Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 217 (2006) 806–823
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

# A fluid–structure interaction method with solid-rigid contact for heart valve dynamics

R. van Loon [a,1], P.D. Anderson [b], F.N. van de Vosse [a,*]

[a] Department of Biomedical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[b] Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Received 23 May 2005; received in revised form 20 January 2006; accepted 22 January 2006
Available online 3 March 2006

**Abstract**

A computational method is proposed for problems where fluid–structure interaction is combined with solid-rigid contact. This combination is particularly important for the dynamics and impact of heart valves. The Navier–Stokes equation in an Eulerian setting is coupled to a Neo-Hookean solid model using a Lagrangian description. A fictitious domain method extended with a local mesh adaptation algorithm provides the required flexibility with respect to the motion and deformation of the valve. In addition, it ensures the solids ability of sustaining pressures present in the fluid. Making use of the fact that the fluid and solid mesh are not required to be connected conformingly, it is shown that the model can be extended with a contact algorithm without introducing meshing complications near the contact surfaces.


*Keywords:* Heart valves; Lagrange multipliers; Adaptive meshing; Solid-rigid contact; Fluid–structure interaction; Fictitious domains

## 1. Introduction

Fluid–structure interaction models are increasingly used in biomedical engineering applications and one of the most challenging fluid–structure problems that can be found in the human body involves the dynamics of heart valves. The most extensively studied valves are the mitral and the aortic valve. The former is a bileaflet valve located between the left atrium and left ventricle, the latter is a trileaflet and is located between the left ventricle and the aortic root. Both valves have extremely thin leaflets, which should hamper flow as little possible when opened, but need to prevent blood back-flow (regurgitation) if closed. In the closed state the leaflets are in mutual contact and a large transvalvular pressure gradient will occur. The tissue of which the heart valves are built, has an anisotropic and heterogeneous structure. The arterial walls and heart muscle are compliant and therefore play an important role in the process of opening and closing. Altogether, the combination makes the problem extremely complex to model.

* Corresponding author.
*E-mail addresses:* r.v.loon@imperial.ac.uk (R. van Loon), F.N.v.d.Vosse@tue.nl (F.N. van de Vosse).
[1] Present address: Imperial College London, Department of Aeronautics, South Kensington Campus, London SW7 2AZ, United Kingdom. Tel.: +44 2075945129.

0021-9991/$ - see front matter 
doi:10.1016/j.jcp.2006.01.032

Different ways of modelling fluid–structure interaction (FSI) have been proposed in the past, each having its advantages and disadvantages. Arbitrary Lagrangian Eulerian (ALE) methods, as exploited by e.g. [1,2], are most commonly used for FSI problems and have the advantage to provide a strong coupling. As long as rotations, translations and/or deformations of the solid remain within certain limits, this method works very well and is recommended. However, for problems in which these limits are violated, elements become ill-shaped and ALE alone does not suffice. As a solution to this problem an often-used combination is ALE with some form of remeshing. This can, however, be a difficult and time consuming task.

A more elegant way to solve the system allowing free movements of a structure through a fluid domain was proposed by Peskin [3]. He introduced a method that later became known as the immersed boundary method (IBM) [4] where flow-induced solid body motions could be computed without adjusting the fluid grid/mesh. By defining a set of interconnected points related to each other by some elastic law local body forces were enforced to the fluid. Extensions of this model to three dimensional heart (valve) problems were published in e.g. [5,6] and the method is still used in many fields.

A similar method that could cope with these large translations and rotations was used by Glowinski et al. [7–9]. Using a fictitious domain method the sedimentation of rigid particles in a fluid domain could be computed. The rigid particles are immersed on the fluid domain and are coupled to the fluid by applying constraints across the particle using a Lagrange multiplier. This way of coupling allowed for large translations as well as rotations of the particles.

A method that resembles the above mentioned methods was introduced [11,10] for slender deforming bodies. In this method a fluid mesh is considered with an immersed solid mesh, and the solid mesh and fluid mesh are coupled by a Lagrange multiplier (or local body forces) at the boundary of the solid. With respect to heart valve modelling good results have been obtained in 2D [11] as well as in 3D [12–14]. The elegance of this method is its simplicity and flexibility. Stijnen et al. [17] introduced a model for mechanical heart valves using this fictitious domain method. The model was capable of computing a full cardiac cycle, using the fact that the position of a closed mechanical heart valve is known a priori. By creating the fluid mesh such that a curve of fluid edges coincided with the solid boundary in the closed state, the occurring drop in pressure across the valve could be described. Recently, fictitious domain methods have been proposed that are not restricted to slender bodies by introducing Lagrange multipliers across the whole solid body instead of only along its boundaries [15,16]. This enables the computation of a pressure drop across the solid body without alignment of the meshes and such makes the method suitable for a wider range of applications.

The fictitious domain approaches described above have not proven capable of computing shear stresses along the solid boundaries. Shear stress has been related to blood damage and the upregulation of the endothelial cells that line the entire cardiovascular system and are therefore a parameter of interest in this paper. A combination of the fictitious domain method with adaptive meshing was proposed by van Loon et al. [18] to compute moving pressure discontinuities across a flexible leaflet without an a priori known closed position. It was also shown that the method was able to compute shear stresses quantitatively at both sides of the solid. An extension of this method to three dimensions has also been published recently by van Loon et al. [19] together with an improved adaptive meshing scheme. Although some of the flexibility of the original fictitious domain methods is lost by the introduction of adaptive meshing, accuracy with respect to shear stress is gained. Note, however, that different discretisations for the fluid and the solid are still allowed.

This paper focusses on a fluid–structure interaction method as the one proposed by van Loon et al. [18] extending it with a Lagrange multiplier based contact algorithm which enables interaction of the solid with walls of a fluid domain. The adaptive meshing scheme is a two-dimensional version of the one presented in [19]. The combination of the fictitious domain method with other established techniques like adaptive meshing and ALE methods allows not only for accurate motion of the solid body but also accurate fluid behaviour near the solid–fluid interface. The presented method is able to capture the opening and closing behaviour of a heart valve in a model problem at sufficient accuracy, that it can be used for heart valve analysis during a complete cardiac cycle if extended to three dimensions.

The paper is organised as follows. First, the governing equations are given, treating the equations for solid and fluid, followed by the constraints concerning the fluid–structure interaction and contact. Then the full set of equations is given in the weak form, after which the discretisation and adaptive meshing are treated. All techniques are combined in two illustrative finite element model problems, describing thin flexible solid bodies

in a pulsatile flow, where the solid can make contact with the wall. Pressure fields, shear stresses and solid motion are observed closely.

## 2. Governing equations

### 2.1. Fluid and solid

First, the governing equations that describe an incompressible Newtonian fluid and an incompressible Neo-Hookean solid, are presented,. The set of equations for the fluid domain, denoted by $\Omega^{\mathrm{f}}$, read

$$\rho\left(\frac{\partial \boldsymbol{u}^{\mathrm{f}}}{\partial t}+\boldsymbol{u}^{\mathrm{f}}\cdot\nabla\boldsymbol{u}^{\mathrm{f}}\right)=\nabla\cdot\boldsymbol{\sigma}^{\mathrm{f}}+\rho\boldsymbol{f}^{\mathrm{f}}, \tag{1}$$

$$\nabla\cdot\boldsymbol{u}^{\mathrm{f}}=0, \tag{2}$$

$$\boldsymbol{\sigma}^{\mathrm{f}}=2\eta\boldsymbol{D}-p^{\mathrm{f}}\boldsymbol{I}\quad\text{with}\quad\boldsymbol{D}=\frac{1}{2}\left(\nabla\boldsymbol{u}^{\mathrm{f}}+(\nabla\boldsymbol{u}^{\mathrm{f}})^{\mathrm{T}}\right), \tag{3}$$

and for the solid domain, $\Omega^{\mathrm{s}}$

$$\nabla\cdot\boldsymbol{\sigma}^{\mathrm{s}}=\rho\boldsymbol{f}^{\mathrm{s}}, \tag{4}$$

$$\det(\boldsymbol{F})=1,\quad\text{with}\quad\boldsymbol{F}=(\nabla_0\boldsymbol{x})^{\mathrm{T}}, \tag{5}$$

$$\boldsymbol{\sigma}^{\mathrm{s}}=G(\boldsymbol{B}-\boldsymbol{I})-p^{s}\boldsymbol{I}\quad\text{with}\quad\boldsymbol{B}=\boldsymbol{F}\cdot\boldsymbol{F}^{\mathrm{T}}. \tag{6}$$

Eqs. (1)–(3) and (4)–(6) are the momentum balance, the continuity and the constitutive relation describing the rheological behaviour of the fluid and solid, respectively. The quantities $\rho$, $\eta$ and $G$ are the density, dynamic viscosity and shear modulus. The gradient operators with respect to the current and the initial configuration are denoted with $\nabla$ and $\nabla_0$, respectively. The vectors $\boldsymbol{x}$ and $\boldsymbol{u}$ denote the position and velocity vector, respectively, and $p$ the pressure. The superscripts 'f' and 's' will be used in this paper to distinguish fluid and solid, respectively. Body forces $\boldsymbol{f}^{\mathrm{s}}$ and $\boldsymbol{f}^{\mathrm{f}}$ are zero and gravity forces are negligible since a heart valve is almost neutrally buoyant.

### 2.2. Fluid–structure coupling

The fluid domain $\Omega^{\mathrm{f}}$ with an embedded solid domain $\Omega^{\mathrm{s}}$ is considered. In order to capture the fluid–structure interaction, these two domains need to be coupled. This coupling is obtained by applying a no-slip condition

$$\boldsymbol{u}^{\mathrm{f}}-\boldsymbol{u}^{\mathrm{s}}=0, \tag{7}$$

at the boundary of the solid $\partial\Omega^{\mathrm{s}}$. Note that the partial sign $\partial$ will be used to denote the boundary of a domain.

