

RESEARCH ARTICLE

Open Access



Numerical simulation of solid deformation driven by creeping flow using an immersed finite element method

Thomas Rüberg^{1,2*} and José Manuel Garcí Aznar¹

*Correspondence:

thomas.rueberg@gmx.de

²Fundación ARAID, María de Luna 11, 50018 Zaragoza, Spain
Full list of author information is available at the end of the article

Abstract

An immersed finite element method for solid–fluid interaction is presented with application focus on highly deformable elastic bodies in a Stokes flow environment. The method is based on a global balance equation which combines the solid and fluid momentum balances, the fluid mass balance and, in weak form, the interface conditions. By means of an Updated Lagrangian description for finite elasticity, only one analysis mesh is used, where the solid particles are backtracked in order to preserve the deformation history. The method results in a full coupling of the solid–fluid system which is solved by an exact Newton method. The location of the material interface is captured by a signed distance function and updated according to the computed displacement increments and the help of an explicit surface parameterisation; no body-fitted volume meshes are needed. Special emphasis is placed on the accurate integration of finite elements traversed by the interface and the related numerical stability of the shape function basis. A number of applications for compressible Neo-Hookean solids subject to creeping flow are presented, motivated by microfluidic experimentation in mechanobiology.

Keywords: Immersed finite elements, Fluid–solid interaction, Updated Lagrangian method, Computational mechanobiology, Nitsche’s method

Background

In a large class of biological and biomedical problems, the interaction of fluid flow and highly deformable solids plays an important role. Consider, as examples, the biological response of cells to a mechanical stimulus [1], the flow of capsules in a narrowed tube [2], the deformation of thin-walled blood vessels [3], or the motion of red blood cells [4]. Not only the mechanical response of cells and tissues in a fluid environment is of great interest, but also the biological implications of the mechanical environment (that is, *mechanobiology*). It is, for instance, suggested that fluid shear stresses can control the phenotype of a living cell: to regulate the cell’s differentiation or proliferation, or simply damage it, see for example [5] for endothelial cells. To this end, microfluidic experimentation [6] has proven a powerful tool for the *in vitro* analysis of such phenomena because it allows to precisely control the mechanical and chemical environment of individual cells or aggregates thereof. Nevertheless, this methodology is expensive and time-consuming. Therefore, *computational mechanobiology* [7] often proves to be a promising alternative to assist the

experimentation. For the considered field of applications, inertia forces are negligible (flow with very low Reynolds numbers, $Re \ll 1$) and the ratio of elastic to viscous stresses is by orders of magnitude smaller than in systems akin to aeroelasticity (for example, heaving wings [8] and insect flight [9]). For the modelling of living cells in a continuum mechanical framework, the most common approaches are liquid-filled, elastic membranes or (visco-)elastic solids. Both descriptions have their merits and both can be supported or rejected by experiments [10]. In this work, we consider a solid domain occupied by a homogeneous compressible Neo-Hookean material that is subject to Stokes flow. This class of problems yields an initial, but generic approach to above range of applications with the potential of future extension.

Over the last decades a large quantity of methods has been developed for the numerical analysis of fluid-structure interaction problems (see, for instance, the journal issues [11–13] dedicated entirely to this topic, or the monograph [14]). Broadly speaking, these can be classified into *monolithic* methods, e.g. [15], and *iterative coupling* approaches, e.g. [16,17]. Whereas in the former class of methods a fully coupled system of equations is formulated, in the latter class the equations of fluid and solid domains are solved independently and provide the boundary conditions (velocities or forces) to the respectively other domain. Although this iterative approach bears the potential of higher numerical efficiency and software modularity, the stability of the iteration process often requires severe restrictions of the size of the time step or sub-iterations [18]. In order to avoid this intricacy, the approach presented here is monolithic.

Another important classification is to divide the methods into those using *body-fitted* meshes and their counterpart, immersed methods [19]. A body-fitted mesh conforms to the geometry of the problem, this means that the interface between solid and fluid is solved and tracked by the mesh [20], see also [21] for space-time formulation. The advantage of this approach is a simplified implementation of interface conditions and has been successfully applied to a wide range of applications: biomedical (for instance, airways and arteries [3,22]), parachutes [23,24], or the iso-geometric analysis of wind turbines [25]. Nevertheless, the use of body-fitted meshes has the downside that large solid deformations can lead to a virtual destruction of the analysis mesh. Even though this problem is addressed by the ALE formalism [26], there are situations where a body-fitted mesh cannot be maintained. Tedious re-meshing and solution mapping techniques are often the consequence of the severe mesh distortions and are avoided by the immersed techniques. The most prominent approach with non-body-fitted meshes is the *immersed boundary method* [27] and tailored towards elastic surfaces subject to a flow environment. It has been successfully applied to simulations of blood flow and the deformation of red blood cells [28]. Moreover, a finite element counterpart has been developed in [29]. Despite its great success, the numerical stability of the immersed boundary method is subject to severe restrictions of the time step [27] due to its mainly explicit character. Apart from the mentioned techniques which rely domain discretisations, the use of boundary integral equations is a popular alternative [30]. Due to the surface-only description of the flow field, the numerical cost can be reduced significantly in this class of methods. See, for instance, the simulation of vesicle flow [31,32] where an unbounded fluid domain and liquid-filled membranes are used as model problem. Nevertheless, there are restrictions on the use of boundary integral equations: they rely on a homogeneous Stokes flow and any extension to non-linear flow behaviour is not straightforward; moreover, the robust implementation

of singular integral equations is a demanding task. The application target of the aforementioned methods is mostly the interaction between a fluid and a reduced solid, such as a membrane or a shell. Here we focus on voluminous solids instead as, for instance, in [33,34], where an explicit finite difference method is presented for this problem class.

