the isoparametric triangle can be calculated efficiently by using a Gaussian quadrature. In the present study, a three-point Gaussian quadrature for the isoparametric triangle is adopted, with the integration points located at $(2/3, 1/6)$, $(1/6, 1/6)$, and $(1/6, 2/3)$, and the three weighting coefficients all being $1/3$. Then we apply the temporal discretization on Eq. (40), which yields

$$\mathbf{M} \left[\rho \frac{(\mathbf{X}_e^{n+1} - 2\mathbf{X}_e^n + \mathbf{X}_e^{n-1}))}{\Delta t^2} + \lambda \frac{(\mathbf{X}_e^{n+1} - \mathbf{X}_e^{n-1}))}{2\Delta t} + \mathbf{F}_e^n \right] + \mathbf{K} \mathbf{X}_e^{n+1} = 0, \quad (42)$$

where the superscript n denotes the n th step of time advancement. It should be pointed out that the elastic force term is treated implicitly, while the stiff matrix \mathbf{K} is evaluated explicitly. Moreover, the Lagrangian forcing \mathbf{F}_e is calculated explicitly, but \mathbf{M} is time-independent. After rearrangement, Eq. (42) becomes

$$\mathbf{A}\mathbf{X}_e^{n+1} = \mathbf{B}, \quad (43)$$

where $\mathbf{A} = (\rho + \lambda\Delta t/2)\mathbf{M} + \Delta t^2\mathbf{K}^n$ and $\mathbf{B} = \mathbf{M}[2\rho\mathbf{X}_e^n - (\rho - \lambda\Delta t/2)\mathbf{X}_e^{n-1} - \Delta t^2\mathbf{F}_e^n]$. By using $\mathbf{U}_e^n = (\mathbf{X}_e^n - \mathbf{X}_e^{n-1})/\Delta t$, the right hand side of Eq. (43) is rewritten as $\mathbf{B} = \mathbf{M}[(\rho + \lambda\Delta t/2)\mathbf{X}_e^n + (\rho - \lambda\Delta t/2)\Delta t\mathbf{U}_e^n - \Delta t^2\mathbf{F}_e^n]$. Thus, by giving the initial position and velocity of the flexible body, Eq. (43) can be solved iteratively by the Gauss–Seidel method. In the above linear system, the traction-free boundary condition is automatically satisfied. However, the fixed boundary condition requires some modifications on the linear system. In detail, if the i th ($1 \leq i \leq N$) node is located on the fixed boundary, we change the entries on the i th line of \mathbf{A} to be zero except the (i, i) entry which is set to be unit. Meanwhile the i th entry of \mathbf{B} is given the desired value of \mathbf{X} of the i th node.

3.3. Implementation of the penalty IB method

Although the flow solver and the solid solver are developed separately, they can be assembled to become a fluid-flexible solver by the penalty IB method. As discussed in the previous section, despite the overlap of the fluid and solid computational domains, the fluid motion is solved on the whole Eulerian domain, thus significantly simplifying the complexity of the numerical scheme. The coupling of the fluid and solid motions is realized through the Lagrangian and Eulerian momentum forcings. In the present method, a weak coupling scheme is adopted in computation. In detail, the Lagrangian forcing in Eq. (43) is calculated by

$$\mathbf{F}^n = -\kappa[(\mathbf{X}_{ib}^n - \mathbf{X}^n) + \Delta t(\mathbf{U}_{ib}^n - \mathbf{U}^n)], \quad (44)$$

where the superscript n denotes the current time step. In the above equation, \mathbf{X}_{ib}^n and \mathbf{U}_{ib}^n are obtained as follows, according to Eqs. (16) and (17) respectively,

$$\mathbf{X}_{ib}^n = \mathbf{X}_{ib}^0 + \sum_{i=1}^n \mathbf{U}_{ib}^i \Delta t = \mathbf{X}_{ib}^{n-1} + \mathbf{U}_{ib}^n \Delta t, \quad (45)$$

$$\mathbf{U}_{ib}^n = \sum_{\mathbf{x} \in g_h} \mathbf{u}^n \delta_h(\mathbf{X}^n - \mathbf{x}) \Delta V, \quad (46)$$

where $\Delta V = h^2$ for the 2D flow or $\Delta V = h^3$ for the 3D flow, and similarly

$$\delta_h(\mathbf{x}) = \frac{1}{h^2} \phi\left(\frac{x}{h}\right) \phi\left(\frac{y}{h}\right) \quad \text{or} \quad \delta_h(\mathbf{x}) = \frac{1}{h^3} \phi\left(\frac{x}{h}\right) \phi\left(\frac{y}{h}\right) \phi\left(\frac{z}{h}\right), \quad (47)$$

where

$$\phi(r) = \begin{cases} \frac{1}{8} (3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}), & 0 \leq |r| < 1, \\ \frac{1}{8} (5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}), & 1 \leq |r| < 2, \\ 0, & 2 \leq |r| \end{cases} \quad (48)$$

denotes the four-point smoothed delta function [19]. Here the Eulerian grid size is assumed to be uniformly distributed inside or near the solid domain along all the Cartesian axes, and is denoted by h . Moreover, to improve the computational efficiency in Eq. (46), the support for the smoothed delta function denoted by g_h is used for integration instead of the whole domain as in Eq. (17). After calculating \mathbf{F}^n , the Eulerian forcing \mathbf{f}^n in Eq. (19) is obtained using Eq. (18), which is discretized as,

$$\mathbf{f}^n = \sum_{\mathbf{x} \in \Omega_S} \mathbf{F}^n \delta_h(\mathbf{x} - \mathbf{X}^n) \Delta S, \quad \forall \mathbf{x} \in g_h, \quad (49)$$

where ΔS denotes the surface area associated with each node of the triangular mesh which can be estimated using the area of the surrounding triangles. It should be pointed out that the Lagrangian forcing (Eq. (44)) is calculated on all the solid nodes, and is then distributed to the nearby Eulerian grids by the smoothed delta function (Eq. (49)). This implies that the fluid overlapped by the solid domain is forced to move with the solid nodes, thus behaving like a neutrally buoyant solid.

The overall computation process of the present numerical algorithm for simulating the fluid-flexible body interaction is summarized as follows:

- (1) Initialize the computation parameter values, the meshes, and the fluid and solid motions; set $\mathbf{X}_{ib}^0 = \mathbf{X}^0$.
- (2) At the n th time step, we know the fluid velocity \mathbf{u}^n and the solid position \mathbf{X}^n and velocity \mathbf{U}^n . Interpolate the fluid velocity at the Lagrangian points to obtain \mathbf{U}_{ib}^n using Eq. (46), and calculate the position of the massless points \mathbf{X}_{ib}^n using Eq. (45). Then calculate the Lagrangian momentum forcing \mathbf{F}^n using Eq. (44).