### 2.3. Contact

Modelling the interaction of a deformable solid with a rigid contact surface is not very different from modelling the interaction between a fluid and a structure. The main difference lies in the fact that contact is a temporary state while it is assumed that the fluid and structure interact permanently. If a solid approaches a contact surface nothing will happen until the gap between them approaches zero. From that moment a contact force, $\boldsymbol{\lambda}^{\mathrm{c}}$, between the bodies will be induced which inhibits the solid body from penetrating the contact surface and contact occurs until this contact force equals zero again. If we describe the contact surface as a function, $g(\boldsymbol{x},t)=0$, then for every point at the solid boundary, $\forall\boldsymbol{x}^{\mathrm{s}}\in\partial\Omega^{\mathrm{s}}$, a gap distance can be defined as $\delta^{\mathrm{c}}=(\boldsymbol{x}^{\mathrm{c}}-\boldsymbol{x}^{\mathrm{s}})\cdot\boldsymbol{n}^{\mathrm{c}}$, where $\boldsymbol{x}^{\mathrm{c}}$ is a position at the contact surface ($g(\boldsymbol{x}^{\mathrm{c}},t)=0$) and $\boldsymbol{n}^{\mathrm{c}}$ the corresponding normal vector with respect to this body in the direction of the solid body. For every $\boldsymbol{x}^{\mathrm{s}}$ a point $\boldsymbol{x}^{\mathrm{c}}$ is chosen such that the distance, $\min(\|\boldsymbol{x}^{\mathrm{c}}-\boldsymbol{x}^{\mathrm{s}}\|)$, between them is minimized with $\|\boldsymbol{x}\|=\sqrt{\boldsymbol{x}\cdot\boldsymbol{x}}$. A second important quantity with respect to the contact problem is the contact force, which is defined as $\boldsymbol{\lambda}^{\mathrm{c}}=\lambda^{\mathrm{c}}\cdot\boldsymbol{n}^{\mathrm{c}}$. The constraints corresponding to the different contact states read in the case of "contact"

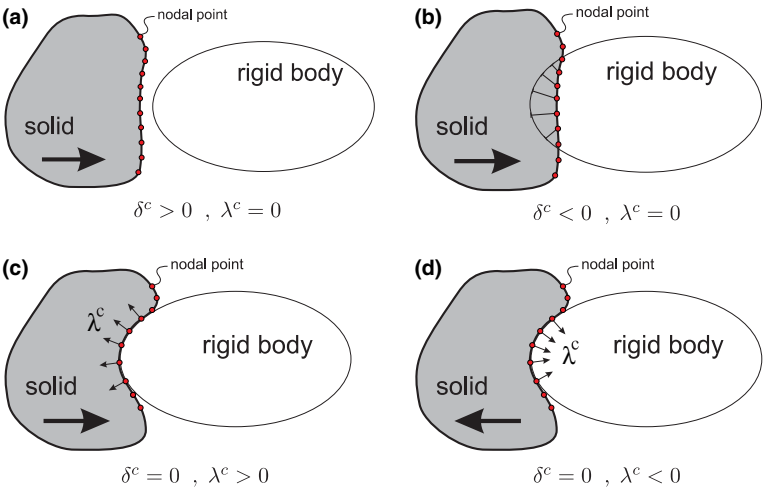


Fig. 1. Different stages of the contact algorithm.

$$\delta^{\mathrm{c}} = 0, \tag{8}$$

and in the case of "no-contact"

$$\lambda^{\mathrm{c}} = 0. \tag{9}$$

The moment that the solid body penetrates the rigid surface (Fig. 1(b)), the gap distance will be forced to zero (Eq. (8)) and contact is established (Fig. 1(c)). When the contact force becomes negative (Fig. 1(d)) the solid body should be released from the contact surface and the trivial "no-contact" constraint (Eq. (9)) is applied. When a node is in contact the switch of constraints is triggered by evaluating the contact pressure and if a node is not in contact the switch is triggered by looking at the gap distance. Note that frictionless contact is assumed since Eq. (8) only restricts the displacement in normal direction.

## 3. Weak formulation

Based on the governing set of equations and the set of constraints to which they should comply as defined in the previous sections, the weak formulation for the total set of equations will be derived.

If we define the Lebesgue space $L_2(\Omega)$ and Sobolev space $H^{\kappa}(\Omega)$ as

$$L_2(\Omega) = \left\{ v : \int_{\Omega} v^2 \,\mathrm{d}\Omega < \infty \right\},$$
$$H^{\kappa}(\Omega) = \left\{ v \in L_2(\Omega) : D^{\beta} v \in L_2(\Omega), |\beta| \leqslant \kappa \right\},$$

with $D^{\beta} v$ the weak derivatives of order $\beta$, spaces of acceptable solutions for the solid and fluid velocities can be defined as

$$\mathbb{V}^{\alpha}(\Omega^{\alpha}) = \left\{ \boldsymbol{v} : \boldsymbol{v} \in H^1(\Omega^{\alpha})^2, \boldsymbol{v} = \boldsymbol{u}_0^{\alpha} \quad \text{on} \quad \partial\Omega^{\alpha} \right\} \quad \text{for} \quad \alpha = \mathrm{f}, \mathrm{s}.$$

The corresponding trial functions should lie in the space

$$\mathbb{V}_0^{\alpha}(\Omega^{\alpha}) = \left\{ \mathbb{V}(\Omega^{\alpha}) : \boldsymbol{u}_0^{\alpha} = 0 \right\} \quad \text{for} \quad \alpha = \mathrm{f}, \mathrm{s}.$$

The kinematic constraints given by Eqs. (7)–(9) are enforced by means of the two Lagrange multipliers, $\boldsymbol{\lambda}^{\mathrm{fsi}}$ and $\boldsymbol{\lambda}^{\mathrm{c}}$. Both multipliers are typically defined along the boundary of the solid domain $\partial\Omega^{\mathrm{s}}$, but depending on the problem alternative domains can be chosen as will be done later in this paper. In order to distinguish the two Lagrange multiplier domains, $\gamma^{\mathrm{f}}$ and $\gamma^{\mathrm{c}}$ are introduced to denote the domain corresponding to the fluid–structure interaction and the solid contact surface. By using the momentum balance for fluid and solid, Eqs. (1) and (4), including the multipliers and applying all the constraints equations (2), (5) and (7)–(9), the

weak formulation for the total set of equations can be formed. Find $\boldsymbol{u}^{\mathrm{f}} \in \mathbb{V}^{\mathrm{f}}(\Omega^{\mathrm{f}})$, $\boldsymbol{u}^{\mathrm{s}} \in \mathbb{V}^{\mathrm{s}}(\Omega^{\mathrm{s}})$, $p^{\mathrm{s}} \in L_2(\Omega^{\mathrm{s}})$, $p^{\mathrm{f}} \in L_2(\Omega^{\mathrm{f}})$, $\boldsymbol{\lambda}^{\mathrm{fsi}} \in L_2(\gamma^{\mathrm{f}})^2$ and $\boldsymbol{\lambda}^{\mathrm{c}} \in L_2(\gamma^{\mathrm{c}})^2$ such that

$$\int_{\Omega^{\mathrm{f}}} \boldsymbol{w}^{\mathrm{f}} \cdot \left( \rho \frac{\partial \boldsymbol{u}^{\mathrm{f}}}{\partial t} + \rho \boldsymbol{u}^{\mathrm{f}} \cdot \nabla \boldsymbol{u}^{\mathrm{f}} \right) \mathrm{d}\Omega^{\mathrm{f}} - \int_{\Omega^{\mathrm{f}}} 2\eta \boldsymbol{D}(\boldsymbol{u}^{\mathrm{f}}) : \boldsymbol{D}(\boldsymbol{w}^{\mathrm{f}}) \, \mathrm{d}\Omega^{\mathrm{f}} + \int_{\Omega^{\mathrm{f}}} p^{\mathrm{f}} \nabla \cdot \boldsymbol{w}^{\mathrm{f}} \, \mathrm{d}\Omega^{\mathrm{f}} + \int_{\gamma^{\mathrm{f}}} \boldsymbol{w}^{\mathrm{f}} \cdot \boldsymbol{\lambda}^{\mathrm{fsi}} \, \mathrm{d}\gamma^{\mathrm{f}} = \int_{\partial\Omega^{\mathrm{f}}} \boldsymbol{w}^{\mathrm{f}} \cdot (\boldsymbol{\sigma}^{\mathrm{f}} \cdot \boldsymbol{n}^{\mathrm{f}}) \, \mathrm{d}\partial\Omega^{\mathrm{f}}, \tag{10}$$

$$\int_{\Omega^{\mathrm{f}}} q^{\mathrm{f}} \nabla \cdot \boldsymbol{u}^{\mathrm{f}} \, \mathrm{d}\Omega^{\mathrm{f}} = 0, \tag{11}$$

$$\int_{\Omega^{\mathrm{s}}} \boldsymbol{w}^{\mathrm{s}} \cdot \nabla \cdot \boldsymbol{\tau}^{\mathrm{s}} \, \mathrm{d}\Omega^{\mathrm{s}} - \int_{\Omega^{\mathrm{s}}} p^{\mathrm{s}} \nabla \cdot \boldsymbol{w}^{\mathrm{s}} \, \mathrm{d}\Omega^{\mathrm{s}} - \int_{\gamma^{\mathrm{f}}} \boldsymbol{w}^{\mathrm{s}} \cdot \boldsymbol{\lambda}^{\mathrm{fsi}} \, \mathrm{d}\gamma^{\mathrm{f}} - \int_{\gamma^{\mathrm{c}}} \boldsymbol{w}^{\mathrm{s}} \cdot \boldsymbol{\lambda}^{\mathrm{c}} \, \mathrm{d}\gamma^{\mathrm{c}} = \int_{\partial\Omega^{\mathrm{s}}} \boldsymbol{w}^{\mathrm{s}} \cdot (\boldsymbol{\sigma}^{\mathrm{s}} \cdot \boldsymbol{n}^{\mathrm{s}}) \, \mathrm{d}\partial\Omega^{\mathrm{s}}, \tag{12}$$

$$\int_{\Omega^{\mathrm{s}}} q^{\mathrm{s}} (\det(\boldsymbol{F}) - 1) \, \mathrm{d}\Omega^{\mathrm{s}} = 0, \tag{13}$$

$$\int_{\gamma^{\mathrm{f}}} \boldsymbol{w}^{\mathrm{fsi}} \cdot (\boldsymbol{u}^{\mathrm{f}} - \boldsymbol{u}^{\mathrm{s}}) \, \mathrm{d}\gamma^{\mathrm{f}} = 0, \tag{14}$$

$$\int_{\gamma^{\mathrm{c}}} w^{\mathrm{c}} h \, \mathrm{d}\gamma^{\mathrm{c}} = 0 \quad \text{with} \quad h = \begin{cases} \delta^{\mathrm{c}} & \text{for} \quad \lambda^{\mathrm{c}} > 0, \ \delta^{\mathrm{c}} \leqslant 0, \\ \lambda^{\mathrm{c}} & \text{for} \quad \lambda^{\mathrm{c}} \leqslant 0, \ \delta^{\mathrm{c}} \geqslant 0, \end{cases} \tag{15}$$

for all trial functions $\boldsymbol{w}^{\mathrm{f}} \in \mathbb{V}_0^{\mathrm{f}}(\Omega^{\mathrm{f}})$, $\boldsymbol{w}^{\mathrm{s}} \in \mathbb{V}_0^{\mathrm{s}}(\Omega^{\mathrm{s}})$, $q^{\mathrm{f}} \in L_2(\Omega^{\mathrm{f}})$, $q^{\mathrm{s}} \in L_2(\Omega^{\mathrm{s}})$, $\boldsymbol{w}^{\mathrm{fsi}} \in L_2(\gamma^{\mathrm{f}})^2$ and $w^{\mathrm{c}} \in L_2(\gamma^{\mathrm{c}})$. The normals at the solid boundary $\partial\Omega^{\mathrm{s}}$ and the fluid boundary $\partial\Omega^{\mathrm{f}}$ are denoted by $\boldsymbol{n}^{\mathrm{s}}$ and $\boldsymbol{n}^{\mathrm{f}}$, respectively. Note that the kinematic constraint with respect to fluid–structure interaction is applied in a "weak" manner, meaning that the integrals over the fluid and solid velocities are forced to be equal along boundary $\gamma^{\mathrm{f}}$. Similarly, the kinematic constraint for the contact problem is applied. As $\boldsymbol{\lambda}^{\mathrm{c}}$ represents the contact force that acts between the solid and rigid body in the case of contact, so the physical representation of $\boldsymbol{\lambda}^{\mathrm{fsi}}$ is the traction force between solid and fluid. These "additional body forces" are applied to the weak formulations of the balance of momentum for fluid and solid, Eqs. (10) and (12), in order to satisfy the kinematic constraints. Finally, note that Eqs. (10) and (12) contain non-linear parts, and should therefore be linearised.