The here presented approach falls into the class of finite element methods with immersed boundaries and the spatial discretisation does not conform with the location of the solid-fluid interface. Henceforth, we refer to our method as an *immersed finite element* method even though the term has already been employed in [29] (see also, [35]) for a finite element analogy to the immersed boundary method of [27], which basically consists of an explicit coupling scheme where structural forces are imposed on the fluid mesh and the computed velocities modify the structural configuration. The method presented in [33] and typically the immersed boundary method do not sharply resolve the interface but work with a transition zone between the two media that depends on the grid resolution. The method proposed here works with a so-called *sharp* interface representation [19] and is based on the immersed b-spline finite element method proposed in [36]; see also [37,38] for finite element methods with embedded interfaces. Therein, the application to viscous fluid flow with moving boundaries has been targeted and later carried over to the partitioned analysis of fluid-structure interaction in [8]. In this latter reference, the structure is resolved by a Lagrangian mesh and the fluid with a Eulerian mesh in which an implicit geometry representation is superimposed in order to capture the interface. Here, this idea is pushed further by using only one analysis mesh for both solid and fluid, in the spirit of the methods developed in [39,40], which are based on a fully Eulerian description, see also [41–43]. The drawback of fully Eulerian approaches for the solid is that displacement and velocity become independent fields that are coupled by an additional advection equation which needs to be solved [42]. In order to avoid this additional equation and have solid velocities as an extra field, we choose an *Updated Lagrangian method* [44] combined with particle tracking, as advertised in [45] for plasticity problems with large deformations. Moreover, this choice of expressing the solid equilibrium in the latest known configuration avoids the need of shape derivatives [46] in the linearisation process for the used Newton method. Under the assumption of small velocities, the time derivatives are completely neglected from the field equations and only appear through the interface condition between solid and fluid. This condition is weakly incorporated using Nitsche's method [47], see also [8,46] for this technique in the context of an iterative coupling method. A signed distance function is used to capture the location of the interface as a level set [48] for the field computation. At the end of every time step an explicit surface parameterisation is generated and moved with the computed displacement increment. Note that we combine explicit and implicit geometry descriptions; the implicit version is used for a quick interface detection at the beginning of each simulation step. Falling back to a temporary explicit description allows to accurately trace the interface location without the need of additional advection equations as common in level set methods [48]. We thereby avoid possible distortions of the surface mesh when using an entirely explicit surface description.

Several test applications demonstrate the potential of the method. First, the convergence behaviours of the updated Lagrangian method for a solid-only problem and a fully coupled fluid-solid interaction problem are studied. Next, the cases of a solid subject to shear flow, the flow through a constricted pipe section and driven cavity flow are analysed numerically.

Based on the computed field data, several quantities of interest are studied, such as the shape variation of the immersed solid, particle motion, Cauchy stresses in fluid and solid, and fluid velocity patterns.

The outline of this article is as follows. In “Fluid–solid interaction” section, the basic balance equations of solid and fluid are presented and a global solid–fluid balance equation that incorporates the interface conditions weakly is derived. Other than customary, we first discretise this expression in time and linearise it before introducing the spatial discretisation in “Immersed finite element method” section. More specifically, in “Updated Lagrangian method and linearisation” section the linearisation is based on an Updated Lagrangian formalism that avoids the need of shape derivatives. Regarding the immersed finite element discretisation, a special emphasis is placed on the integration and numerical stability related to the elements which are traversed by the interface in “Cut elements” section. Moreover, the displacement history is maintained with a particle tracking algorithm presented in “Interface update and particle tracking” section. Finally, in “Example applications” section several example applications are presented which demonstrate the potential of the new method.

Fluid–solid interaction

We begin by deriving our coupling formulation from the balances of momentum and mass of the solid and the fluid parts, respectively. Here, we restrict ourselves to the case of very low Reynolds numbers such that inertia terms and fluid advection can be safely discarded. This simplification is justified by the fact that viscous forces are significantly larger than inertial forces [49].

Static balance laws

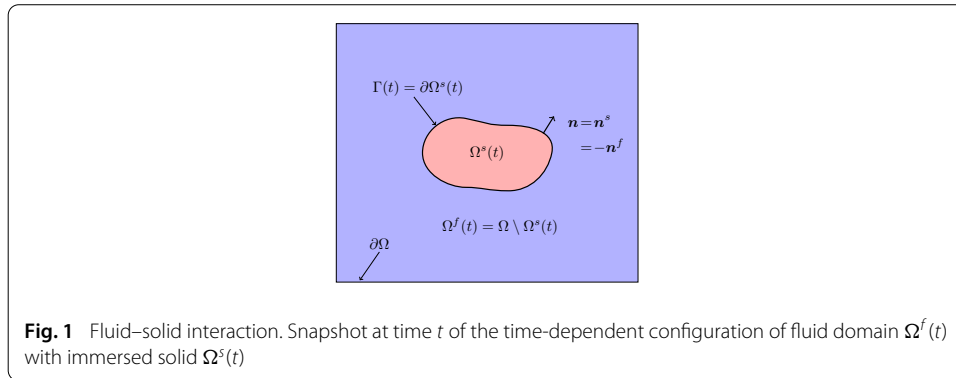
The balance laws for solid and fluid can be found, among others, in [50–52]. The main equations important for this work are presented in this section and help to introduce the chosen notation. Let $\Omega^s(t) \subset \mathbb{R}^D$ ($D = 2$ or 3 denotes the spatial dimension) be occupied by a hyperelastic solid and immersed in a control volume domain Ω at the time t . The remainder $\Omega^f(t) = \Omega \setminus \Omega^s(t)$ of this volume is occupied by a viscous incompressible fluid. The boundary of the solid domain is denoted by $\Gamma(t)$ and the solid is assumed to be strictly inside Ω at all times, such that $\partial\Omega \cap \Gamma(t) = \emptyset$. The outward unit normal vector to the solid domain is denoted by \mathbf{n} . This vector refers to the current configuration is therefore time-dependent, but we omit to write $\mathbf{n}(t)$ for simplicity. This choice of notation is illustrated in Fig. 1 and it has to be emphasised that the configuration of the domains is time-dependent: the interaction between solid and fluid changes shape and location of $\Omega^s(t)$ and thus $\Omega^f(t)$. But the outer boundary $\partial\Omega$ is assumed to remain fixed. In view of the considered applications, we neglect inertia and advection terms and formulate the following local balance laws

$$-\nabla \cdot \boldsymbol{\sigma}^s(\mathbf{d}) = \mathbf{0} \quad \mathbf{x} \in \Omega^s(t) \quad (1)$$

$$-\nabla \cdot \boldsymbol{\sigma}^f(\mathbf{u}, p) = \mathbf{0} \quad \mathbf{x} \in \Omega^f(t) \quad (2)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \mathbf{x} \in \Omega^f(t). \quad (3)$$

Here, $\boldsymbol{\sigma}^s$ and $\boldsymbol{\sigma}^f$ are the Cauchy stresses in the solid and the fluid, respectively, and we have the primal field variables solid displacement \mathbf{d} , fluid velocity \mathbf{u} and fluid pressure p ;



all functions of the spatial coordinate \boldsymbol{x} and time t . The spatial divergence operation is symbolised by $\nabla \cdot (\cdot)$. Equations (1) and (2) are the quasi-static momentum balances in solid and fluid, and (3) is the balance of mass for an incompressible fluid.

The fluid is Newtonian and the solid is assumed to be hyperelastic. Therefore, the above introduced stresses fulfil the material laws

$$\boldsymbol{\sigma}^f(\boldsymbol{u}, p) = -p\boldsymbol{I} + 2\mu^f \boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{4}$$

$$\boldsymbol{\sigma}^s(\boldsymbol{d}) = \frac{1}{\det \boldsymbol{F}} \frac{\partial W(\boldsymbol{F})}{\partial \boldsymbol{F}} \boldsymbol{F}^\top \quad \text{with} \quad \boldsymbol{F} = \boldsymbol{F}(\boldsymbol{d}), \tag{5}$$

where \boldsymbol{I} is the D -dimensional identity tensor, μ^f the fluid’s dynamic viscosity, $\boldsymbol{\varepsilon}$ denotes the symmetric gradient, \boldsymbol{F} the deformation gradient and W the strain energy density function [50,53].