- (3) Spread the Lagrangian momentum forcing to the Eulerian grid using Eq. (49). Solve Eqs. (21)–(24) to obtain the updated fluid velocity field and the pressure field.
- (4) Substitute \mathbf{F}^n into Eq. (43) and solve this linear system to obtain the solid position at the new time step \mathbf{X}^{n+1} , as well as the structure velocity $\mathbf{U}^{n+1} = (\mathbf{X}^{n+1} - \mathbf{X}^n)/\Delta t$. Return to step 2 and march to the next time step.

4. Numerical examples

4.1. Vibrations of a circular ring in vacuum

In this section, we first validate the solid solver by simulating a vibrating elastic circular ring in the absence of ambient fluid. The initial shape of the elastic ring is circular, and is discretized using the unstructured mesh as shown in Fig. 2(b). At the undeformed status, the initial diameter of the outer circle is set to be 1 and the ring thickness is denoted by a . Here we consider three cases: (1) The outer boundary is fixed with a small displacement in the radial direction, while the inner boundary is fixed without displacement. (2) The outer boundary is the same as Case 1 but the inner boundary is free. (3) Both the outer and the inner boundaries are free, and the deformation is caused by giving an initial velocity. To explain more clearly, the deformed shapes together with the undeformed reference frame are depicted in Fig. 3 for the three cases.

In the following simulations, we set $\varphi = 0.5$ and $\rho = 1.0$ for the elastic ring, and use a time step of $\Delta t = 0.001$. Since the fluid is absent, the Lagrangian forcing is zero in Eq. (14). For Cases 1 and 2, the damping effect is considered by setting $\lambda = 1.0$, so the ring finally becomes stationary after a sufficient long time. Therefore, the problem can be approximated as static and linearly elastic, and the theoretical solution of the radial displacement is expressed by [33],

$$d(r) = -\frac{R_1^2 R_2}{R_2^2 - R_1^2} \frac{s}{r} + \frac{R_2}{R_2^2 - R_1^2} r s, \quad (50)$$

and

$$d(r) = \frac{R_1^2 R_2}{R_1^2 + R_2^2} \frac{s}{r} + \frac{R_2}{R_1^2 + R_2^2} r s, \quad (51)$$

for Case 1 and Case 2, respectively, where r denotes the distance to the ring center, R_1 and R_2 denote the radii of the inner and outer boundaries, and s denotes the displacement of the outer boundary. Here we have made use of the fact that the Poisson's ratio $\nu_s = 0$ as implied in the constitutive law of Eq. (9). Fig. 4 shows a comparison of the present numerical results with the theoretical solutions (Eqs. (50) and (51)) for $s = 0.05$, $R_1 = 0.3$ and $R_2 = 0.5$ (i.e. $a = 0.2$), and we can see a good agreement is obtained. For Case 3, the elastic ring is set to vibrate freely by giving an initial velocity field,

$$(U, V) = (0.1 \cos \theta, 0), \quad (52)$$

where θ denotes the azimuth measured counter-clockwise from the rightward horizontal radius. In this simulation, the damping effect is neglected, i.e. $\lambda = 0$, and the ring is initially undeformed. Fig. 5 shows the time history of the transverse position of the vertex point and the corresponding energy spectrum for $a = 0.25$. It is observed that there are two modes of vibrations, i.e. flexural vibration and extensional vibration. Moreover, we can also see a higher harmonic besides the two modes in Fig. 5, which is attributed to the fact that the frequency of the extensional vibration is close to a multiple of that of the flexural vibration. However, this harmonic disappears quickly by changing the parameters such as the ring thickness and the shear modulus. Under the assumption that the ring thickness is small in comparison with the initial radius of the center line ($r_c = 0.5 - a/2$), we can obtain theoretically the natural frequencies for the flexural vibration and the extensional vibration as follows [34], respectively,

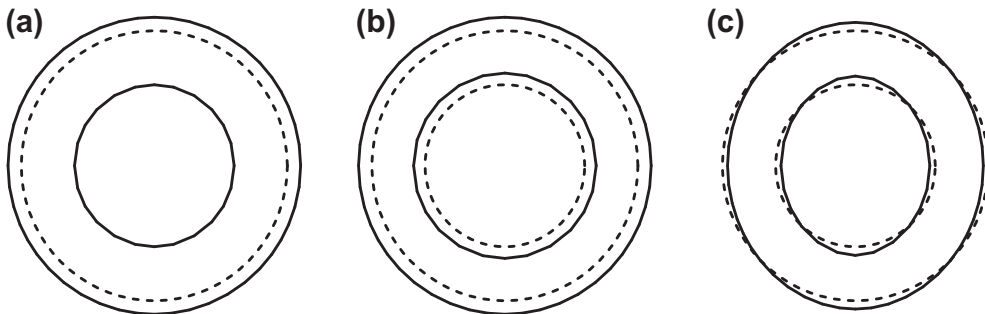


Fig. 3. Deformation of the elastic circular ring: (a) Case 1: Fixed inner boundary and fixed displacement of outer boundary; (b) Case 2: Free inner boundary and fixed displacement of outer boundary; (c) Case 3: Free inner and outer boundaries. Solid line: deformed shape; dashed line: undeformed shape.

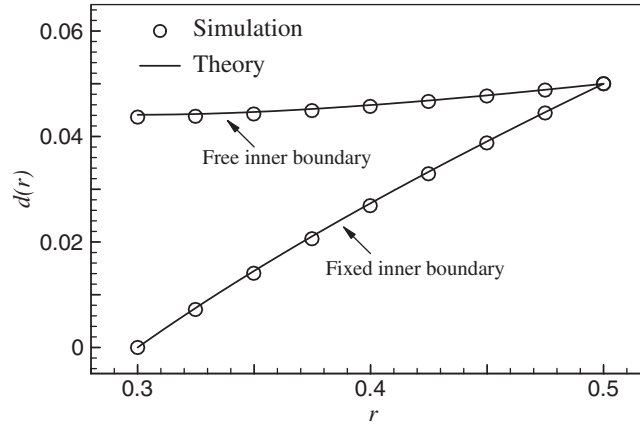


Fig. 4. Nodal displacement on the horizontal line ($y = 0$) of the circular ring where a small displacement $s = 0.05$ is given at the outer boundary while the inner boundary is either fixed or free.