## 4. Discretisation

Finite-element discretisations are applied for the fluid and solid domain and for the Lagrange multipliers corresponding to fluid–structure coupling and the contact law. First, we focus on the discretisations of the fluid and solid domain and the corresponding adaptive meshing scheme. Next, the discretisations of the Lagrange multipliers corresponding to the fluid–structure coupling and solid-rigid contact will be discussed, the first of which depends on the fluid discretisation and the latter on the solid discretisation.

### 4.1. Discretisation of fluid and solid

For convenience, first some definitions concerning a finite element mesh are posed. Therefore, we consider a domain $\Omega \subset \mathbb{R}^2$ and its boundary $\partial\Omega \subset \mathbb{R}^2$. If $\Omega$ is subdivided into $N_\Omega$ elements $\Omega_e$ with $e = 1, \ldots, N_\Omega$, the following relations can be defined with respect to the elements $\Omega_e$, the elemental edges $\Gamma_j$ and elemental vertex points $\boldsymbol{x}_k$

$$\overline{\Omega} = \bigcup_{e=1}^{N_\Omega} \overline{\Omega}_e \quad \text{and} \quad {}^{\Omega}S = \{ e \in \mathbb{N} : \Omega_{e \neq i} \cap \Omega_i = \emptyset \} \quad \forall i \in \mathbb{N},$$

$$\partial\Omega_e = \bigcup_{j=1}^{N_\Gamma} {}_e\overline{\Gamma}_j \quad \text{and} \quad {}^{\Gamma}S = \{ k \in \mathbb{N} : \Gamma_k = {}_e\Gamma_j, \Gamma_{k \neq l} \cap \Gamma_l = \emptyset \} \quad \forall {}_e\Gamma_j, \forall l \in \mathbb{N},$$

$$\partial\Gamma_j = \bigcup_{i=1}^{2} {}_j\boldsymbol{x}_i \quad \text{and} \quad {}^{x}S = \{ k \in \mathbb{N} : \boldsymbol{x}_k = {}_j\boldsymbol{x}_i, \boldsymbol{x}_{k \neq l} \neq \boldsymbol{x}_l \} \quad \forall {}_j\boldsymbol{x}_i, \forall l \in \mathbb{N},$$

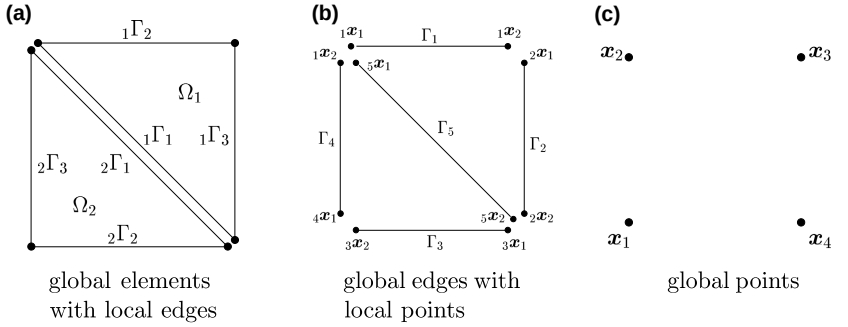


Fig. 2. Topological definitions.

where $N_\Gamma$ is the number of local edges ${}_e\Gamma_j$ in element $\Omega_e$. A graphical representation is presented in Fig. 2. Note that local edges are denoted as ${}_i\Gamma_j$ and global edges as $\Gamma_j$ and similarly ${}_j\boldsymbol{x}_i$ and $\boldsymbol{x}_i$ are local and global points. The sets ${}^{\Gamma}S$ and ${}^{x}S$ contain the numbers of all elemental edges and nodal points of the element vertices in $\Omega$, respectively. Every edge and vertex point in the mesh is unique. Now the domains $\Omega^{\mathrm{f}} \subset \mathbb{R}^2$ and $\Omega^{\mathrm{s}} \subset \mathbb{R}^2$ in the initial configuration can be discretised independently of one another to obtain the non-overlapping, conforming meshes ${}^{\Omega}S^{\mathrm{f}}$ and ${}^{\Omega}S^{\mathrm{s}}$, respectively. Since these meshes are created independently, they will be non-conforming with respect to each other and overlapping. Mesh ${}^{\Omega}S^{\mathrm{f}}$ is a triangulation while ${}^{\Omega}S^{\mathrm{s}}$ consists of quadrilaterals.

*4.1.1. Adaptive meshing*

An adaptive meshing procedure is now introduced for adapting the fluid mesh based on the position of the solid mesh, which is similar to the one proposed by Lock et al. [20]. Although the shape of the solid elements can be chosen arbitrarily, the element shape of the fluid elements is restricted to a triangular shape for this meshing procedure. The fluid mesh will be adapted such that an inner fluid curve $\gamma^{\mathrm{f}} \subset \Omega^{\mathrm{f}}$ is created based on the position of a solid boundary $\gamma^{\mathrm{s}} \subset \Omega^{\mathrm{s}}$. Along this boundary the solid and fluid mesh will be coupled using Eq. (14). Unlike the adaptive meshing scheme proposed earlier [18], local topological changes are permitted. We first consider the solid domain $\Omega^{\mathrm{s}}$ and fluid domain $\Omega^{\mathrm{f}}$ such that $\overline{\Omega}^{\mathrm{s}} \cap \overline{\Omega}^{\mathrm{f}} \neq \emptyset$. A curve $\gamma^{\mathrm{s}}$ needs to be chosen as a collection of $N^{\mathrm{s}}$ elemental edges of the solid along which the fluid–structure coupling should be established:

$$\gamma^{\mathrm{s}} = \bigcup_{i=1}^{N^{\mathrm{s}}} \overline{\Gamma}_i^{\mathrm{s}} \quad \text{with} \quad i \in {}^{\Gamma}S^{\mathrm{s}}.$$

Based on the position of this curve the fluid mesh can now be adapted. Therefore, the set of intersection points ought to be found of all the fluid edges that intersect $\gamma^{\mathrm{s}}$

$$S^{\mathrm{f}} = \left\{ i \in {}^{\Gamma}S^{\mathrm{f}} : \Gamma_i^{\mathrm{f}} \cap \gamma^{\mathrm{s}} \neq \emptyset, \Gamma_i^{\mathrm{f}} \cap \gamma^{\mathrm{s}} \neq \Gamma_i^{\mathrm{f}} \right\},$$
$$X_i = \left\{ \boldsymbol{x} : \boldsymbol{x} \in \Gamma_i^{\mathrm{f}} \cap \gamma^{\mathrm{s}} \right\} \quad \forall i \in S^{\mathrm{f}}.$$

Note that $S^{\mathrm{f}}$ contains the edge numbers of all edges that intersect $\gamma^{\mathrm{s}}$ and $X_i$ contains all corresponding intersection points. If more than one intersection point is stored in $X_i$, only one, arbitrarily chosen, point is taken into account in the remainder of the adaptive meshing algorithm. Now, a parameter $\varepsilon_i$ can be introduced, which describes the relative position of an intersection point along an edge:

$$\varepsilon_i = \frac{\|\boldsymbol{x}_i - {}_i\boldsymbol{x}_1\|}{\|{}_i\boldsymbol{x}_2 - {}_i\boldsymbol{x}_1\|} \quad \forall \boldsymbol{x}_i \in X_i.$$

Based on this parameter together with the pre-defined parameter $\hat{\varepsilon}$, set $S^{\mathrm{f}}$ can be subdivided into the subsets

$$S^{\mathrm{f1}} = \left\{ i \in S^{\mathrm{f}} : \varepsilon_i < \hat{\varepsilon} \|\Gamma_i^{\mathrm{f}}\| \right\},$$
$$S^{\mathrm{f2}} = \left\{ i \in S^{\mathrm{f}} : \varepsilon_i > (1 - \hat{\varepsilon}) \|\Gamma_i^{\mathrm{f}}\| \right\},$$
$$S^{\mathrm{f3}} = \left\{ i \in S^{\mathrm{f}} : \hat{\varepsilon} \|\Gamma_i^{\mathrm{f}}\| < \varepsilon_i < (1 - \hat{\varepsilon}) \|\Gamma_i^{\mathrm{f}}\| \right\},$$

in which $\|\Gamma_i^{\mathrm{f}}\| \equiv \|{}_i\boldsymbol{x}_2 - {}_i\boldsymbol{x}_1\|$ is the length of an edge. The arbitrary parameter $\hat{\varepsilon}$ should have a value between 0 and 0.5 and is explained graphically in Fig. 3. In this way, every intersected fluid edge is subdivided into three parts and sets $S^{\mathrm{f1}}$, $S^{\mathrm{f2}}$ and $S^{\mathrm{f3}}$ contain the edge numbers of the edges that are intersected in part one, two or three, respectively. Using these three sets the actual adaptation of the mesh is performed as follows:

$$\begin{aligned}
\boldsymbol{x}_{n_1} &= \boldsymbol{x}_i \quad \text{with} \quad \boldsymbol{x}_i \in X_i, \;\; n_1 \in {}^{x}S^{\mathrm{f}}, \;\; \forall i \in S^{\mathrm{f1}}, \\
\boldsymbol{x}_{n_2} &= \boldsymbol{x}_i \quad \text{with} \quad \boldsymbol{x}_i \in X_i, \;\; n_2 \in {}^{x}S^{\mathrm{f}}, \;\; \forall i \in S^{\mathrm{f2}}, \\
\hat{\boldsymbol{x}}_{\hat{n}_i} &= \boldsymbol{x}_i \quad \text{with} \quad \boldsymbol{x}_i \in X_i, \;\; \forall i \in S^{\mathrm{f3}},
\end{aligned}$$

in which $\boldsymbol{x}_{n_1} = {}_i\boldsymbol{x}_1$, $\boldsymbol{x}_{n_2} = {}_i\boldsymbol{x}_2$ and $\hat{\boldsymbol{x}}_{\hat{n}_i}$ is the position of a newly created node $\hat{n}_i$ on edge $\Gamma_i^{\mathrm{f}}$. From the above it follows that fluid nodes are either repositioned or added to the mesh. At the intersection point of the edges in $S^{\mathrm{f3}}$, a new node is created. Consequently, each of these intersected edges is divided into two new edges. For all edges in $S^{\mathrm{f1}}$ and $S^{\mathrm{f2}}$ the begin or end node are shifted to the position of the intersection, respectively. We impose that the position of a node is allowed to be changed only once, even if this node is shared by several intersected edges. After the procedure of relocating and adding edges, topological changes are processed. Depending on the number of intersections a triangular fluid element is split into several elements. Elements with one, two and three intersected edges are subdivided into two, three and four new elements, respectively. There are two ways of subdividing a triangle with two intersections into three sub-triangles and in our case the subdivision was chosen, which leads to the best-shaped sub-triangles, i.e., resembling an ideal triangular reference element closest. Note that for small values of parameter $\hat{\varepsilon}$ more elements will be created (Fig. 4), however, the created elements will be increasingly ill-shaped. Now, using the edges $\Gamma_i^{\mathrm{f}} \subset \Omega^{\mathrm{f}}$, of the adapted fluid mesh ${}^{\Omega}S^{\mathrm{f}*}$, the new subdomain $\gamma^{\mathrm{f}} \subset \Omega^{\mathrm{f}}$ can be defined as

$$\gamma^{\mathrm{f}} = \left\{ \boldsymbol{x} \in \Gamma_i^{\mathrm{f}} : {}^{i}\boldsymbol{x}_1 \in \gamma^{\mathrm{s}}, {}_i\boldsymbol{x}_2 \in \gamma^{\mathrm{s}} \right\} \quad \forall i. \tag{16}$$

Since $\gamma^{\mathrm{s}}$ is part of the solid, which moves through the fluid domain, curve $\gamma^{\mathrm{f}}$, which is based on $\gamma^{\mathrm{s}}$, will also change every time step.