On the current location of the interface $\Gamma(t)$ between solid and fluid, we have the conditions

$$\dot{\boldsymbol{d}} := \frac{\partial \boldsymbol{d}}{\partial t} = \boldsymbol{u} \quad \text{and} \quad \boldsymbol{\sigma}^s \boldsymbol{n} = \boldsymbol{\sigma}^f \boldsymbol{n}, \tag{6}$$

which are commonly referred to as *continuity* and *equilibrium* conditions. It remains to specify boundary conditions for the fluid boundary $\partial\Omega$. Here, either of the two possibilities

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \quad \text{or} \quad \boldsymbol{\sigma}^f \boldsymbol{n} = \mathbf{0} \tag{7}$$

is used which are a prescribed flow velocity or an *outflow* condition.

Equations (1), (2), (3) together with the material laws, (4) and (5), and the interface and boundary conditions, (6) and (7), completely describe the problem. Note that even though we work with the static balance laws (neglected inertia), the problem is time-dependent due to the change of the configurations: $\Omega^s(t)$ and $\Omega(t)$ are functions of \boldsymbol{d} and thus of time. Due to this dependency, the problem is non-linear in addition to the nonlinearity given by the solid stresses (4). Moreover, the first condition in (6) couples solid with fluid velocities and thus renders the solid sub-problem time-dependent.

Coupling formulation

Here, a global balance law for fluid and solid is derived which does not rely on essential boundary conditions. This means that the interface conditions (6) are not directly fulfilled by the choice of the finite element basis, which will be later introduced in Section “Immersed finite element method”. Its main advantage is to allow for a configuration of

solid and fluid that changes independently of the used finite element mesh. Nevertheless, the conditions imposed on $\partial\Omega$ for the outer fluid boundary are treated in the classic way by imposing $\mathbf{u} = \bar{\mathbf{u}}$ essentially and $\boldsymbol{\sigma}^f \mathbf{n} = \mathbf{0}$ naturally. Since this boundary does not move, there is no disadvantage of this standard approach.

We begin with a weighted residual approach using test functions $\delta \mathbf{d}$, $\delta \mathbf{u}$ and δp with the only condition that $\delta \mathbf{u} = \mathbf{0}$ on the parts of $\partial\Omega$ where the condition $\mathbf{u} = \bar{\mathbf{u}}$ is applied. Other than that, these test functions are not restricted by any condition. Weighting the solid momentum balance (1) by $\delta \mathbf{d}$, the fluid momentum balance (2) by $\delta \mathbf{u}$, and the fluid mass balance (3) by δp and application of the divergence theorem gives

$$\int_{\Omega^s(t)} \boldsymbol{\sigma}^s(\mathbf{d}) : \boldsymbol{\varepsilon}(\delta \mathbf{d}) \, d\Omega + \int_{\Omega^f(t)} \boldsymbol{\sigma}^f(\mathbf{u}, p) : \boldsymbol{\varepsilon}(\delta \mathbf{u}) \, d\Omega - \int_{\Omega^f(t)} \delta p(\nabla \cdot \mathbf{u}) \, d\Omega - \int_{\Gamma(t)} [\boldsymbol{\sigma}^s(\mathbf{d})\mathbf{n} \cdot \delta \mathbf{d} - \boldsymbol{\sigma}^f(\mathbf{u}, p)\mathbf{n} \cdot \delta \mathbf{u}] \, d\Gamma = 0. \tag{8}$$

The boundary terms on $\partial\Omega$ have been dropped due the treatment of boundary conditions as discussed above. For sake of legibility, we introduce the following abbreviations

$$a^s(\mathbf{d}; \delta \mathbf{d}) = \int_{\Omega^s(t)} \boldsymbol{\sigma}^s(\mathbf{d}) : \boldsymbol{\varepsilon}(\delta \mathbf{d}) \, d\Omega \tag{9}$$

$$a^f(\mathbf{u}, p; \delta \mathbf{u}, \delta p) = \int_{\Omega^f(t)} \boldsymbol{\sigma}^f(\mathbf{u}, p) : \boldsymbol{\varepsilon}(\delta \mathbf{u}) \, d\Omega - \int_{\Omega^f(t)} \delta p(\nabla \cdot \mathbf{u}) \, d\Omega \tag{10}$$

$$a^\Gamma(\mathbf{d}, \mathbf{u}, p; \delta \mathbf{d}, \delta \mathbf{u}, \delta p) = \int_{\Gamma(t)} [\boldsymbol{\sigma}^s(\mathbf{d})\mathbf{n} \cdot \delta \mathbf{d} - \boldsymbol{\sigma}^f(\mathbf{u}, p)\mathbf{n} \cdot \delta \mathbf{u}] \, d\Gamma, \tag{11}$$

such that Equation (8) can be recast in the form

$$a^{fsi}(\mathbf{d}, \mathbf{u}, p; \delta \mathbf{d}, \delta \mathbf{u}, \delta p) = a^s(\mathbf{d}; \delta \mathbf{d}) + a^f(\mathbf{u}, p; \delta \mathbf{u}, \delta p) - a^\Gamma(\mathbf{d}, \mathbf{u}, p; \delta \mathbf{d}, \delta \mathbf{u}, \delta p) = 0. \tag{12}$$

The next step is to incorporate conditions (6) into this balance equation. Observe that the integrand in expression (11) is the difference of a product. One can show that

$$\begin{aligned} &\boldsymbol{\sigma}^s \mathbf{n} \cdot \delta \mathbf{d} - \boldsymbol{\sigma}^f \mathbf{n} \cdot \delta \mathbf{u} \\ &= \left(\beta \boldsymbol{\sigma}^s \mathbf{n} + (1 - \beta) \boldsymbol{\sigma}^f \mathbf{n} \right) \cdot (\delta \mathbf{d} - \delta \mathbf{u}) + \left((1 - \beta) \delta \mathbf{d} + \beta \delta \mathbf{u} \right) \cdot (\boldsymbol{\sigma}^s \mathbf{n} - \boldsymbol{\sigma}^f \mathbf{n}) \end{aligned} \tag{13}$$

for any $\beta \in \mathbb{R}$. Hence, we obtain for the interface term

$$\begin{aligned} a^\Gamma(\mathbf{d}, \mathbf{u}, p; \delta \mathbf{d}, \delta \mathbf{u}, \delta p) &= \int_{\Gamma(t)} \left(\beta \boldsymbol{\sigma}^s(\mathbf{d})\mathbf{n} + (1 - \beta) \boldsymbol{\sigma}^f(\mathbf{u}, p)\mathbf{n} \right) \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma \\ &\quad + \int_{\Gamma(t)} \left((1 - \beta) \delta \mathbf{d} + \beta \delta \mathbf{u} \right) \cdot (\boldsymbol{\sigma}^s(\mathbf{d})\mathbf{n} - \boldsymbol{\sigma}^f(\mathbf{u}, p)\mathbf{n}) \, d\Gamma. \end{aligned} \tag{14}$$