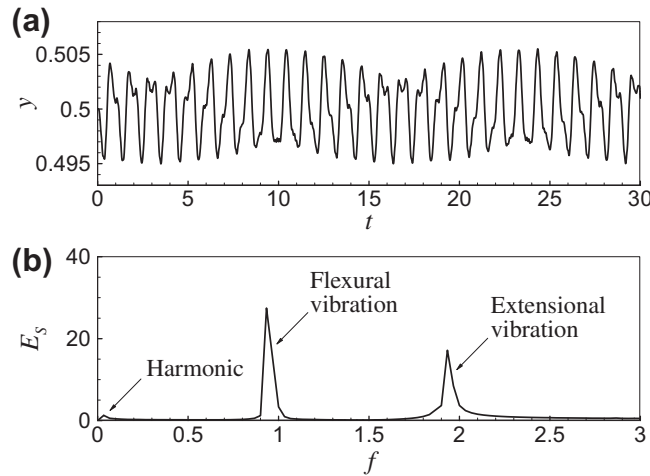


Fig. 5. Free vibration of an elastic circular ring: (a) time history of the transverse position of the vertex and (b) its energy spectrum.

$$f_F = \frac{1}{2\pi} \sqrt{\frac{Ei^2(1-i^2)^2}{\rho A r_c^4(1+i^2)}} \quad \text{and} \quad f_E = \frac{1}{2\pi} \sqrt{\frac{E(1+i^2)}{\rho r_c^2}}, \quad (53)$$

where E denotes Young's modulus, I denotes the second moment of area of cross section about the center line, A denotes the area of the cross section, and i denotes the order of modes. For the present case, it is easily obtained that $I = a^3/12$ and $A = a$ per unit axial length, and $E = 2\phi$. Taking the fundamental frequencies of both vibration modes, i.e. $i = 2$ for the flexural vibration and $i = 0$ for the extensional vibration, the theoretical solutions (Eq. (53)) are plotted in Fig. 6 as functions of ring thickness. The results from our numerical simulation are also presented for comparison. A good agreement can be seen in Fig. 6, especially for $a \leq 0.2$. The grid refinement test is also carried out for the three cases. Fig. 7 presents the L_∞ and L_2 norms of the relative error of the transverse positions as functions of the mesh size. We can see that the error shows a second-order convergence rate for all the three cases as the mesh size decreases.

4.2. Two-dimensional simulation of a circular disk in shear flows

4.2.1. Deformation of a circular disk in a linear shear flow

The initial undisturbed velocity field is given by

$$u^0(x, y) = \dot{\gamma}y, \quad v^0 = 0, \quad (54)$$

where $\dot{\gamma}$ denotes the shear rate. After startup of simulation, the fluid velocities remain unchanged at the boundaries of the fluid domain, but evolve inside the domain due to the interaction with an elastic circular disk. In computation, the fluid do-

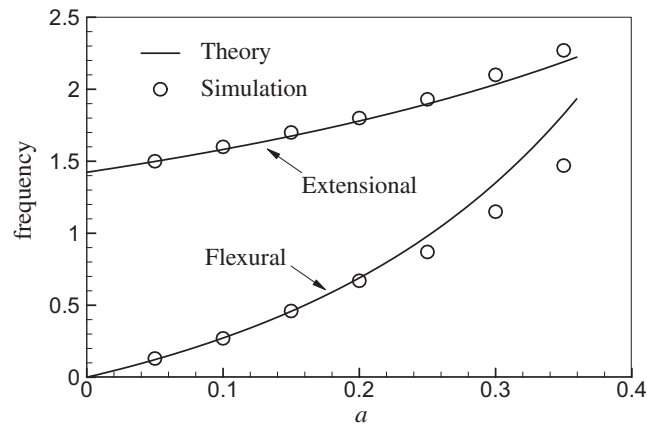


Fig. 6. Natural frequencies of the ring with flexural vibration and extensional vibration as functions of ring thickness.

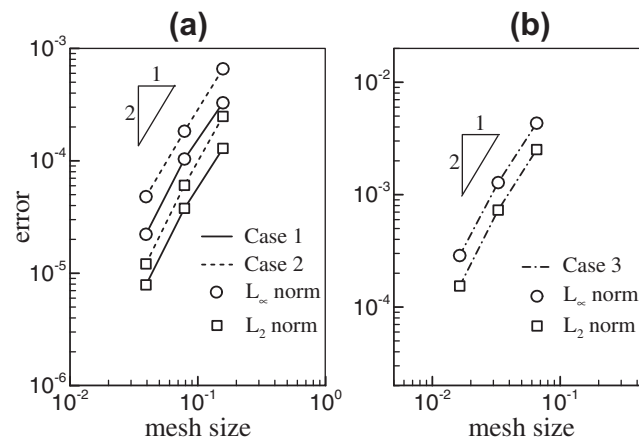


Fig. 7. Convergence of nodal positions of the circular ring with mesh size for (a) Cases 1 and 2 and (b) Case 3.

main ranges from -2 to 2 in the x -direction, and from -1 to 1 in the y -direction, which are scaled by the initial diameter of the circular disk. The disk is initially located at the center of the fluid domain. The unstructured mesh of the circular disk is plotted in Fig. 2(a). The fluid domain is discretized by 129×65 grids uniformly, and the 4th level of quadrissection of the unstructured mesh is used for the disk, which includes totally 817 nodes and 1536 elements. Fig. 8 shows the instantaneous disk shapes together with the streamlines of the fluid motion. Parameter values of fluid properties are $\dot{\gamma} = 1$ and $Re = 1$ based on the initial disk diameter, and those of solid material properties are $\varphi = 5$, $\rho = 0.1$ and $\lambda = 0$. The free constant in Eq. (44) is set to be $\kappa = 10^5$, and a time step of $\Delta t = 0.0003$ is adopted for computation, which are unchanged in the following simulations unless otherwise stated. As shown in Fig. 8, the initially circular disk is deformed by the shear flow and is tilted to the flow direction. Gradually, the disk and the flow field reach a steady state when the disk shape and the surrounding streamlines do not change any more. It should be pointed out that the disk always rotates about its center due to the shearing effect of the surrounding flow. It is also interesting to see that the streamlines penetrate the disk before the steady state is reached, which reflects the local velocity of the solid motion. At the steady state, the streamlines do not cross over the fluid–solid interface, indicating that the normal component of the interfacial velocity vanishes. Fig. 9 shows the relative error of fluid velocities, fluid pressure and solid positions as functions of the mesh size. Here we use the numerical solution described above as the exact one to be compared with those calculated on coarser meshes. In the L_2 norm (Fig. 9(a)), we can obtain that $\varepsilon_u \sim O(h^{1.6})$, $\varepsilon_v \sim O(h^{1.6})$, $\varepsilon_p \sim O(h^{1.8})$, $\varepsilon_x \sim O(h^{1.8})$ and $\varepsilon_y \sim O(h^{1.8})$, where h denotes the mesh size and may be different for the fluid and the solid. However, in the L_∞ norm (Fig. 9(b)), it is obtained that $\varepsilon_u \sim O(h^{1.1})$, $\varepsilon_v \sim O(h^{1.1})$, $\varepsilon_p \sim O(h^{1.5})$, $\varepsilon_x \sim O(h^{1.8})$ and $\varepsilon_y \sim O(h^{1.6})$, indicating that the convergence rate is lower than that in the L_2 norm, especially for the fluid solver, due to the use of the smoothed delta function. Examining the spatial distribution of the relative error indicates that the maximum error is always located near the fluid–solid interface, which is not shown here. Moreover, except the four-point delta function used in the present simulation, we also tested different smoothed delta functions in the present simulation, e.g. the two-, three- and six-point delta functions, and obtained similar results on the numerical accuracy of the proposed method.