#### 4.1.2. Smoothing

In order to improve the shape of the fluid elements near $\gamma^{\mathrm{f}}$, Laplacian smoothing is applied [21] to the fluid vertex nodes lying in the vicinity of, but not on $\gamma^{\mathrm{f}}$, according to

$$\boldsymbol{x}_j = \frac{1}{w^N} \sum_{k=1}^{N} w_k \boldsymbol{x}_k \quad \text{with} \quad w^N = \sum_{k=1}^{N} w_k. \tag{17}$$

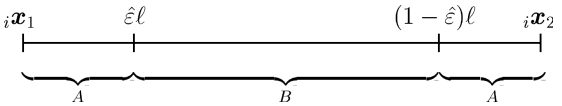


Fig. 3. Fluid edge of length $\ell$ that is subdivided into parts A and B depending on $\hat{\varepsilon}$. If the fluid edge is intersected in part A then the closest fluid node is shifted, if part B is disected then a new fluid node is added.

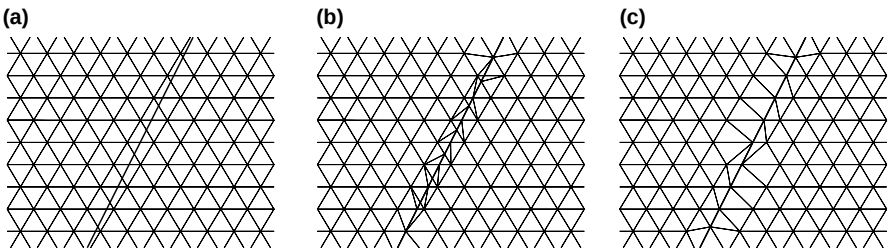


Fig. 4. A triangular fluid mesh is intersected by a boundary (a), after which the fluid mesh is adapted for $\hat{\varepsilon} = 0.2$ (b) and $\hat{\varepsilon} = 0.5$ (c).

Symbol $N$ denotes the number of mesh points $\boldsymbol{x}_k = {}_i\boldsymbol{x}_1$ sharing an edge $\Gamma_i$ with point $\boldsymbol{x}_j = {}_i\boldsymbol{x}_2$. The weight functions $w_k$ corresponding to $\boldsymbol{x}_k$ can be used for introducing gradients in element size. In the case $w_k = 1$, point $\boldsymbol{x}_j$ will be repositioned to the geographical center of the surrounding points. However, if values larger than one are chosen for nodes that lie closer to $\gamma^{\mathrm{f}}$, a refinement is obtained perpendicular to this boundary which is convenient since velocity gradients are largest in the boundary layer. This way the overall mesh size can be coarser reducing computational effort without compromising accuracy. The size of the area which will be smoothed is adjustable depending on the needs.

#### 4.1.3. Element choice

The solid domain is discretised using quadratic nine noded quadrilaterals or so-called $Q_2 - P_1$ elements as for the fluid domain seven noded quadratic triangles or $P_2^+ - P_1$ elements are used. Both elements are so-called Crouzeix–Raviart type of elements, i.e., elements with a discontinuous pressure approximation. Space $Q_2$ contains biquadratic polynomials used for the approximation of the solid displacements. Spaces $P_2$ and $P_1$ contain the polynomials of order two and one, respectively, used for the approximation of the fluid velocity and the pressure. The + sign denotes the enrichment of the elements by a third-order term which vanishes at the element boundary. Note that these elements are not arbitrarily chosen, since the discontinuous pressure interpolation for the fluid elements (across element borders) will be used to capture pressure gradients in the fluid domain across the solid as will be discussed in Section 5.1. Finite element spaces for velocity vectors $\boldsymbol{u}^{\mathrm{f}}$, $\boldsymbol{u}^{\mathrm{s}}$ and pressures $p^{\mathrm{f}}$ and $p^{\mathrm{s}}$ read, respectively

$$\mathbb{W}_{\mathrm{h}}^{\mathrm{f}} = \left\{ \boldsymbol{u}_{\mathrm{h}}^{\mathrm{f}} \in C^0(\overline{\Omega})^2 : \boldsymbol{u}_{\mathrm{h}}^{\mathrm{f}}|_e \in (P_2^+)^2, \forall e \in {}^{\Omega}S^{\mathrm{f}*}, \boldsymbol{u}_{\mathrm{h}}^{\mathrm{f}}|_{\partial\Omega^{\mathrm{f}}} = 0 \right\},$$
$$\mathbb{W}_{\mathrm{h}}^{\mathrm{s}} = \left\{ \boldsymbol{u}_{\mathrm{h}}^{\mathrm{s}} \in C^0(\overline{\Omega})^2 : \boldsymbol{u}_{\mathrm{h}}^{\mathrm{s}}|_e \in (Q_2)^2, \forall e \in {}^{\Omega}S^{\mathrm{s}}, \boldsymbol{u}_{\mathrm{h}}^{\mathrm{s}}|_{\partial\Omega^{\mathrm{s}}} = 0 \right\},$$
$$L_{\mathrm{h}}^{\mathrm{f}} = \left\{ q_{\mathrm{h}}^{\mathrm{f}} : q_{\mathrm{h}}^{\mathrm{f}}|_e \in P_1, \forall e \in {}^{\Omega}S^{\mathrm{f}*} \right\},$$
$$L_{\mathrm{h}}^{\mathrm{s}} = \left\{ q_{\mathrm{h}}^{\mathrm{s}} : q_{\mathrm{h}}^{\mathrm{s}}|_e \in P_1, \forall e \in {}^{\Omega}S^{\mathrm{s}} \right\}.$$

Note that the introduction of second-order elements for fluid and solid also allows for quadratically curved element edges. To achieve this, the mesh can be adapted as described in Section 4.1.1 based on the vertex nodes of the triangular elements, performing only one additional step to change $\gamma^{\mathrm{f}}$. The fluid edges $\Gamma_i^{\mathrm{f}}$ that have ${}_i\boldsymbol{x}_1$ and ${}_i\boldsymbol{x}_2$ on $\gamma^{\mathrm{f}}$ become curved by projecting their center nodes onto $\gamma^{\mathrm{s}}$ in normal direction of edge $\Gamma_i^{\mathrm{f}}$. This way the geometrical difference between $\gamma^{\mathrm{s}}$ and $\gamma^{\mathrm{f}}$ reduces, improving the accuracy of solutions.

### 4.2. Discretisation of the Lagrange multiplier for fluid–structure interaction

Since boundary $\gamma^{\mathrm{f}}$ changes every time step, this has some implications for the discretisation of the Lagrange multiplier $\boldsymbol{\lambda}^{\mathrm{fsi}}$, that is defined along $\gamma^{\mathrm{f}}$. As derived earlier $\gamma^{\mathrm{f}}$ consists of a set of fluid edges and the discretisation of $\boldsymbol{\lambda}^{\mathrm{fsi}}$ is taken equal to discretisation of the fluid element edges of which $\gamma^{\mathrm{f}}$ consists. Piecewise discontinuous linear polynomials are used for the multiplier resulting in the corresponding finite element space

$$\mathbb{W}_{\mathrm{h}}^{\mathrm{fsi}} = \left\{ \boldsymbol{\lambda}_{\mathrm{h}}^{\mathrm{fsi}} : \boldsymbol{\lambda}_{\mathrm{h}}^{\mathrm{fsi}}|_e \in P_1, \forall e \in {}^{\gamma^{\mathrm{f}}}S \right\},$$

where ${}^{\gamma^{\mathrm{f}}}S$ contains the edge numbers of the fluid edges in $\gamma^{\mathrm{f}}$. Although different kinds of interpolation polynomials were tested, discontinuous linear interpolation led to the best result (without spurious currents at the solid–fluid interface), which might be expected since internal stresses and pressures are similarly described.

As stated in several papers [8,10,12,15], a combination of sufficient accuracy and stability has to be found in a fictitious domain method. Finer discretisations of Lagrange multiplier $\boldsymbol{\lambda}^{\mathrm{fsi}}$ lead to higher accuracy, while stability problems or even locking can occur when the discretisation of $\boldsymbol{\lambda}^{\mathrm{fsi}}$ becomes too fine with regard to the fluid or solid mesh. Additionally, the order of the interpolation functions used for multiplier and gradients in the element size for the multipliers are of importance. It is hard to find general rules that apply to the various existing problems. This is demonstrated by the findings of Yu [15] that the solid element size $h^{\mathrm{s}}$ should be equal or larger than the fluid element size $h^{\mathrm{f}}$, when using the same discretisation for the multiplier as for the solid. Baaijens [10], on the other hand, found that in his approach $h^{\mathrm{s}}$ should be preferably smaller than $h^{\mathrm{f}}$.

In Baaijens et al. [10], the Lagrange multiplier was discretised based on the discretisation of the solid resulting in a fixed amount of coupling elements throughout the computation. This approach has been examined with the incorporation of the adaptive meshing scheme for slender solid bodies moving in a fluid domain. However, depending on the position of the solid (and therefore coupling) elements with respect to the fluid mesh, the fluid solution near the solid was too low to derive sufficiently accurate shear stress information. Therefore, we decided to discretise the multiplier based on the fluid mesh such that each fluid element coincided with exactly one coupling element. The result was an accurate fluid solution near the solid boundary without introducing inaccuracies for the solid solution. The solid is probably less sensitive to the discretisation of the Lagrange multiplier due to its high stiffness compared to the low-viscid fluid. Only when $h^{\mathrm{s}} < 0.2h^{\mathrm{f}}$ spurious modes have been obtained in the solid body, but this highly depends on the type of loading, e.g., compression (bending) versus tension, and the corresponding deformations. Variations of the solid mesh size up to $h^{\mathrm{s}} = 3.5h^{\mathrm{f}}$ have given good results, reserved that it is possible to compute the deformation in a static case with that solid mesh.