Note that the second integral is zero due to interface equilibrium (6)₂ and is now removed from this expression. With reference to Nitsche’s method [37, 47, 54] for the incorporation of Dirichlet boundary conditions in a weak form, two new terms are added to this result

$$\begin{aligned} a^\Gamma(\mathbf{d}, \mathbf{u}, p; \delta \mathbf{d}, \delta \mathbf{u}, \delta p) &= \int_{\Gamma(t)} \left(\beta \boldsymbol{\sigma}^s(\mathbf{d})\mathbf{n} + (1 - \beta) \boldsymbol{\sigma}^f(\mathbf{u}, p)\mathbf{n} \right) \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma \\ &\quad + \int_{\Gamma(t)} \left(\beta \tilde{\boldsymbol{\sigma}}^s(\delta \mathbf{d})\mathbf{n} + (1 - \beta) \boldsymbol{\sigma}^f(\delta \mathbf{u}, \delta p)\mathbf{n} \right) \cdot (\dot{\mathbf{d}} - \mathbf{u}) \, d\Gamma \\ &\quad - \gamma \int_{\Gamma(t)} (\dot{\mathbf{d}} - \mathbf{u}) \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma. \end{aligned} \tag{15}$$

Note that the same expression has been derived in [25] via a Lagrange multiplier approach and a subsequent elimination of the multipliers. Here, $\gamma > 0$ is some parameter which will

be determined in section “Cut elements” and $\bar{\sigma}^s$ denotes a stress-like function of the test displacements $\delta \mathbf{d}$. Note that γ turns out to be dependent on the choice of the finite element discretisation, as shown already in [37] and for this reason its specification is postponed. Inserting expression (15) into the balance equation (12) gives a family of formulations for fluid–solid coupling based on Nitsche’s method [16,25] which is parameterised by the number β . From now on, we fix $\beta = 0$ and consider only the interface term

$$a^\Gamma(\mathbf{d}, \mathbf{u}, p; \delta \mathbf{d}, \delta \mathbf{u}, \delta p) = \int_{\Gamma(t)} \boldsymbol{\sigma}^f(\mathbf{u}, p) \mathbf{n} \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma + \int_{\Gamma(t)} \boldsymbol{\sigma}^f(\delta \mathbf{u}, \delta p) \mathbf{n} \cdot (\dot{\mathbf{d}} - \mathbf{u}) \, d\Gamma - \gamma \int_{\Gamma(t)} (\dot{\mathbf{d}} - \mathbf{u}) \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma. \tag{16}$$

Hence, the expression comprises a fluid-sided “mortaring” as coined in [16], see also [25] for a motivation of this choice of parameter. On the other hand, we refer [38,55] for an analysis of the choice of this parameter β for interface problems in which both domains are governed by the same mathematical model. Inserting this interface term (16) into expression (12) gives the desired global fluid–solid balance equation that incorporates the interface conditions (6). The time semi-discretisation and linearisation of this problem are given in the remainder of this section, whereas the finite element space discretisation is introduced in “Immersed finite element method” section.

Time semi-discretisation

The fluid–solid balance (12) as derived in the previous section is time-dependent and nonlinear. Typically, such expressions are discretised by following the concept of the method of lines: the discretisation in space leads to a system of (nonlinear) ODEs which is then tackled by a time discretisation. Here, we reverse this order and make use of what is referred to as Rothe’s method [56]. The aim of this work is to use a fixed, stationary finite element mesh in which the fluid-solid domain configuration moves freely. Nevertheless the finite element spaces vary in function of this configuration and, for this reason, it is preferred to begin with time discretisation and linearisation before finally applying a spatial discretisation with in finite elements as outlined in “Immersed finite element method” section.

For simplicity, the Euler backward method [57] is used for time stepping although the presented approach is not restricted to this choice. Therefore, let a specific time instant be denoted by t_n and the size of the current time step by Δt . Moreover, the approximation of the principal unknown fields at a time instant, are indicated by the same subscript; for instance, $\mathbf{d}_n(\mathbf{x})$ approximates $\mathbf{d}(\mathbf{x}, t_n)$. Based on the Euler-backward scheme, the displacement velocity becomes

$$\dot{\mathbf{d}}_{n+1} = \frac{\mathbf{d}_{n+1} - \mathbf{d}_n}{\Delta t}. \tag{17}$$

Using this notation, the non-linear problem to find the system state $(\mathbf{d}_{n+1}, \mathbf{u}_{n+1}, p_{n+1})$ reads

$$a_{n+1}^s(\mathbf{d}_{n+1}; \delta \mathbf{d}) + a_{n+1}^f(\mathbf{u}_{n+1}, p_{n+1}; \delta \mathbf{u}, \delta p) - \int_{\Gamma_{n+1}} \boldsymbol{\sigma}^f(\mathbf{u}_{n+1}, p_{n+1}) \mathbf{n} \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma - \int_{\Gamma_{n+1}} \boldsymbol{\sigma}^f(\delta \mathbf{u}, \delta p) \mathbf{n} \cdot \left(\frac{\mathbf{d}_{n+1}}{\Delta t} - \mathbf{u}_{n+1} \right) \, d\Gamma + \gamma \int_{\Gamma_{n+1}} \left(\frac{\mathbf{d}_{n+1}}{\Delta t} - \mathbf{u}_{n+1} \right) \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma = \gamma \int_{\Gamma_{n+1}} \frac{\mathbf{d}_n}{\Delta t} \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma - \int_{\Gamma_{n+1}} \boldsymbol{\sigma}^f(\delta \mathbf{u}, \delta p) \mathbf{n} \cdot \frac{\mathbf{d}_n}{\Delta t} \, d\Gamma, \tag{18}$$

where the current state $(\mathbf{d}_n, \mathbf{u}_n, p_n)$ is given. Here, the subscripts $n + 1$ at the solid and fluid domain contributions, a^s and a^f , and at the interface Γ emphasise that the position of solid and fluid domain at the new time instant t_{n+1} is considered. Note that this configuration is unknown since it depends on the displacement field \mathbf{d}_{n+1} . Problem (18) is therefore nonlinear due to the solid contribution a^s and the implicit dependence on the domain configuration.