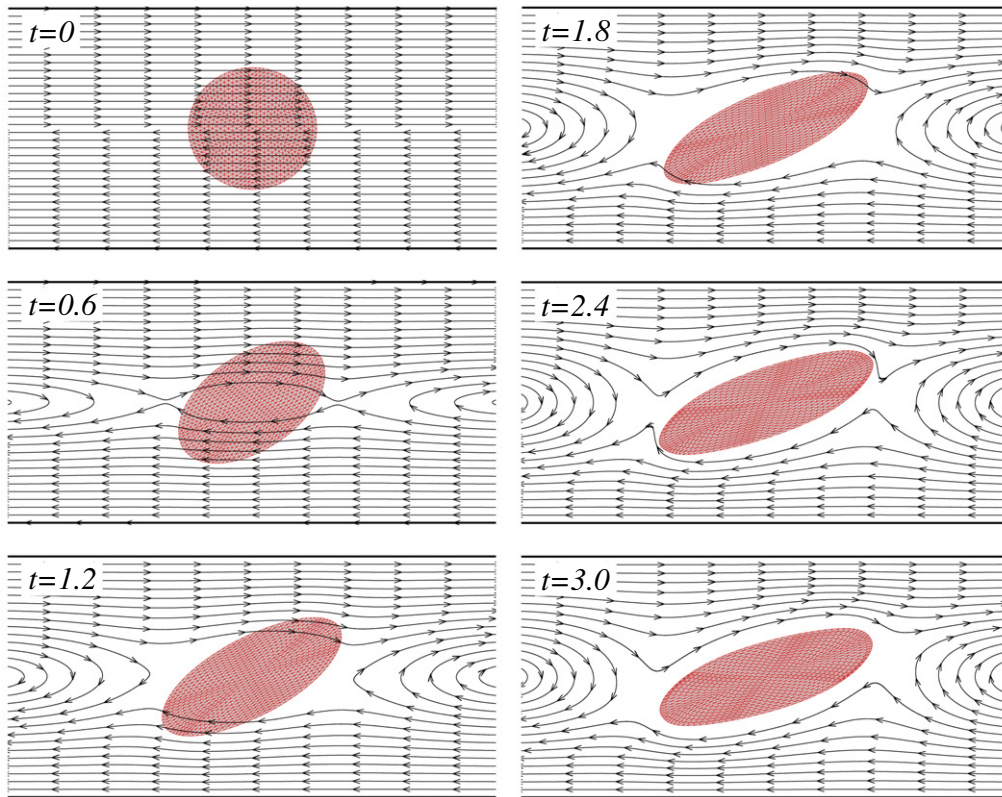


Fig. 8. Deformation of a circular disk in a linear shear flow at $Re = 1$ and $\dot{\gamma} = 1$: streamlines and disk shapes.

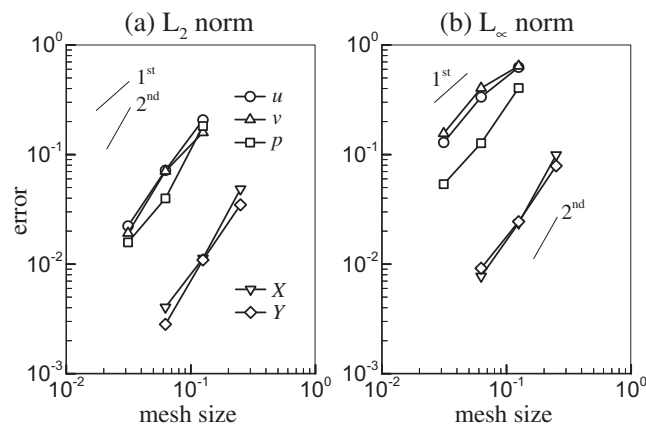


Fig. 9. Convergence of fluid velocities, fluid pressure and solid positions with mesh size: (a) L_2 norm error; (b) L_∞ norm error.

4.2.2. An elastic circular disk moving through a constricted channel

In this example, we examine the interaction between the fluid and the flexible body with the presence of rigid boundaries. The fluid domain is a channel with a symmetric circular constriction at the middle of the length. The channel ranges from -8 to 8 in the streamwise (x) direction and from -1 to 1 in the normal (y) direction. At $x = 0$, there are half-circle rigid obstacles mounted on the upper and lower walls, both of which have radius of 0.6 . A flow with a parabolic velocity profile, i.e. $u^0(x, y) = 1 - y^2$, is given as the initial flow field. Dirichlet boundary conditions ($u = 0$, $v = 0$) are applied at the top and bottom walls ($y = \pm 1$), while a constant inflow of parabolic profile is given at the inlet ($x = -8$) and a convective boundary condition is used at the outlet ($x = 8$). The no-slip conditions on the constricted surfaces are enforced by the IB method. An initially circular disk is located at $x = -4$, $y = 0$ and is released at $t = 0$. A grid size of 257×33 is used for the fluid domain, while the 3rd level quadrisection of the unstructured mesh (217 nodes and 384 elements) is adopted for the solid domain.

Fig. 10 shows the instantaneous disk positions and the streamwise fluid velocity field at $Re = 10$. The solid material parameter values are set to be $\varphi = 50$, $\rho = 0.1$ and $\lambda = 0$. Before encountering the constricted region, the disk moves by following the fluid flow with slight deformation. It is then compressed due to the acceleration of fluid when moving through the constricted region, without contacting directly with the rigid walls. After passing the constricted region, the disk is relaxed with elastic vibration and is gradually recovering its original circular shape. It is interesting to see that since there is no direct contact between the disk and the walls due to the lubrication force, we do not need to build a physical model for the solid–solid contacting phenomena. The instantaneous pressure field is shown in Fig. 11 together with the disk positions. As the disk approaches the constricted region ($t = 3.5$), it is decelerated and causes the jam of the flow, thus the pressure is increased greatly upstream and is decreased downstream. When the disk is moving through the constricted region ($t = 4.0$), it is accelerated and makes the pressure decreased both upstream and downstream. After the disk passes the constricted region ($t = 4.5$), it is decelerated again and the pressure field is similar to that of $t = 3.5$. Then the pressure difference across the constricted region is decreased as the disk moves downstream. From this example we can see a smooth solution is obtained for fluid velocity and pressure in the whole computational domain.