In case boundary $\gamma^{\mathrm{s}}$ is not a closed boundary it is likely to occur that $\gamma^{\mathrm{f}}$ will not be equal to $\gamma^{\mathrm{s}}$. When $\gamma^{\mathrm{s}}$ ends within a fluid element the adaptive meshing scheme does not accommodate for a coupling element in this area. A small part of $\gamma^{\mathrm{s}}$, which depends on the element size of the fluid elements, will therefore not be coupled to the fluid, introducing leakage through this part. In the model problems, presented in Section 5, this error was reduced by applying a Lagrange multiplier at the free end of $\gamma^{\mathrm{s}}$ and couple fluid and solid in a more global manner as proposed by de Hart et al. [11].

### 4.3. Discretisation of the Lagrange multiplier for solid-rigid contact

The Lagrange multiplier corresponding to the contact problem is discretised somewhat differently to that in Section 4.2. For sake of simplicity with respect to the determination of the gap distance, the multiplier is only evaluated in the nodal points of the solid contact surface $\gamma^{\mathrm{c}}$. In this way the curvature of the quadratic solid elements is not fully taken into account in the case of contact, which is allowed for fine discretisations of the solid domain, resulting in low curvature per element. The corresponding finite element space for this collocation method then reads

$$\mathbb{W}_{\mathrm{h}}^{\mathrm{c}} = \left\{ \lambda_{\mathrm{h}}^{\mathrm{c}} : \lambda_{\mathrm{h}}^{\mathrm{c}}|_i = \lambda_{\mathrm{h},i}^{\mathrm{c}} \delta(\boldsymbol{x} - \boldsymbol{x}_i^{\mathrm{s}}), \lambda_{\mathrm{h},i}^{\mathrm{c}} \in \mathbb{R}, \ \ \forall i \in {}^{\gamma^{\mathrm{c}}}S \right\},$$

with ${}^{\gamma^{\mathrm{c}}}S$ a set containing the numbers of all solid nodes on boundary $\gamma^{\mathrm{c}}$ and $\delta$ denoting the Dirac function.

### 4.4. Combined fluid–structure interaction/contact

In the case of contact, two contradicting boundary conditions are present in this region. Firstly, Eq. (7), which couples the fluid to the solid and secondly, Eq. (8) which provides for the contact between fluid wall and solid, but allows slip. Furthermore, Dirichlet boundary conditions are present at the fluid wall. Therefore, a choice should be made as to these constraints should be the predominant one. If Eq. (7) would 'overrule' Eq. (8), a no-slip condition along the fluid wall would also be enforced to the solid, preventing it to come off the wall. In case Eq. (8) is dominant over Eq. (7) any essential boundary condition, applied to the wall will hold no effect on the solid. In the model presented here, fluid–structure coupling elements that have nodes with Dirichlet boundary conditions, are 'disabled'. For the degree(s) of freedom to which these boundary conditions are applied Eq. (7) is replaced by

$$\boldsymbol{\lambda}^{\mathrm{fsi}} \cdot \boldsymbol{n} = 0, \tag{18}$$

where $\boldsymbol{n}$ denotes the direction in which the Dirichlet boundary condition holds. The Lagrange multiplier is such eliminated from the momentum equations by enforcing it to zero. This allows for free slip of the solid along and loosening from the fluid wall, although boundary conditions may enforce the opposite.

### 4.5. Full set of discretised equations

If a backward Euler scheme is used for the time discretisation. Taking a first-order approximation for the solid velocity $\boldsymbol{u}^{\mathrm{s}} = \Delta \boldsymbol{x}^{\mathrm{s}}/\Delta t$ and applying it to Eqs. (12), (13) and (15) then leads to a set of equations in

which fluid and solid velocities are solved. The structure of the total set of discretised equations is then given by

$$\left[\begin{array}{cc|cc|cc} \underline{K}^{\mathrm{f}} & \underline{L}^{\mathrm{f}} & \underline{0} & \underline{0} & \underline{\Lambda}^{\mathrm{f}}_{fsi} & \underline{0} \\ (\underline{L}^{\mathrm{f}})^{\mathrm{T}} & \underline{0} & \underline{0} & \underline{0} & \underline{0} & \underline{0} \\ \hline \underline{0} & \underline{0} & \Delta t\underline{K}^{\mathrm{s}} & \underline{L}^{\mathrm{s}} & \underline{\Lambda}^{\mathrm{s}}_{\mathrm{fsi}} & \underline{\Lambda}^{\mathrm{cont}} \\ \underline{0} & \underline{0} & (\underline{L}^{\mathrm{s}})^{\mathrm{T}} & \underline{0} & \underline{0} & \underline{0} \\ \hline \underline{G}^{\mathrm{f}}_{\mathrm{fsi}} & \underline{0} & \underline{G}^{\mathrm{s}}_{\mathrm{fsi}} & \underline{0} & \underline{G}^{\mathrm{no\text{-}fsi}} & \underline{0} \\ \hline \underline{0} & \underline{0} & \underline{C}^{\mathrm{cont}} & \underline{0} & \underline{0} & \underline{C}^{\mathrm{no\text{-}cont}} \end{array}\right] \left[\begin{array}{c} \underset{\sim}{u}^{\mathrm{f}} \\ \underset{\sim}{p}^{\mathrm{f}} \\ \underset{\sim}{u}^{\mathrm{s}} \\ \underset{\sim}{p}^{\mathrm{s}} \\ \underset{\sim}{\lambda}^{\mathrm{fsi}} \\ \underset{\sim}{\lambda}^{\mathrm{c}} \end{array}\right] = \left[\begin{array}{c} \underset{\sim}{f}^{\mathrm{f}}_{u} \\ \underset{\sim}{f}^{\mathrm{f}}_{p} \\ \underset{\sim}{f}^{\mathrm{s}}_{u} \\ \underset{\sim}{f}^{\mathrm{s}}_{p} \\ \underset{\sim}{f}^{\mathrm{fsi}} \\ \underset{\sim}{f}^{\mathrm{c}} \end{array}\right],$$

in which the solid and fluid system matrices $\underline{K}^{\mathrm{f}}$ and $\underline{K}^{\mathrm{s}}$ can be recognised with corresponding divergence matrices $\underline{L}^{\mathrm{f}}$ and $\underline{L}^{\mathrm{s}}$. With respect to fluid-interaction the matrices $\underline{\Lambda}^{\mathrm{f}}_{\mathrm{fsi}}$ and $\underline{\Lambda}^{\mathrm{s}}_{\mathrm{fsi}}$ are added to meet the dynamic constraint and corresponding matrices $\underline{G}^{\mathrm{f}}_{\mathrm{fsi}}$, $\underline{G}^{\mathrm{f}}_{\mathrm{fsi}}$ and $G_{\mathrm{no\text{-}fsi}}$ in order to meet the kinematic constraint. This last matrix is filled in case there are nodes in contact. Similarly, $\underline{\Lambda}^{\mathrm{cont}}$ is added to meet the dynamic constraint and $\underline{C}^{\mathrm{cont}}$ and $\underline{C}^{\mathrm{no\text{-}cont}}$ take care of the kinematic condition for the contact problem in the case of contact and no-contact, respectively. A Newton–Raphson iteration scheme is performed each time step to solve the non-linear set of equations. Note that, within this scheme the current state of contact is evaluated at each iteration step. This implies that the contact distance or direction of normal $\boldsymbol{n}^{\mathrm{c}}$ can alter and nodes that are in contact might be released in the iterative procedure. However, the system is in equilibrium after convergence of the scheme at the end of each time step. The finite element package SEPRAN [22] is used and extended for building the matrix and mapping the solutions. The set of equations is solved fully coupled and the asymmetric sparse matrix is solved using a direct method based on a sparse multifrontal variant of Gaussian elimination (HSL/MA41) [23].

### 4.6. Combined ALE and mapping

Coupling a non-conforming fluid and solid mesh allows for a relatively simple way of extending the fluid–structure algorithm with contact. In ALE methods as often used for fluid–structure interaction problems, the Eulerian fluid mesh moves in a Lagrangian way 'fixed' to the solid. If the solid approaches a boundary (e.g., fluid wall) fluid elements will be squeezed. Unlike these ALE methods, the boundary ($\gamma^{\mathrm{f}}$) at which the no-slip condition (Eq. (7)) is applied, changes in time. As shown earlier, $\gamma^{\mathrm{f}} \subseteq \gamma^{\mathrm{s}}$, which implies that the region along which no-slip holds can be smaller than the region along which it should hold. Although this introduces an inaccuracy as mentioned earlier, it allows for approaching boundaries and sliding along boundaries without the risk of squeezing fluid elements.

The reason for introducing an adaptive meshing algorithm was the need for accurate shear stresses and pressures at both sides of the solid. Since the Lagrangian solid moves in time, the fluid mesh requires continuous updating. Every time step an adapted fluid mesh ${}^{\Omega}S^{\mathrm{f}*}$ is computed based on the initially generated fluid mesh ${}^{\Omega}S^{\mathrm{f}}$ using the adaptive meshing algorithm as described in Section 4.1.1. The set of equations is solved using the adapted mesh ${}^{\Omega}S^{\mathrm{f}*}$ after which the solid is updated. Another adapted mesh ${}^{\Omega}S^{\mathrm{f}**}$ is then generated based on the new position of the solid and on mesh ${}^{\Omega}S^{\mathrm{f}}$. Since ${}^{\Omega}S^{\mathrm{f}*}$ and ${}^{\Omega}S^{\mathrm{f}**}$ are both created from ${}^{\Omega}S^{\mathrm{f}}$, large parts of these meshes will coincide except near the solid position. Within one time step an ALE and mapping procedure are used for projecting the old solution onto the ${}^{\Omega}S^{\mathrm{f}**}$ as accurately as possible, which will be explained next and which is also drawn schematically in Fig. 5.

Firstly, an auxiliary mesh is generated by repositioning all nodes $n_i$ that lie on curve $\gamma^{\mathrm{f}}$ with step size, $\Delta \boldsymbol{x}_i = \boldsymbol{u}_i * \Delta t$, resulting in a mesh which is topologically identical to ${}^{\Omega}S^{\mathrm{f}*}$, but has a set of fluid nodes that are shifted towards the new position of the solid. Applying these steps incrementally and alternating with a

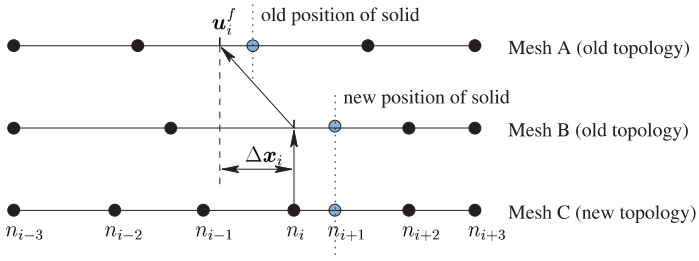


Fig. 5. Schematic representation of mapping procedure using 1D meshes. The velocity field is available on Mesh A and projected through the topological identical Mesh B to Mesh C. For every node $n_i$ a solution $\boldsymbol{u}_i^{\mathrm{f}}$ can be found by interpolation. The distance $\Delta \boldsymbol{x}_i$ between the positions in Mesh A and Mesh B at which this solution is found can be divided by the time step to obtain the grid velocity of node $n_i$.