Updated Lagrangian method and linearisation

A full linearisation of the nonlinear problem (18) evidently leads to shape derivatives [46]. We aim to avoid this complexity by expressing the balance law (18) not in the unknown spatial configuration, but in the latest known configuration. This approach, also known as *Updated Lagrangian*, is well established for large-deformation analysis, see [44]. Moreover, the unknown interface location Γ_{n+1} in (18) is simply replaced by the known interface Γ_n , see e.g. [40]. Based on this explicit treatment of the fluid–solid interface, its location is known in every time step and updated after the new system state $(\mathbf{d}_{n+1}, \mathbf{u}_{n+1}, p_{n+1})$ has been determined. In the following, only the solid state is considered. For the treatment of the fluid part, we simply disregard the discrepancies between the unknown and the latest known configurations. A thorough error analysis of this approach is still pending, but the results given in “Example applications” section convey that this defect is not detrimental to the overall approach.

At first, consider the three solid domain configurations occurring in the Updated Lagrangian Method: the initial configuration Ω_0^s , the latest known configuration Ω_n^s and the unknown configuration Ω_{n+1}^s , see Fig. 2. Coordinates in these configurations are denoted with the same subscript, for instance, $\mathbf{x}_n \in \Omega_n$. The maps are between two configurations, Ω_a and Ω_b are denoted by $\varphi_a^b : \Omega_a \rightarrow \Omega_b$. For instance, the coordinate \mathbf{x}_{n+1} results from either mapping from the initial or the latest known configuration, i.e., $\mathbf{x}_{n+1} = \varphi_0^{n+1}(\mathbf{x}_0) = \varphi_n^{n+1}(\mathbf{x}_n)$. At last, the deformation gradients are defined as $\mathbf{F}_a^b = \partial \mathbf{x}_b / \partial \mathbf{x}_a$ and are maps between the respective tangent spaces. We refer to [44, 50, 53] for more details on large elastic deformations and the notion of configurations.

The deformations φ_0^n and φ_0^{n+1} in Fig. 2 are represented by the Lagrangian displacements \mathbf{d}_n and \mathbf{d}_{n+1} . Consider the solid contribution

$$a_{n+1}^s(\mathbf{d}_{n+1}; \delta \mathbf{d}) = \int_{\Omega_{n+1}^s} \boldsymbol{\sigma}^s(\mathbf{d}_{n+1}) : \boldsymbol{\varepsilon}(\delta \mathbf{d}) \, d\Omega = \int_{\Omega_{n+1}^s} \boldsymbol{\sigma}^s(\mathbf{d}_{n+1}) : \nabla_{n+1} \delta \mathbf{d} \, d\Omega, \quad (19)$$

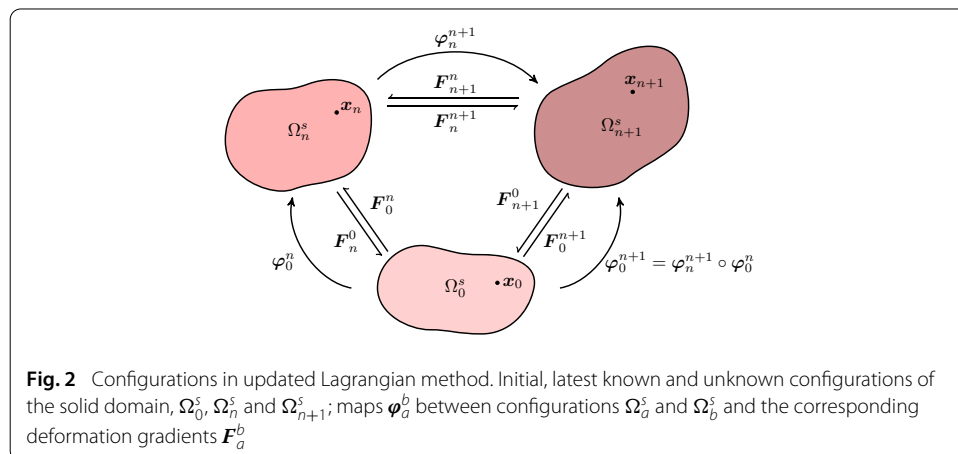


Fig. 2 Configurations in updated Lagrangian method. Initial, latest known and unknown configurations of the solid domain, Ω_0^s , Ω_n^s and Ω_{n+1}^s ; maps φ_a^b between configurations Ω_a^s and Ω_b^s and the corresponding deformation gradients \mathbf{F}_a^b

where the subscript to the ∇ -operator indicates the coordinate with respect to which the differentiation is carried out. This expression is now mapped to the latest known configuration Ω_n^s ,

$$\begin{aligned} a_{n+1}^s(\mathbf{d}_{n+1}; \delta \mathbf{d}) &= \int_{\Omega_n^s} (\boldsymbol{\sigma}(\mathbf{d}_{n+1}) : \nabla_{n+1} \delta \mathbf{d}) \circ \boldsymbol{\varphi}_n^{n+1} \det \mathbf{F}_n^{n+1} \, d\Omega \\ &= \int_{\Omega_n^s} \left[\det \mathbf{F}_n^{n+1} \left(\boldsymbol{\sigma}(\mathbf{d}_{n+1}) \circ \boldsymbol{\varphi}_n^{n+1} \right) (\mathbf{F}_n^{n+1})^{-\top} \right] : (\nabla_n (\delta \boldsymbol{\varphi} \circ \boldsymbol{\varphi}_n^{n+1})) \, d\Omega \\ &= \int_{\Omega_n^s} \mathbf{P}_n^{n+1}(\mathbf{d}_{n+1}) : \nabla_n \delta \mathbf{d} \, d\Omega. \end{aligned} \tag{20}$$

Here, the first line is obtained from (19) by mapping the integration domain, the second line results from the chain rule in order to change from ∇_{n+1} to ∇_n , and the last line introduces a new stress tensor \mathbf{P}_n^{n+1} . Note that for $n = 0$, \mathbf{P}_0^{n+1} coincides with the standard definition of the first Piola-Kirchhoff stress tensor [53]. Moreover, with an abuse of notation the composition of the test displacements $\delta \mathbf{d}$ with the map $\boldsymbol{\varphi}_n^{n+1}$ is also denoted by $\delta \mathbf{d}$. Now, the result of (20) is introduced in (18) and the explicit treatment of the interface used. The nonlinear fluid–solid balance now reads