4.3. Three-dimensional simulation of fluid–membrane interaction

4.3.1. Deformation of a spherical capsule in a linear shear flow

The initial undisturbed velocity field is similar with that described by Eq. (54), but now it is extended to 3D, i.e.

$$u^0(x, y, z) = \dot{\gamma}y, \quad v^0 = w^0 = 0. \quad (55)$$

The fluid domain size is $16 \times 16 \times 16$ in the x -, y - and z -directions respectively, which is discretized uniformly by $65 \times 65 \times 65$ grids in the three directions. On the other hand, an initially spherical capsule of radius 1 is located at the center of the fluid domain, and the 3rd level quadrissection of the unstructured mesh (386 nodes and 768 elements) is applied for discretization of the spherical capsule, as shown in Fig. 2(d). The boundary conditions for the fluid motion are the same as the 2D case in the x - and y -directions, while in the z -direction we adopt a periodic condition. Fig. 12 shows the steady state of the deformation of an initially spherical capsule in a linear shear flow at $\dot{\gamma} = 1$ and $Re = 0.025$ based on its initial radius. Results of four different dimensionless shear rates are compared, i.e. $\eta = 0.1, 0.05, 0.025$, and 0.0125 , which is defined as $\eta = \mu\dot{\gamma}a/E_s$ with a the capsule radius and E_s the surface Young's modulus. Parameter values of the capsule material properties are set to be $\rho = 0.1$ and $\lambda = 0$, while $E_s = 400, 800, 1600$ and 3200 according to the dimensionless shear rate. From Fig. 12 we can see that as η increases, the capsule deformation is also increased because the dimensionless shear rate indicates the ratio of the fluid viscous force to the elastic resistance of the membrane. The streamlines at the center plane, i.e. $z = 0$, is also plotted in Fig. 12. We can see that the streamlines are conforming to the body surface, similar with the 2D case. In the present

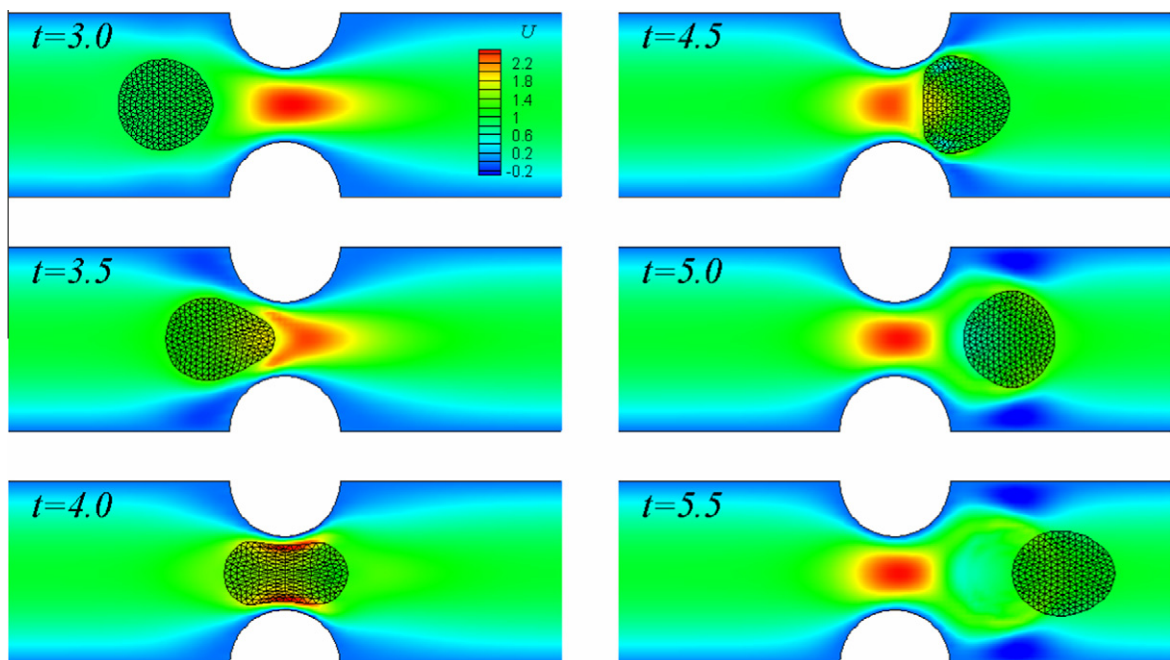


Fig. 10. An elastic circular disk moving through a channel with contraction at $Re = 10$: streamwise velocity and disk positions.

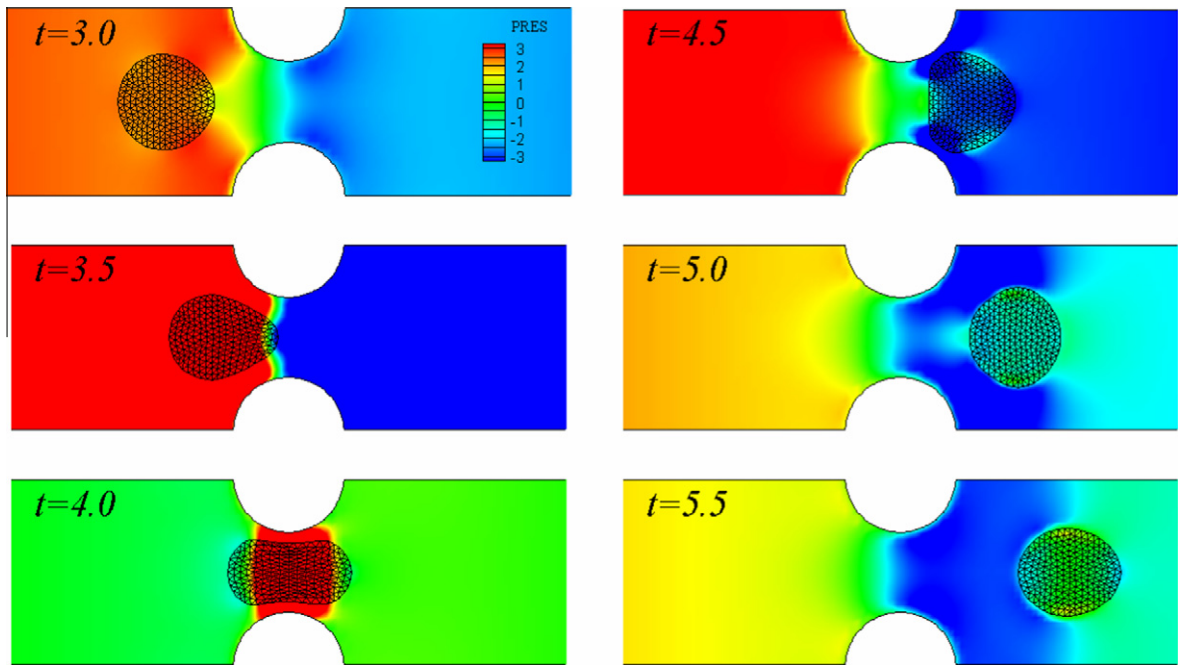


Fig. 11. An elastic circular disk moving through a channel with contraction at $Re = 10$: pressure field and disk positions.