Laplacian smoothing procedure as described in Section 4.1.1, prevents elements with negative Jacobian. Note that near borders of the fluid domain squeezing elements to zero-volume is possible in the case of contact. This does, however, not cause problems since this auxiliary mesh will not be used for computational purpose, but is only a tool for mapping solutions from ${}^{\Omega}S^{\mathrm{f}*}$ to ${}^{\Omega}S^{\mathrm{f}**}$ more correctly.

Secondly, for every node $n_i$ in ${}^{\Omega}S^{\mathrm{f}**}$ an old velocity $\boldsymbol{u}_i^{\mathrm{f}}$ and grid velocity $\boldsymbol{u}_i^{\mathrm{grid}}$ have to be determined. Therefore, we need to find in which element of the auxiliary mesh $n_i$ is situated and determine the relative position within this element, after which the old velocity $\boldsymbol{u}_i^{\mathrm{f}}$ is computed using the interpolation functions of the $P_2^+ - P_1$ elements. Note that the topological similarities between the meshes contribute to fast element searching. Since the old solution is derived from the auxiliary mesh and not from ${}^{\Omega}S^{\mathrm{f}*}$, convection of this point should be accounted for by determining the global position in ${}^{\Omega}S^{\mathrm{f}*}$ based on the local position within the element. The distance between this global position and current position of $n_i$ divided by the time step gives the grid velocity $\boldsymbol{u}_i^{\mathrm{grid}}$, which can be used to incorporate the convection of point $n_i$ [2]. The convective part in Eq. (3) can then be rewritten as

$$\rho(\boldsymbol{u}^{\mathrm{f}} - \boldsymbol{u}^{\mathrm{grid}}) \cdot \nabla \boldsymbol{u}^{\mathrm{f}}. \tag{19}$$

Some remarks can be made with respect to the projection of the velocity field from one mesh to another. Mapping solutions between two topologically different meshes by interpolation is known to result in a non-divergence free flow, especially when the velocity gradients are large. However, this introduced error is found to be small for several reasons. Firstly, an ALE step is undertaken in order to capture the large velocity gradients across the solid more correctly. Secondly, the parts of the meshes, which are topologically different, are small. Thirdly, the velocities at the solid boundary are constrained (Eq. (7)) and divergence free flow enforced every time step (Eq. (2)), which prevents error cumulation. In addition, we should note that although zero-volume elements are allowed in the auxiliary mesh, these elements should not be taken into account for the interpolation.

## 5. Numerical experiments

Two-model problems are presented that treat the movement of a flexible slender solid body through a fluid domain induced by a pulsatile flow that is applied at the inlet of the fluid domain. The method allows for interaction of the solid with the walls of the fluid domain. The fluid dynamics (velocities and pressures) as well as the movement and deformation of the solid (displacements and pressures) are presented. Different velocity and pressure fields at either side of the solid are shown and furthermore the computation of derived quantities like shear stresses along the solid boundary or stresses and strains inside the solid body, which are important in heart valve analysis, is possible.

### 5.1. Contact of solid slab in pulsatile flow

In the first model problem we consider a rectangular fluid domain with a thin rectangular solid inside as shown in Fig. 6, with the following essential boundary conditions:

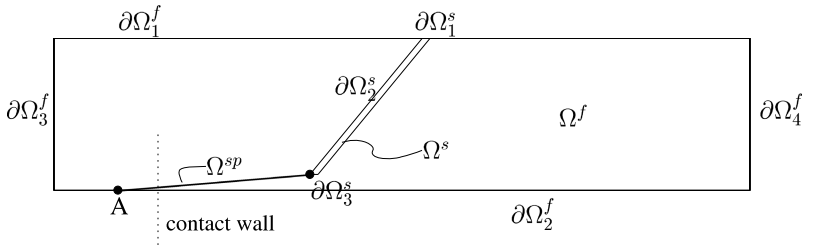


Fig. 6. Schematic representation of a mitral valve $\Omega^s$ in a fluid canal $\Omega^f$. The papillary muscle is denoted by $\Omega^{sp}$ which is attached to $\Omega^s$ at one side and can slide along $\partial\Omega_2^s$ at the other with point 'A' prohibited to tresspass the contact wall.

$$\begin{aligned} u_1^f &= 0 \quad \text{at} \quad \overline{\partial\Omega}_1^f, \\ u_1^f &= u_{\max}\sin(2\pi t/T) \quad \text{at} \quad \partial\Omega_3^f, \\ u_2^f &= 0 \quad \text{at} \quad \overline{\partial\Omega}_i^f \quad \text{with} \quad i = 1,\ldots,4, \\ \boldsymbol{u}^s &= 0 \quad \text{at} \quad \overline{\partial\Omega}_1^s, \end{aligned} \tag{20}$$

with $\boldsymbol{u}^f = [u_1^f, u_2^f]$. Note that domain $\Omega^{sp}$ is not incorporated in this first model problem. In Fig. 7 the relevant dimensional parameters are given and the corresponding values for the models in this and the following section are shown in Table 1. The fluid domain has a finer discretisation in the area around the solid but for the long inlet and outlet, that are introduced to prevent boundary effects, a coarser discretisation is used. The solid mesh is discretised into 40 quadrilateral elements (2 elements across the thickness times 20 elements along the length). For the fluid mesh a total of 3402 triangular elements is used. Note that this is the number of fluid elements prior to the computations, but that this amount increases due to adaptive meshing. In this model problem parameter $\hat{\varepsilon}$ is set to 0.2, which causes the actual number of fluid elements to vary between 3424 and 3472. The number of coupling elements lies between 26 and 40. Since the width of the solid is very small compared to its length, it is assumed that only coupling along boundary $\overline{\partial\Omega}_1^s$ is needed. As a consequence the mesh will be adapted only along this solid boundary. Consequentially, boundary $\partial\Omega_2^s$ is defined as the deformable contact surface in the contact problem, while boundary $\overline{\partial\Omega}_2^f$ is defined as the rigid contact surface in the problem.

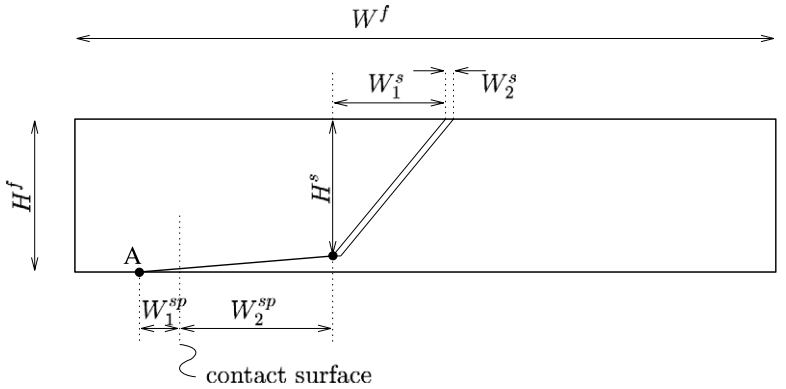


Fig. 7. Schematic representation of the solid and fluid including the relevant geometric parameters. Corresponding values for the numerical experiments performed in Sections 5.1 and 5.2 are given in Table 1.

Table 1
Relevant geometric parameter values for the numerical experiments

| | $H^f$ (cm) | $W^f$ (cm) | $H^s$ (cm) | $W_1^s$ (cm) | $W_2^s$ (cm) | $W_1^{sp}$ (cm) | $W_2^{sp}$ (cm) |
|---|---|---|---|---|---|---|---|
| Numerical experiment 5.1 | 2.0 | 40.0 | 1.9 | 1.0 | $5\times10^{-2}$ | – | – |
| Numerical experiment 5.2 | 1.0 | 24.0 | 0.9 | 2.0 | $5\times10^{-2}$ | 1.0 | 3.0 |

Table 2
Relevant parameter values for the numerical experiments

| | $u_{max}$ (m/s) | $p_{max}$ (Pa) | $\eta$ (Pa s) | $\rho$ (kg/m$^3$) | $\Delta t$ (s) | $T$ (s) | $G$ (MPa) |
|---|---|---|---|---|---|---|---|
| Numerical experiment 5.1 | 0.2 | – | $4.0\times10^{-3}$ | $10^3$ | $10^{-3}$ | 1.0 | 1.2 |
| Numerical experiment 5.2 | – | 500.0 | $4.0\times10^{-3}$ | $10^3$ | $10^{-3}$ | 1.0 | 1.0 |

The material and time related parameter values used for the numerical experiments of this and the following section are shown in Table 2. This leads to a maximum Reynolds number of 1000 for this problem and a Strouhal number of 0.6, taking the height of the canal and the maximum applied velocity at the inlet as characteristic length and velocity, respectively.

Streamline plots at several moments are given (Fig. 8(a)–(d)) to get an idea of the flow field in time. The computation starts when the solid body is about to be in contact with $\partial\Omega_2^f$. After several time steps it is in contact with $\partial\Omega_2^f$ and, induced by the applied flow at $\partial\Omega_3^f$, the solid starts to slide along $\partial\Omega_2^f$. At this point we need to remark that due to the choice of $\partial\Omega_2^s$ as the deformable contact surface penetration of the remaining part of the solid body is allowed. However, defining the entire solid boundary as the deformable contact surface would lead to leakage in the closed situation, since FSI coupling is only applied along $\partial\Omega_2^s$. Due to its stiffness, the solid inhibits the flow and the pressure will start to rise in the left part of the fluid (Fig. 9(a)). Note that the model is able to describe this pressure drop, by the combination of adaptive meshing and use of the $P_2^+ - P_1$ fluid elements. The adaptive meshing provides for a set of fluid edges coinciding with $\partial\Omega_2^s$ and the $P_2^+ - P_1$ fluid elements are able to compute the discontinuity of the pressure across this set of fluid edges [18]. The solid will slide along $\partial\Omega_2^f$ until it has bent sufficiently such that the contact pressure becomes 0.