$$\begin{aligned} &\int_{\Omega_n^s} \mathbf{P}_n^{n+1}(\mathbf{d}_{n+1}) : \nabla_n \delta \mathbf{d} \, d\Omega + \int_{\Omega_n^f} \boldsymbol{\sigma}^f(\mathbf{u}_{n+1}, p_{n+1}) : \boldsymbol{\varepsilon}(\delta \mathbf{u}) \, d\Omega - \int_{\Omega_n^f} \delta p (\nabla \cdot \mathbf{u}_{n+1}) \, d\Omega \\ &- \int_{\Gamma_n} \boldsymbol{\sigma}^f(\mathbf{u}_{n+1}, p_{n+1}) \mathbf{n} \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma - \int_{\Gamma_n} \boldsymbol{\sigma}^f(\delta \mathbf{u}, \delta p) \mathbf{n} \cdot \left(\frac{\mathbf{d}_{n+1}}{\Delta t} - \mathbf{u}_{n+1} \right) \, d\Gamma \\ &+ \gamma \int_{\Gamma_n} \left(\frac{\mathbf{d}_{n+1}}{\Delta t} - \mathbf{u}_{n+1} \right) \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma = \gamma \int_{\Gamma_n} \frac{\mathbf{d}_n}{\Delta t} \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma \end{aligned} \tag{21}$$

and only its first term, the solid domain contribution, remains nonlinear. A Newton method [53] is applied to this expression. Let k be the iteration counter placed as a left superscript and $\Delta \mathbf{d}$ the unknown displacement increment such that the latest iterate for the unknown \mathbf{d}_{n+1} becomes

$${}^{k+1} \mathbf{d}_{n+1} = {}^k \mathbf{d}_{n+1} + \Delta \mathbf{d}. \tag{22}$$

Note that (21) is linear in \mathbf{u}_{n+1} and p_{n+1} such that we can directly work with ${}^{k+1} \mathbf{u}_{n+1}$ and ${}^{k+1} p_{n+1}$ as unknowns without increments. Linearisation of the integrand of the solid part in direction of the increment $\Delta \mathbf{d}$ gives

$$\begin{aligned} \mathbb{D} \left({}^k \mathbf{P}_n^{n+1} : \nabla_n \delta \mathbf{d} \right) [\Delta \mathbf{d}] &= \left[(\nabla_n \delta \mathbf{d}) \cdot \left({}^k \mathbf{F}_n^{n+1} \right)^{-1} \cdot {}^k \mathbf{P}_n^{n+1} \right] : \nabla_n \Delta \mathbf{d} \\ &+ \det {}^k \mathbf{F}_n^{n+1} \left[(\nabla_n \delta \mathbf{d}) \cdot \left({}^k \mathbf{F}_n^{n+1} \right)^{-1} \right] : {}^k \mathbb{C}_{n+1} : \left[\nabla_n \Delta \mathbf{d} \cdot \left({}^k \mathbf{F}_n^{n+1} \right)^{-1} \right], \end{aligned} \tag{23}$$

where \mathbb{C}_{n+1} refers to the material elasticity tensor [50,53]. Note that expression (23) requires the deformation gradient ${}^k \mathbf{F}_n^{n+1}$ and, moreover, the material evaluations for ${}^k \mathbf{P}_n^{n+1}$ and ${}^k \mathbb{C}_{n+1}$ are based on the deformation gradient ${}^k \mathbf{F}_0^{n+1}$. We note that

$${}^k \mathbf{F}_0^{n+1} = {}^k \mathbf{F}_n^{n+1} \cdot \mathbf{F}_0^n = {}^k \mathbf{F}_n^{n+1} \cdot \left(\mathbf{F}_n^0 \right)^{-1} \tag{24}$$

whose factors are furthermore computed by means of

$$\begin{aligned} {}^k \mathbf{F}_n^{n+1} &= \frac{\partial}{\partial \mathbf{x}_n} \left[\mathbf{x}_n + \left({}^k \mathbf{d}_{n+1} - \mathbf{d}_n \right) \right] = \mathbf{I} + \nabla_n \left({}^k \mathbf{d}_{n+1} - \mathbf{d}_n \right) \\ \mathbf{F}_n^0 &= \frac{\partial}{\partial \mathbf{x}_n} \left[\mathbf{x}_n - \mathbf{d}_n \right] = \mathbf{I} - \nabla_n \mathbf{d}_n. \end{aligned} \tag{25}$$

Now, the linearisation process can be summarised and the following Newton step

$$\begin{aligned}
 & \int_{\Omega_n^s} \mathbf{D} \left({}^k \mathbf{P}_n^{n+1} : \nabla_n \delta \mathbf{d} \right) [\Delta \mathbf{d}] \, d\Omega + \int_{\Omega_n^f} \boldsymbol{\sigma}^f \left({}^{k+1} \mathbf{u}_{n+1}, {}^{k+1} p_{n+1} \right) : \boldsymbol{\varepsilon}(\delta \mathbf{u}) \, d\Omega \\
 & - \int_{\Omega_n^f} \delta p (\nabla \cdot {}^{k+1} \mathbf{u}_{n+1}) \, d\Omega - \int_{\Gamma_n} \boldsymbol{\sigma}^f \left({}^{k+1} \mathbf{u}_{n+1}, {}^{k+1} p_{n+1} \right) \mathbf{n} \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma \\
 & - \int_{\Gamma_n} \boldsymbol{\sigma}^f(\delta \mathbf{u}, \delta p) \mathbf{n} \cdot \left(\frac{\Delta \mathbf{d}}{\Delta t} - {}^{k+1} \mathbf{u}_{n+1} \right) \, d\Gamma + \gamma \int_{\Gamma_n} \left(\frac{\Delta \mathbf{d}}{\Delta t} - {}^k \mathbf{u}_{n+1} \right) \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma \\
 & = - \int_{\Omega_n^s} {}^k \mathbf{P}_n^{n+1} : \nabla_n \delta \mathbf{d} \, d\Omega + \gamma \int_{\Gamma_n} \frac{\mathbf{d}_n}{\Delta t} \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma \\
 & + \int_{\Gamma_n} \boldsymbol{\sigma}^f(\delta \mathbf{u}, \delta p) \mathbf{n} \cdot \frac{{}^k \mathbf{d}_{n+1}}{\Delta t} \, d\Gamma - \gamma \int_{\Gamma_n} \frac{{}^k \mathbf{d}_{n+1}}{\Delta t} \cdot (\delta \mathbf{d} - \delta \mathbf{u}) \, d\Gamma \tag{26}
 \end{aligned}$$

followed by the update (22). This last expression represents a time-discretised and linearised version of the global fluid–solid balance (12) which incorporates the interface conditions (6). The remaining step for a numerical solution is the spatial discretisation by finite elements as outlined in the following section.

Immersed finite element method

The global fluid–solid balance equation (12) and its time-discretised and linearised version (26) are perfectly suited for an immersed finite element method [19]. Note that here *immersed* solely refers to the fact that the interface location is independent of the finite element mesh and we do not refer to the method of [29]. Observe that Dirichlet boundary conditions only appear on the boundary $\partial\Omega$ which is fixed in space and time, and that the interface conditions (6) have been incorporated in a weak form. This implies that the finite element spaces need only be equipped with essential boundary conditions on $\partial\Omega$ but are not affected by the specific location of the interface Γ .