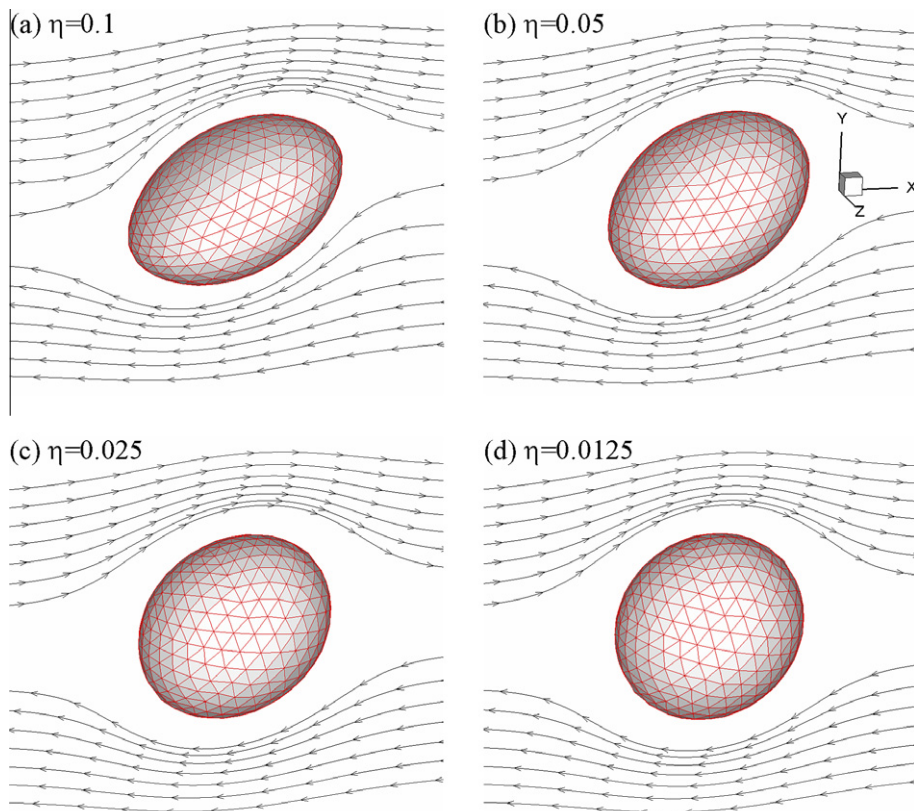


Fig. 12. Deformation of a spherical capsule in shear flow at $Re = 0.025$: (a) $\eta = 0.1$; (b) $\eta = 0.05$; (c) $\eta = 0.025$; (d) $\eta = 0.0125$.

study, the volume change of the capsule is less than 1% at the end of simulation for all the cases. To measure the deformation of the capsule, a deformation parameter is defined as

$$\varepsilon_{xy} = \frac{|L_1 - L_2|}{L_1 + L_2}, \quad (56)$$

where L_1 and L_2 denote the two semi axis lengths of the ellipsoid in the center plane, i.e. $z = 0$. Fig. 13 shows the deformation parameter ε_{xy} as a function of η and compares the two constitutive laws, i.e. the Skalak (SK) law (Eq. (9)) and the neo-Hookean (NH) law (Eq. (11)). Note that the Poisson's ratio is different for the two constitutive laws as implied in Eqs. (9) and (11), so we have $E_s = 2\varphi$ for the SK law and $E_s = 3\varphi$ for the NH law. Here we use the same surface Young's modulus for both constitutive laws in the simulation. In this example, the reason of selecting a low Reynolds number is to compare the present numerical results with the asymptotic solution in the Stokes flow regime. In the limit of the Stokes flow and small dimensionless shear rate, the deformation parameter ε_{xy} can be expressed by [35],

$$\varepsilon_{xy} = \frac{5}{2}(2 + \nu_s)\eta, \quad (57)$$

where the Poisson's ratio $\nu_s = 0$ for the SK law and $\nu_s = 0.5$ for the NH law. From Fig. 13 we can see that the deformation of the SK material is smaller than that of the NH material, while in both cases the numerical results converge to the theoretical predictions (Eq. (56)) as η decreases.

4.3.2. Simulation of a windsock in a uniform flow

In the above simulations of fluid-flexible body interaction problems, the density difference ρ is set to be small. To test the proposed method for problems where the structure mass plays an important role, we simulate the motion of a windsock in a uniform flow, which can be used at airports to make the wind visible to pilots [21]. The computational model is plotted in Fig. 14. The windsock has an initial shape of a truncated cone with $h_c:r_b:r_t = 1:0.2:0.12$, where h_c , r_b and r_t denotes the height

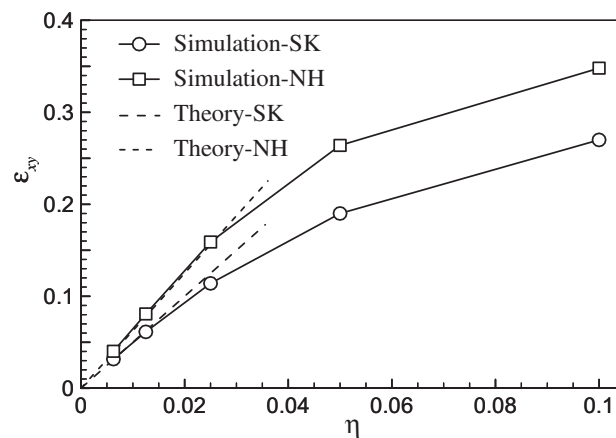


Fig. 13. The deformation parameter ε_{xy} as a function of the dimensionless shear rate: comparison of the SK law and the NH law.

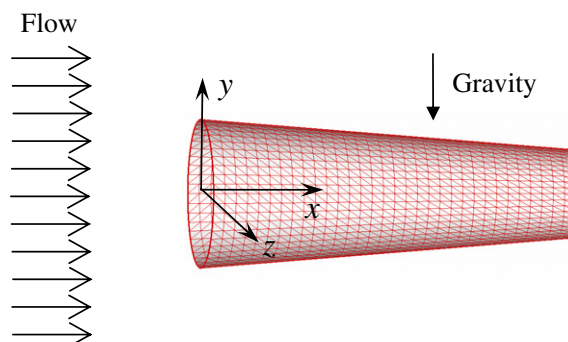


Fig. 14. Computational model of a windsock in a uniform flow.