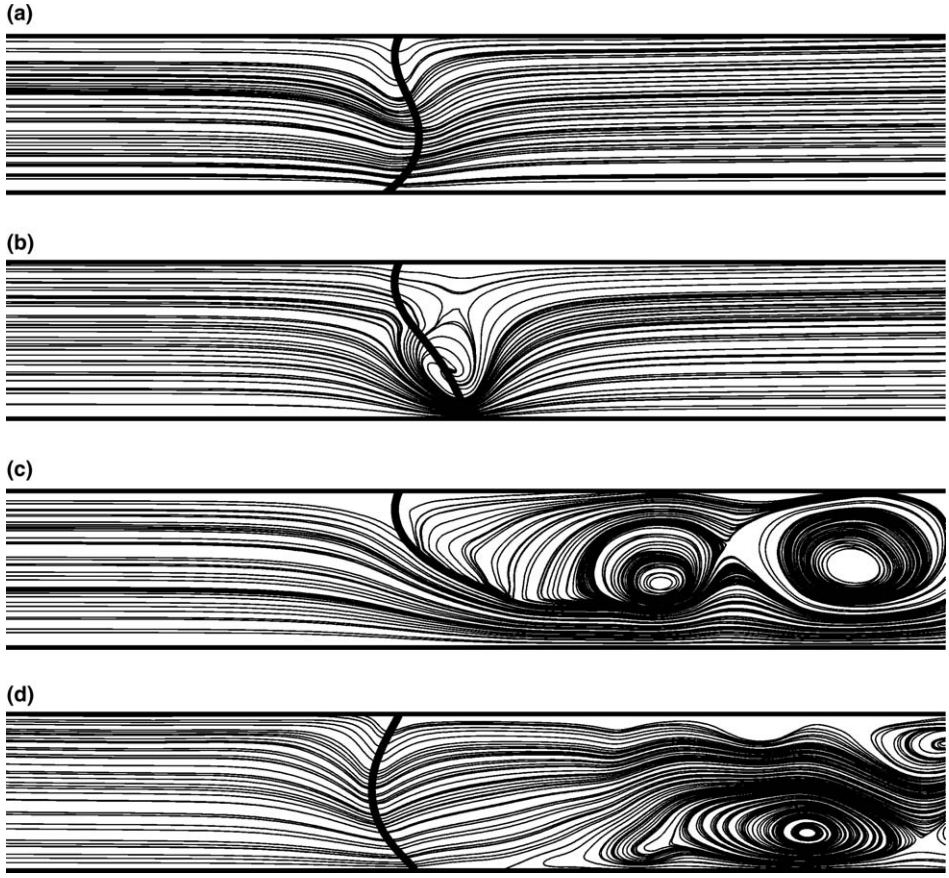


Fig. 8. The streamline plots and position of the solid slab at different moments in time: (a) $t/T = 0.1$, (b) $t/T = 0.12$, (c) $t/T = 0.3$, and (d) $t/T = 0.59$.

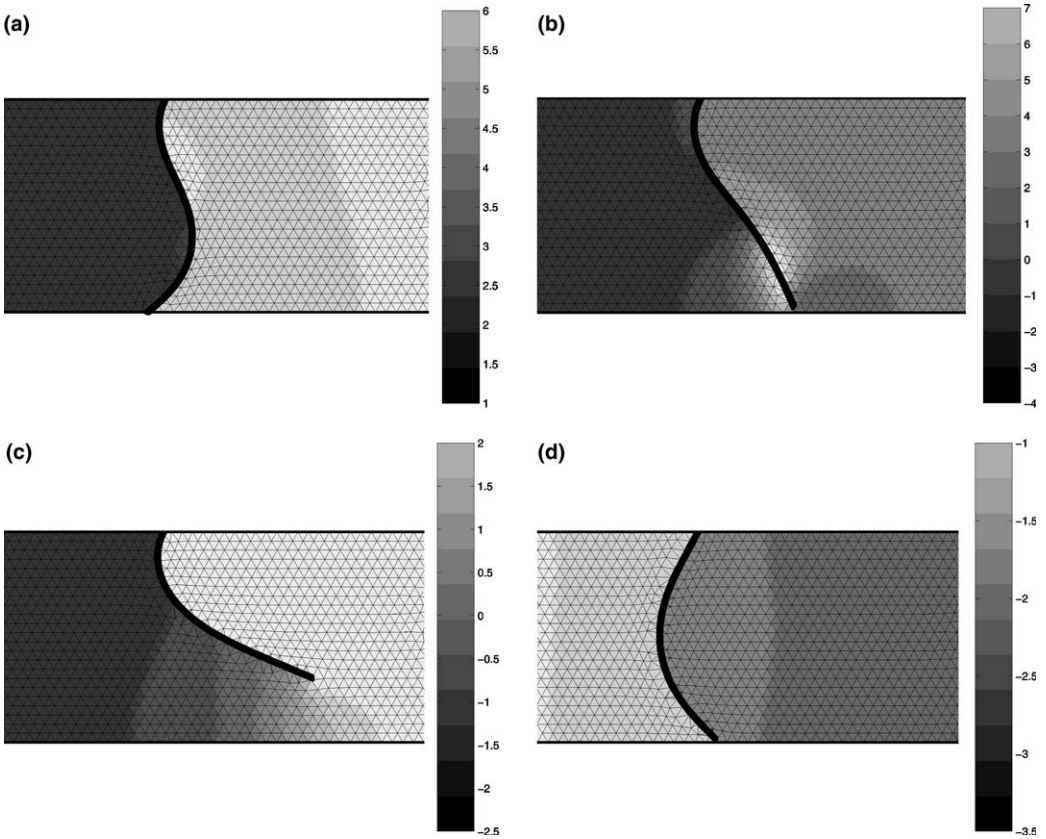


Fig. 9. Pressure contour bands and fluid mesh at different moments in time: (a) $t/T = 0.1$, (b) $t/T = 0.12$, (c) $t/T = 0.3$, and (d) $t/T = 0.59$.

Due to the elastic energy stored in the solid during bending this loosening from the wall happens very quickly even inducing a negative pressure at the left side of the tip (Fig. 9(b)). Contact is released and pressure in the left part of the fluid domain will start to drop caused by the squeezing flow between solid and boundary. The solid bends further until an apparent stationary position is obtained (Fig. 9(c)). In the second part of the computation, a similar movement of the solid is seen only in reversed direction (Fig. 9(d)).

In Fig. 10, the Lagrange multiplier in the $x$-direction is plotted at several time steps at the moment that the solid releases from the wall. This moment is interesting since there is a pressure difference present between the left and right part of the fluid domain. Furthermore, elastic energy is stored in the solid due to bending. This ''unbalance'' in energy is released in a short period of time. As mentioned earlier the physical meaning of the

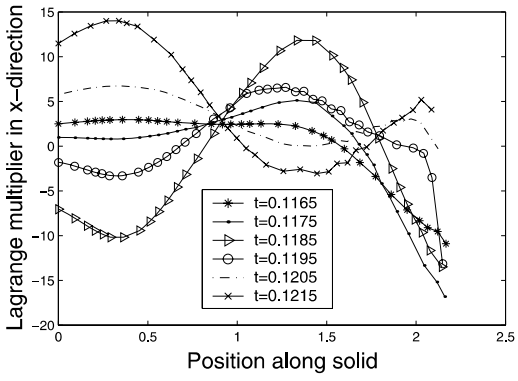


Fig. 10. Lagrange multiplier in $x$-direction along the solid during ''opening''.

multiplier in our problem is the traction stress between fluid and solid, and provides us with a good impression of the actual interaction. At $t/T = 0.1165$ and $t/T = 0.1175$, the solid still makes contact with the surface and it inhibits the fluid to flow in the positive $x$-direction. However, at $t/T = 0.1185$ the valve detaches from the wall and the solid starts to accelerate resulting in a change of sign of the multiplier. For a short period the solid pushes the flow in the positive $x$-direction. The last two time steps in Fig. 10 again show a change in sign.

### 5.2. 2D mitral valve

It is difficult to mimic the behaviour of a three dimensionally shaped aortic valve into a two dimensional computational model, because the shape of the valve has large influence on its overall stiffness, which cannot be captured in two dimensions. In this respect a 2D model of a mitral valve is more representative. Since the shape of this bileaflet does not contribute sufficiently to the overall stiffness nature has 'invented' the chordae tendae. These muscles that are attached to the heart wall on one side and to the leaflet tip on the other side, prevent the leaflets from prolapsing. During opening of the valve the muscle contracts thereby not hampering the leaflets. In this second model problem we aimed to mimic this behaviour by introducing domain $\Omega^{\mathrm{sp}}$ as shown in Fig. 6. This almost rigid solid is attached to $\Omega^{\mathrm{s}}$ at one end allowing free pivoting, and attached to the fluid wall at point 'A' along which it can slide freely. However, this point is not allowed to pass the contact wall denoted by the dotted line. All relevant geometric parameters for this problem are given in Fig. 7 with corresponding values presented in Table 1. In this problem solid mesh is discretised into 100 quadrilateral elements (2 elements across the thickness times 50 elements along the length).

The boundary conditions are similar to the ones defined for the model problem in Section 5.1, except for the essential boundary condition at the inlet $\partial\Omega_3^{\mathrm{f}}$ which is replaced by a natural one reading

$$p^{\mathrm{f}} = p_{\max}\sin(2\pi t/T) \quad \text{at} \quad \partial\Omega_3^{\mathrm{f}}. \tag{21}$$

The motion of domain $\Omega^{\mathrm{s}}$ induced by the fluid is shown in Fig. 11(a) over one full sinus. First, pressure will build up in the left part of the fluid domain pushing the solid to the right as given by the solid lines. Since the leaflets are flexible the solid tends to prolapse as seen in the first model problem. However, at a certain moment

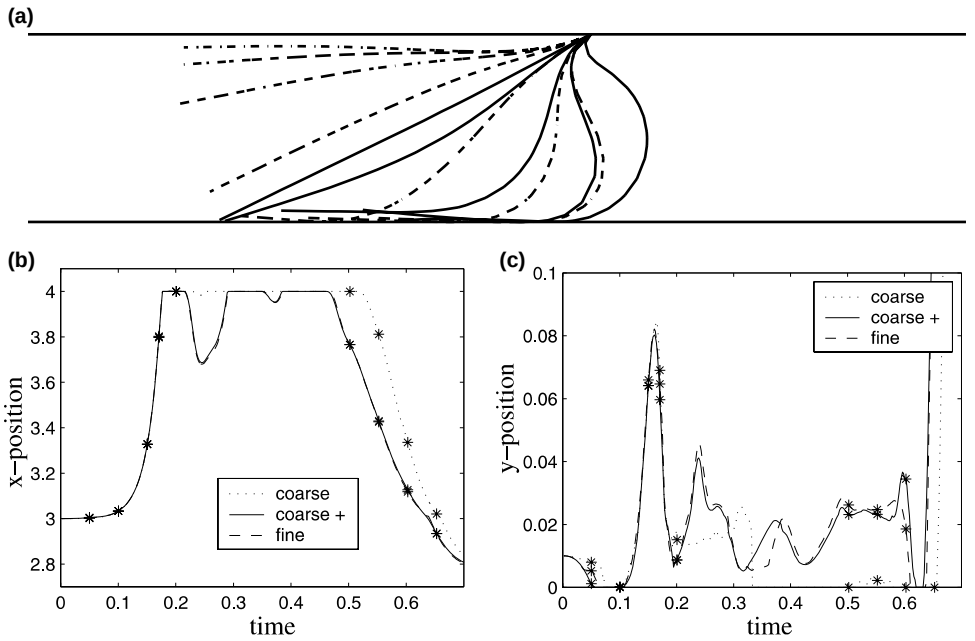


Fig. 11. Motion of a 2D mitral valve in a pulsatile flow (a). The solid and dashed lines represent the motion in time for $t/T < 0.5$ and $t/T > 0.5$, respectively. The corresponding points in time are given in (b) and (c) by the asterisks. In (b) and (c) the $x$- and $y$-position of the tip in time is given for three different meshes, respectively. The difference between 'coarse' and 'coarse +' is that the latter has a thin layer of elements along its contact boundary, improving solutions.