In the implementation, quadrilateral elements in two and hexahedrons in three dimensions are used. For the solid displacement piece-wise linear Lagrange polynomials are used (bi- and tri-linear, to be precise), for the fluid velocity quadratic and for the pressure linear functions. For the system state at time instant t_n the finite element approximation has the form

$$\begin{aligned}
 \mathbf{d}_n(\mathbf{x}) & \approx \sum_i \mathbf{d}_{n,i} N_i^{\mathbf{d}}(\mathbf{x}), \quad \mathbf{u}_n(\mathbf{x}) \approx \sum_j \mathbf{u}_{n,j} N_j^{\mathbf{u}}(\mathbf{x}), \quad \text{and} \\
 p_n(\mathbf{x}) & \approx \sum_k p_{n,k} N_k^p(\mathbf{x}). \tag{27}
 \end{aligned}$$

The specific choice for the discretisation of the fluid variables \mathbf{u} and p corresponds to the Taylor-Hood element and thus guarantees the fulfilment of the inf-sup stability condition [57]. In principle, the entire domain Ω holds the the approximation for the fields \mathbf{d} , \mathbf{u} and p . But for the solid only the degrees of freedom $\mathbf{d}_{n,i}$ are *active* which correspond to points inside Ω^s or, as explained below, in the vicinity of the interface Γ . The same holds for the fluid degrees of freedom $\mathbf{u}_{n,j}$ and $p_{n,k}$ with respect to the domain Ω^f .

It is assumed that the interface $\Gamma_n = \partial\Omega_n^s$ is strictly inside the full domain Ω for all time instants t_n . Therefore, there are no boundary conditions for the solid domain. For the fluid domain, the Dirichlet boundary condition (7)₁ on $\partial\Omega$ is treated essentially. Moreover, in case of $\partial\Omega$ being entirely treated as a Dirichlet boundary, the fluid pressure can only be known up to a constant value and we therefore set $p = 0$ at some point of this boundary in order to ensure solvability of the fluid problem.

Figure 3 shows the configuration of Fig. 1 with an immersed finite element discretisation. A simple structured grid is filling the domain Ω without awareness of the current location of Γ_n . More precisely, the figure shows the respective discretisation of fluid and solid in the left and right pictures. Obviously, the elements which are traversed by the interface Γ_n have a *physical* and a *fictitious* side such that the solution is well-defined until the boundary. These fictitious element parts are highlighted in Fig. 3.

Implicit geometry representation

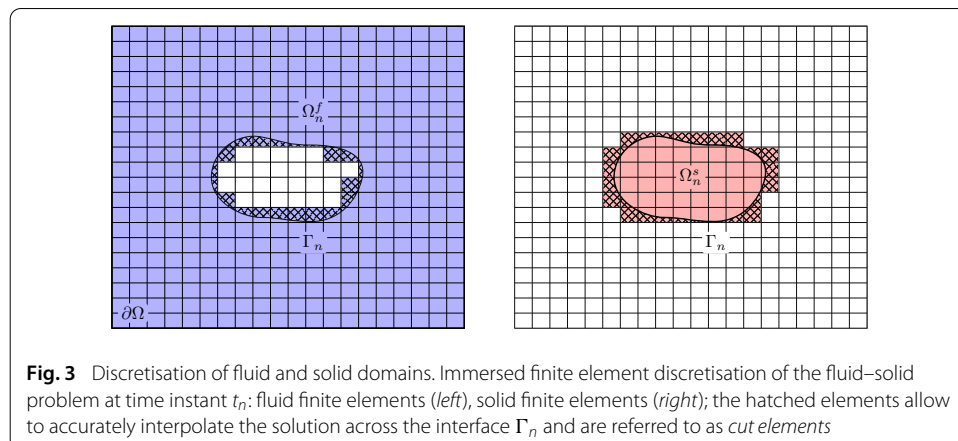
Although the analysis mesh is virtually independent of the configuration of solid and fluid domains, the expressions in the Newton method (26) require integrating over the interface Γ_n and the volumes Ω_n^s and Ω_n^f , respectively. Therefore, it is necessary to know about the location of the interface Γ_n on the element level and we employ a signed distance function

$$\text{dist}_{\Gamma_n}(\mathbf{x}) = s(\mathbf{x}) \min_{y \in \Gamma_n} |\mathbf{x} - \mathbf{y}|, \quad \text{with} \quad s(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \overline{\Omega_n^s} \\ -1 & \text{if } \mathbf{x} \in \overline{\Omega_n^f} \end{cases} \quad (28)$$

This function represents Γ_n implicitly as the level set [48] $\text{dist}_{\Gamma_n}(\mathbf{x}) = 0$. Note that this choice of geometry representation has the effect of smoothing out surface features. Here, this poses no problem since the applications we consider all have a smooth interface from the onset. In case of fluid–solid interfaces with a corners and edges, this choice has to be reviewed carefully, see, for instance, [42]. In the implementation, dist_{Γ_n} is represented by its interpolate using the finite element functions of the background mesh (here, piece-wise linear). The coefficients of this interpolation are the nodal values of the distance function,

$$\text{dist}_{\Gamma_n}(\mathbf{x}) \approx \sum_i r_{n,i} N_i^r(\mathbf{x}) \quad \text{with} \quad r_{n,i} = \text{dist}_{\Gamma_n}(\mathbf{x}_i). \quad (29)$$

Note the dependence of the coefficients $r_{n,i}$ on the time instant t_n due the dynamics of the location of the interface Γ . The calculation of $r_{n,i}$ requires to find the surface element (line or triangle) which is closest to the point \mathbf{x}_i of interest. Given N mesh points \mathbf{x}_i and M surface elements, this task is of complexity $\mathcal{O}(N \times M)$. There is a variety of algorithms with a lower complexity [58], but here we content ourselves with this brute-force approach; after all this part of the computation is not time-critical.

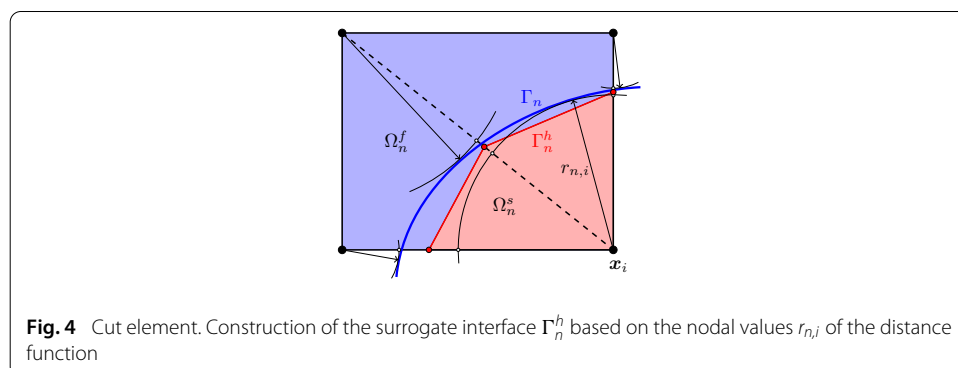


Cut elements

Once the distance function (28) has been computed for all mesh points \mathbf{x}_i , the interface Γ_n is reconstructed in all elements that are traversed by it. That is where a change of the sign of the distance function occurs. To this end, a linear approximation (line elements in two and triangles in three dimensions) is used to construct a polytope Γ_n^h that approximates Γ_n . Such a situation for one element in two dimensions is depicted in Fig. 4. The element is at first decomposed into two triangles (in three dimensions, a hexahedron is decomposed into six tetrahedrons [8]). The arrows in this figure represent the shortest distance from every grid point to the immersed interface Γ_n . By linear interpolation along all edges, intersection points (the red circles) are determined.