of the cone, the radius of the circular top, and the radius of the circular bottom, respectively. Here we refer to the height of the cone as the windsock length, which is selected as the characteristic length. Discretization of the windsock is similar to that of a circular cylinder (Fig. 2(c)) and the 4th level quadrissection of the unstructured mesh (1584 nodes and 3072 elements) is used. The fluid domain is a rectangular box and ranges from $(-1, -1, -1)$ to $(3, 1, 1)$ in the x -, y - and z -directions respectively, which is discretized uniformly by $129 \times 65 \times 65$ grids in the three directions. A uniform flow ($u = U_0$, $v = 0$, $w = 0$) is given at the inlet ($x = -1$) and a convective boundary condition is adopted at the outlet ($x = 3$). Dirichlet boundary conditions ($u = U_0$, $v = 0$, $w = 0$) are used at $y = \pm 1$ and a periodic condition is applied in the z -direction. Initially the central axis of the windsock is aligned with the x -axis with the origin of the coordinates located at the center of the circular bottom. After startup of simulation, the windsock moves freely under the influences of both the fluid flow and the gravity force, except that the circumference of the circular bottom is fixed. To take into account the gravity force, an additional force term should be added in the solid motion equations (Eq. (14)), which can be expressed in the dimensionless form as $\rho \text{Fr} g/g$, where $\mathbf{g} = (0, -g, 0)$ denotes the gravity force directing in the negative y -direction and $\text{Fr} = gh_c/U_0^2$ is the Froude number. Fig. 15 shows the instantaneous shape of the windsock together with the fluid velocity vector field at the mid-plane, i.e. $z = 0$, for

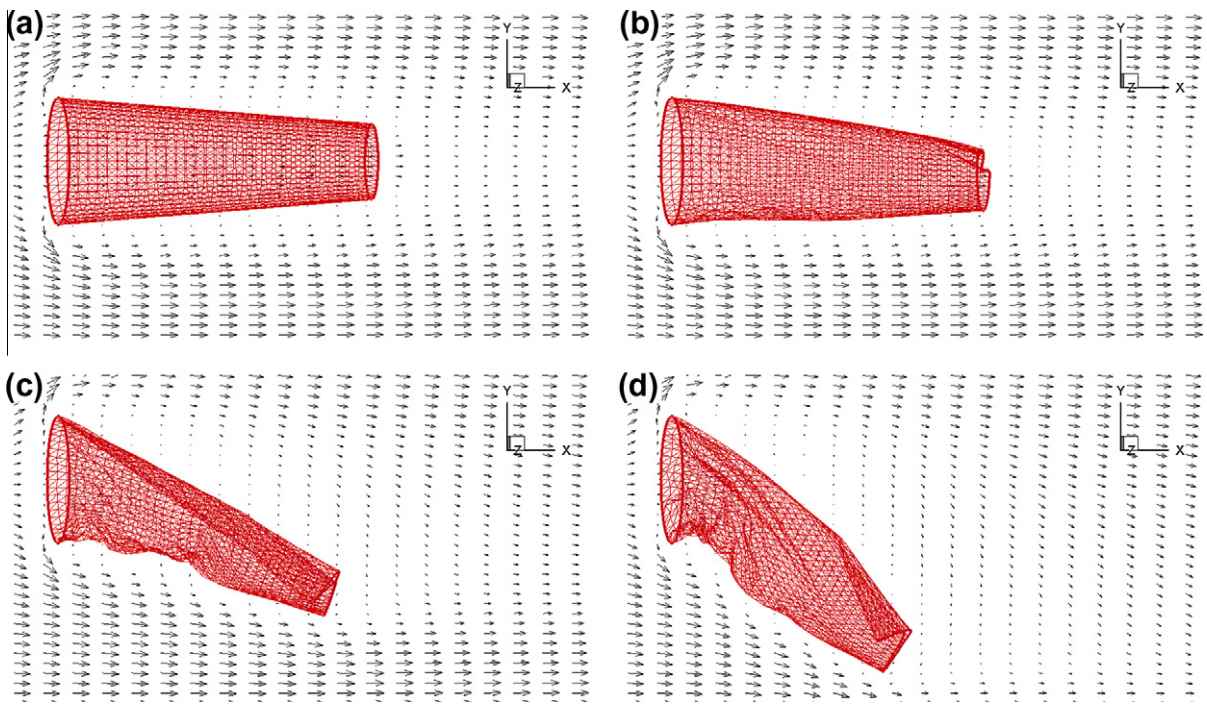


Fig. 15. A windsock in a uniform flow at $Re = 200$ and $Fr = 1.0$: (a) $\rho = 0.1$; (b) $\rho = 0.5$; (c) $\rho = 0.8$; (d) $\rho = 1.0$.

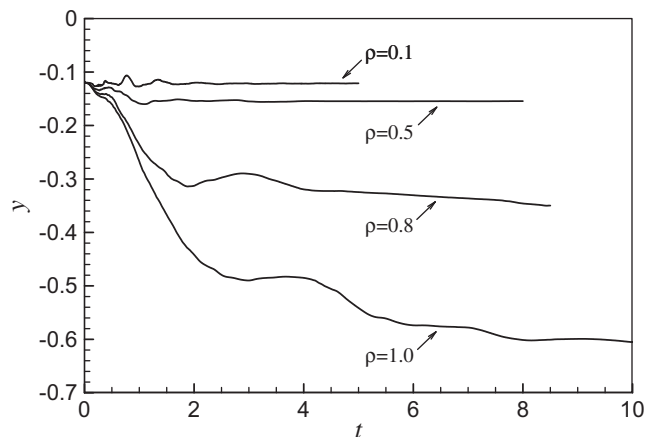


Fig. 16. Time history of the vertical position of the lowest point of the windsock outlet.

$\rho = 0.1, 0.5, 0.8$ and 1.0 . Other parameter values are set to be $Re = 200$, $Fr = 1.0$, $E_s = 500$ and $\lambda = 0$. For $\rho = 0.1$ the windsock behaves like neutrally buoyant material, while for $\rho = 0.5$ it sags down slightly due to the gravity force. It is interesting to see that, however, the windsock sags down obviously for $\rho = 0.8$ and 1.0 . The sagging distance can be measured quantitatively from the time history of the vertical position of the lowest point of the windsock outlet, as shown in Fig. 16, where we can see the sagging distance is increased significantly for $\rho > 0.5$. A close look at the windsock surface shows that buckling occurs at the lower part of the windsock for $\rho = 0.8$ and 1.0 , as seen in Fig. 15(c) and (d), which is caused by the compressive stresses of the elastic membrane. It should be pointed out that the bending force is not taken into account in the present formulation of the elastic membrane, thus the buckling phenomena due to the compressive stresses cannot be accurately captured, which is a problem to be solved in our future work.