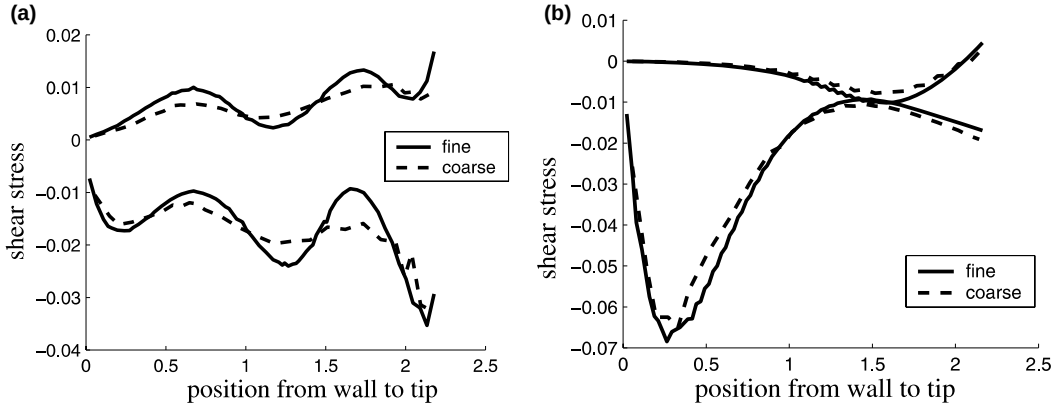


Fig. 12. Shear stresses along both sides of the solid for a coarse and fine mesh at $t/T = 0.64$ (a) and $t/T = 0.96$ (b) (mitral valve computation).

in time point 'A' reaches the contact wall (Fig. 6), which prohibits the solid tip to move further. If the pressure in the left part of the domain drops the solid will start moving to the left as denoted by the dash-dotted lines. In order to zoom in on the behaviour near the fluid wall the $x$- and $y$-position of the tip plotted as a function of time, respectively, for different discretisations of the fluid mesh (Fig. 11(b) and (c)). The asterisks in these graphs denote the points in time at which the solid is plotted in Fig. 11(a). The "coarse" mesh and "coarse +" (approximately 900 fluid elements) mesh have the same discretisation except for a very thin layer of fluid elements along the fluid wall. Comparison of the results shows that this layer improves the solution largely, since it influences the moment of contact and release, which are important for the overall movement of the solid. In case a layer is present the solid motion is almost similar to that computed with a fine mesh (approximately 3800 fluid elements). In the area around the solid this fine mesh has an element size about 2.5 times as small as that of the coarse meshes.

Beside discontinuous pressures, the $P_2^+ - P_1$ elements also allow for discontinuities in the derivatives of the velocities across the element borders, which can be used to compute shear stresses at both sides of the solid. Fig. 12(a) and (b) shows the shear stress along the solid at $t/T = 0.64$ and $t/T = 0.96$, respectively, at both sides of the solid. Although the coarser mesh hardly influenced the solid motion, a certain mesh resolution is needed for capturing the shear stresses accurately. The Reynolds number based on average inlet velocity and height of the domain is approximately 1750 at $t/T = 0.96$. The Strouhal number at this point is about 0.09.

## 6. Conclusion

A method is proposed to compute the motion of a heart valve in a pulsatile blood flow. Transvalvular pressure gradients can be described and shear stresses at either side of the valve can be accurately computed. Furthermore, sliding of the valve along a symmetry axis is performed in a fully coupled FSI/contact problem. The method is tested in two-model problems showing these phenomena.

Using a finite element method a fluid mesh and an immersed solid mesh are generated independently of one another. The fluid mesh is adapted every time step near an arbitrarily defined boundary of the solid at which fluid–structure coupling should take place. Depending on where this boundary intersects a fluid element, a vertex node of this element is shifted onto the boundary or the element is split by adding a node onto the boundary. This way an inner fluid curve is created which consists of fluid edges and coincides with the chosen solid boundary. Next, a Lagrange multiplier can be defined along this fluid curve that imposes that fluid and solid velocities should equal at this curve. By imposing this constraint weakly, discretisations for the solid and fluid mesh can be different.

To describe the closing and opening behaviour of a valve it is necessary to extend this fluid–structure part with contact. Therefore, we define a rigid body, whose position is known, and a contact surface, which is (part of) the boundary of the solid body. By introducing an extra constraint to the existing set of equations the contact surface can be prohibited from penetrating the rigid body. In this paper, free slip along the rigid surface is allowed.

The weak form of fluid–structure coupling allows not only for large movements of the solid through the fluid domain, but also simplifies the introduction of contact. Since fluid and solid meshes are non-conforming with respect to each other the solid can approximate the boundaries without squeezing elements. The method combines the benefits of techniques with Lagrange multipliers, adaptive meshing and ALE methods, resulting in an implicit code solving a fully coupled system and capturing difficult behaviour with good accuracy.

The combination of these techniques were shown and discussed in two-model problems. The first problem treats the movement of a flexible solid in a channel induced by a pulsatile flow at the inlet of the canal. The thin long solid is attached to the upper wall of the canal and is allowed to make contact with the symmetry axis. Throughout the computation the solid makes contact, a pressure difference builds up in the fluid across the solid, the solid releases from the wall, which lowers the pressure difference. The second problem is a 2D representation of a mitral valve thereby introducing an extra solid denoting the papillary muscle, which prevents the leaflets from prolapsing. In this problem the movement near the boundary was studied more closely and shear stresses along both sides were presented.

In this way some complexities of heart valve analysis are overcome, but some critical notes are appropriate here. The model as presented in this paper only addresses 2D computations as 3D influences are essential in heart valve analysis. Furthermore, the compliance of the wall has not been taken into account, which is known to be of importance for the closing behaviour of a valve and the material law of the solid is rather simplistic for modelling the fiber reinforced leaflets. However, the purpose of this work was to propose a method which is flexible enough to accurately capture the movement of a solid in a fluid including a contact problem and which has the potential to being extended to 3D.

## Acknowledgments

This research is performed in the scope of the Hemodyn project, which is a cooperation between Philips Medical Systems Best, the Technical University Eindhoven and the Erasmus University Rotterdam, The Netherlands. It is financially supported by Senter (Dutch Ministry of Economic Affairs) in their TS subsidy program (Technologische Samenwerking).

## References

[1] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all speeds, J. Comput. Phys. 14 (1974) 227–253.
[2] J. Donea, S. Giuliani, J.P. Halleux, An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions, Comput. Meth. Appl. Mech. Eng. 33 (1982) 689–723.
[3] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[4] C.S. Peskin, The immersed boundary method, Act. Num. 11 (2002) 479–517.
[5] C.S. Peskin, D.M. McQueen, A three-dimensional computational method for blood flow in the heart I. Immersed elastic fibers in a viscous incompressible fluid The immersed boundary method, J. Comput. Phys. 81 (1989) 372–405.
[6] D.M. McQueen, C.S. Peskin, Shared-memory parallel vector implementation of the immersed boundary method for the computation of blood flow in the beating mamalian heart, J. Supercomput. 11 (1997) 213–236.
[7] R. Glowinski, T.-W. Pan, J. Périaux, A Lagrange multiplier/fictitious domain method for the numerical simulation of incompressible viscous flow around moving rigid bodies: (I) case where the rigid body motions are known a priori, C.R. Acad. Sci. Paris 25 (5) (1997) 361–369.
[8] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 324 (1999) 755–794.
[9] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, J. Périaux, A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: application to particulate flow, Int. J. Num. Meth. Fluids 30 (8) (1999) 1043–1066.
[10] F.P.T. Baaijens, A fictitious domain/mortar element method for fluid–structure interaction, Int. J. Num. Meth. Fluids 35 (7) (2001) 743–761.
[11] J. de Hart, G.W.M. Peters, P.J.G. Schreurs, F.P.T. Baaijens, A two-dimensional fluid–structure interaction model of the aortic value, J. Biomech. 33 (9) (2000) 1079–1088.
[12] J. de Hart, G.W.M. Peters, P.J.G. Schreurs, F.P.T. Baaijens, A three-dimensional computational analysis of fluid–structure interaction in the aortic valve, J. Biomech. 36 (1) (2003) 103–112.
[13] J. de Hart, G.W.M. Peters, P.J.G. Schreurs, F.P.T. Baaijens, A computational fluid–structure interaction analysis of a fiber-reinforced stentless aortic valve, J. Biomech. 36 (5) (2003) 699–712.
[14] J. de Hart, G.W.M. Peters, P.J.G. Schreurs, F.P.T. Baaijens, Collagen fibers reduce stresses and stabilize motion of aortic valve leaflets during systole, J. Biomech. 37 (2004) 303–311.

[15] Z. Yu, A DLM/FD method for fluid/flexible-body interactions, J. Comput. Phys. 207 (2005) 1–27.
[16] X. Shi, N. Phan-Thien, Distributed Lagrange multiplier/fictitious domain method in the framework of lattice Boltzmann method for fluid–structure interactions, J. Comput. Phys. 206 (2005) 81–94.
[17] J.M.A. Stijnen, J. de Hart, P.H.M. Bovendeerd, F.N. van de Vosse, Evaluation of a fictitious domain method for predicting dynamic response of mechanical heart valves, J. Fluid Struct. 19 (2004) 835–850.
[18] R. van Loon, P.D. Anderson, J. de Hart, F.P.T. Baaijens, A combined fictitious domain/adaptive meshing method for fluid–structure interaction in heart valves, Int. J. Num. Meth. Fluids 46 (2004) 533–544.
[19] R. van Loon, P.D. Anderson, F.P.T. Baaijens, F.N. van de Vosse, A three-dimensional fluid–structure interaction method for heart valve modelling, C.R. Mecanique 333 (2005) 856–866.
[20] N. Lock, M. Jaeger, M. Medale, R. Occelli, Local mesh adaptation technique for front tracking problems, Int. J. Num. Meth. Fluids 28 (1998) 719–736.
[21] D.A. Field, Laplacian smoothing and Delaunay triangulations, Commun. Appl. Num. Meth. 4 (1988) 709–712.
[22] G. Segal, SEPRAN Introduction, User's Manual, Programmer's Guide and Standard Problems, Ingenieursbureau SEPRA, Leidschendam, 2003.
[23] HSL, A collection of Fortran codes for large scale scientific computation. Available from: <http://www.numerical.rl.ac.uk/hsl>, 2002.