5. Conclusions

In the present study, we made a revision on the original pIB method [21] aiming at simulating the fluid–flexible body interaction problems. In our method, the elastic force of solid deformation is exerted on the massive material points instead of the massless material points as in [21]. This simple change brought significant convenience in coupling the flow solver and the solid solver, which were developed in different methods. In this sense, the present method provides a general and effective framework for the fluid–flexible body interaction simulations. We adopted the fractional step method and the staggered Cartesian grid system to solve the incompressible fluid motion. The quadrissection method was used for generation of the unstructured triangular mesh for solid geometries, and the finite element method was developed to calculate the solid motion. The solid solver was validated by using a vibrating circular ring in vacuum. Good agreements were obtained between the numerical results and the theoretical solutions, and the spatial accuracy was seen to be second-order. Then two examples of 2D fluid–flexible body interaction were simulated, i.e. deformation of a circular disk in a linear shear flow and an elastic circular disk moving through a constricted channel. The spatial accuracy was close to second order for both the fluid velocities and the solid positions in the L_2 norm, but was lower in the L_∞ norm. Finally, 3D simulations of fluid–membrane interaction were performed. Deformation of a spherical capsule in a linear shear flow was presented at different dimensionless shear rates, and the SK law and the NH law were compared. The deformation parameter of the capsule was shown to converge to the theoretical solution for both constitutive laws in the limit of the Stokes flow and low dimensionless shear rate. We also simulated a windsock in a uniform flow, where the structure mass plays an important role. It was shown that the surrounding fluid flow tends to keep the windsock straight, while the gravity force causes it to sag down. The sagging distance is small for light structures but is increased significantly for heavy structures, due to the occurrence of buckling of the membrane under the compressive stresses. It should be pointed out that in the present method we did not consider the neutrally buoyant case ($\rho = 0$), which makes the solid solver not applicable. However, in this case the momentum forcing can be calculated directly from Eq. (14) and be substituted into the N–S equations after transformation, so that the present method becomes equivalent to that of Peskin [19]. Moreover, in the framework of the present method, we can consider more complicated solid motions by upgrading the solid solver, e.g. a 3D membrane with bending rigidity and with other constitutive laws, which is coupled to the fluid motion in the same way. This is the work we are currently undertaking.

Acknowledgment

This study was supported by the Creative Research Initiatives (No. 2010-000433) of the National Research Foundation of Korea. HWX also acknowledges the support from the Natural Science Foundation of China (No. 11002081).

References

- [1] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (2005) 239–261.
- [2] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* 161 (2000) 35–60.
- [3] J. Kim, D. Kim, H. Choi, An immersed boundary finite-volume method for simulations of flow in complex geometries, *J. Comput. Phys.* 171 (2001) 132–150.
- [4] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* 192 (2003) 593–623.
- [5] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys.* 105 (1993) 354–366.
- [6] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, *J. Comput. Phys.* 123 (1996) 450–465.
- [7] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2000) 705–719.
- [8] K. Taira, T. Colonius, The immersed boundary method: a projection approach, *J. Comput. Phys.* 225 (2007) 2118–2137.
- [9] S.W. Su, M.C. Lai, C.A. Lin, An immersed boundary technique for simulating complex flows with rigid boundary, *Comput. Fluids* 36 (2007) 313–324.
- [10] D.V. Le, B.C. Khoo, K.M. Lim, An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains, *Comput. Method Appl. Mech. Eng.* 197 (2008) 2119–2130.
- [11] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, *J. Comput. Phys.* 215 (2006) 12–40.
- [12] R. Mittal, H. Dong, M. Bozkurtas, F.M. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.* 227 (2008) 4825–4852.
- [13] D. Kim, H. Choi, Immersed boundary method for flow around an arbitrarily moving body, *J. Comput. Phys.* 212 (2006) 662–680.
- [14] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2006) 448–476.

- [15] S.J. Shin, W.-X. Huang, H.J. Sung, Assessment of regularized delta functions and feedback forcing schemes for an immersed boundary method, *Int. J. Numer. Methods Fluids* 58 (2008) 263–286.
- [16] J. Yang, S. Preidikman, E. Balaras, A strongly coupled, embedded-boundary method for fluid–structure interactions of elastically mounted rigid bodies, *J. Fluids Struct.* 24 (2008) 167–182.
- [17] Z. Wang, J. Fan, K. Luo, Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles, *Int. J. Multiphase Flow* 34 (2008) 283–302.
- [18] C.S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* 25 (1977) 220–252.
- [19] C.S. Peskin, The immersed boundary method, *Acta Numer.* (2002) 479–517.
- [20] L. Zhu, C.S. Peskin, Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method, *J. Comput. Phys.* 179 (2002) 452–468.
- [21] Y. Kim, C.S. Peskin, Penalty immersed boundary method for an elastic boundary with mass, *Phys. Fluids* 19 (2007) 053103.
- [22] Y. Mori, S.C. Peskin, Implicit second-order immersed boundary methods with boundary mass, *Comput. Methods Appl. Mech. Eng.* 197 (2008) 2049–2067.
- [23] H. Luo, R. Mittal, X. Zheng, S.A. Bielałowicz, R.J. Walsh, J.K. Hahn, An immersed-boundary method for flow–structure interaction in biological systems with application to phonation, *J. Comput. Phys.* 227 (2008) 9303–9332.
- [24] A. Gilmanov, S. Acharya, A hybrid immersed boundary and material point method for simulating 3D fluid–structure interaction problems, *Int. J. Numer. Methods Fluids* 56 (2008) 2151–2177.
- [25] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, *J. Comput. Phys.* 207 (2005) 457–492.
- [26] H. Zhao, J.B. Freund, R.D. Moser, A fixed-mesh method for incompressible flow–structure systems with finite solid deformations, *J. Comput. Phys.* 227 (2008) 3114–3140.
- [27] D.V. Le, J. White, J. Peraire, K.M. Lim, B.C. Khoo, An implicit immersed boundary method for three-dimensional fluid–membrane interactions, *J. Comput. Phys.* 228 (2009) 8427–8445.
- [28] D.V. Le, Z. Tan, Large deformation of liquid capsules enclosed by thin shells immersed in the fluid, *J. Comput. Phys.* 229 (2010) 4097–4116.
- [29] Y.C. Fung, P. Tong, *Classical and Computational Solid Mechanics*, World Scientific Publishing Co., Singapore, 2001.
- [30] R. Skalak, A. Tozeren, R.P. Zarda, S. Chien, Strain energy function of red blood cell membranes, *Biophys. J.* 13 (1973) 245–264.
- [31] D. Barthes-Biesel, A. Diaz, E. Dhenin, Effect of constitutive law for two dimensional membranes on flow-induced capsule deformation, *J. Fluid Mech.* 460 (2002) 211–222.
- [32] K. Kim, S.-J. Baek, H.J. Sung, An implicit velocity decoupling procedure for incompressible Navier–Stokes equations, *Int. J. Numer. Methods Fluids* 38 (2002) 125–138.
- [33] Y.C. Fung, *Foundations of Solid Mechanics*, Prentice-Hall, Englewood Cliffs, NJ, 1965.
- [34] S. Timoshenko, D.H. Young, W. Weaver, *Vibration Problems in Engineering*, fourth ed., Wiley, New York, 1974.
- [35] E. Lac, D. Barthes-Biesel, N.A. Pelekasis, J. Tsamopoulos, Spherical capsules in three-dimensional unbounded Stokes flows: effect of the membrane constitutive law and onset of buckling, *J. Fluid Mech.* 516 (2004) 303–334.