After the calculation of the intersection points, piece-wise linear elements form the surrogate interface Γ_n^h in the solution process. Effectively all terms that contain integrals over Γ_n are expressed as integrals over Γ_n^h . In addition, the sub-regions of the cut element which belong to the fluid (Ω_n^f) and the solid (Ω_n^s) side are now polygons or polytopes for which a standard numerical integration is in general not possible. Therefore, these shapes are subdivided into triangles or tetrahedrons on which Gauß quadrature rules [57,59] are used, as it is common in the implementation of XFEM [60]. The integration on the cut elements is therefore carried out with the same accuracy as the volume elements strictly inside the domain. See [61] for an overview of the numerical implementation of integration over cut elements and an alternative approach based on the divergence theorem and surface integration. Alternatively, in [62] an approach for explicit time integration is given in which a surrogate boundary is instead thereby avoiding the cut element integration.

At last, the stability of the finite element basis used in (27) needs to be considered. Consider Fig. 3 and let us assume that all grid points hold degrees of freedom $\mathbf{d}_{n,i}$, $\mathbf{u}_{n,j}$, and $p_{n,k}$. In fact, the quadratic shape functions used for the fluid velocity \mathbf{u} lead to additional degrees of freedom, but they are ignored in this discussion for sake of simplicity. These degrees of freedom can be classified by the intersection of their support with the domain of integration. Let $N_i(\mathbf{x})$ be any shape function of (27) and $S_i = \text{supp}(N_i)$ its support, that is $N_i(\mathbf{x}) = 0$ for all $\mathbf{x} \notin S_i$. Focusing on the discretisation of the solid displacements (the fluid side is treated analogously), the degrees of freedom are classified as follows: *inactive* if $S_i \cap \Omega_n^s = \emptyset$; *critical* if $S_i \cap \Omega_n^s \neq \emptyset$ and $|S_i \cap \Omega_n^s| < \epsilon h$, using some predefined threshold ϵ ; *active* otherwise. Note that this categorisation is dependent on the time instant t_n and therefore the used finite element spaces change between the time instants.



Inactive degrees of freedom do not pose any problem as they are simply discarded from the approximation (27). Similarly, active degrees of freedom do not need any special consideration and are treated as in any standard finite element method. But the critical ones need a special consideration which is due to the fact that the measure $s_{n,i} = |S_i \cap \Omega_n^s|$ relates to the condition of the final system matrix. Obviously, if $s_{n,i} \rightarrow 0$ the corresponding contributions to the system matrix vanish and the resulting linear system is ill-conditioned. There are various approaches which address this stability issue, see [63]. We choose the method proposed by [63], see also [8,36], and precondition the system by constraining the critical degrees of freedom to active ones,

$$\mathbf{d}_{n,i} = \sum_{j \in J_n(i)} c_{ij} \mathbf{d}_{n,j}, \tag{30}$$

where $\mathbf{d}_{n,i}$ denotes a critical degree of freedom. The set $J_n(i)$ contains suitably chosen active degrees of freedom and c_{ij} are the weights of this linear constraint. See [36,63] for the technical details of this approach and [64] for an alternative approach of stabilising the finite element basis on cut cells. Note that introducing (30) into the finite element approximation (27) gives rise to a modified finite element basis, akin to the *extended* B-splines introduced by [63]. Effectively, the support of some shape functions in the vicinity of the immersed interface Γ_n is enlarged as they included linear combinations of shape functions that correspond to critical degrees of freedom. This implies that, even though the intersection of elements with the solid or fluid domain can be arbitrarily small, the shape function supports are bounded from below. In other words, the above introduced measure $s_{n,i}$ of intersection between shape function support and domain of integration is always of a similar magnitude as the mesh size h . Note, alternatively, the ghost penalty approach from [34], where an additional term is added to the Nitsche method in order to give stability for small cut elements.

It remains to discuss the parameter γ of the weak incorporation of boundary conditions as introduced in (15). As shown in [16,51], the global fluid-solid balance (12) can be interpreted as a fluid-only problem or a solid-only problem, using a space decomposition [52]. The latter is simply a problem of linear elasticity with a Robin boundary condition. Here, the parameter γ represents physically the stiffness of the support and $\gamma > 0$ is a sufficient condition for the solvability of the problem [57]. For the fluid problem, on the other hand, the value of γ needs further attention. In order to ensure a stable inverse of the fluid saddle point problem, two conditions need to be satisfied: the inf-sup condition and the ellipticity of the bilinear form which only depends on \mathbf{u} and $\delta \mathbf{u}$. The former condition is already fulfilled by the choice of the finite element discretisation by Taylor-Hood elements. For the ellipticity condition we have to analyse the term

$$\begin{aligned} \tilde{a}^f(\mathbf{u}, \delta \mathbf{u}) = & \int_{\Omega^f(t)} 2\mu^f \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\delta \mathbf{u}) \, d\Omega - \int_{\Gamma(t)} 2\mu^f \boldsymbol{\varepsilon}(\mathbf{u}) \mathbf{n}^f \cdot \delta \mathbf{u} \, d\Gamma \\ & - \int_{\Gamma(t)} 2\mu^f \boldsymbol{\varepsilon}(\delta \mathbf{u}) \mathbf{n}^f \cdot \mathbf{u} \, d\Gamma + \gamma \int_{\Gamma(t)} \delta \mathbf{u} \cdot \mathbf{u} \, d\Gamma. \end{aligned} \tag{31}$$

Note the change of sign that occurs due to the choice of the normal vector $\mathbf{n}^f = -\mathbf{n}$. We require now $\tilde{a}^f(\mathbf{u}, \mathbf{u}) > 0$ for all $\mathbf{u} \neq \mathbf{0}$. By introducing the abbreviation $A(\mathbf{u}, \delta \mathbf{u}) = \int_{\Omega^f} 2\mu^f \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\delta \mathbf{u}) \, d\Omega$, this expression can be estimated as follows

$$\begin{aligned} \tilde{\alpha}^f(\mathbf{u}, \mathbf{u}) &= A(\mathbf{u}, \mathbf{u}) - 2 \int_{\Gamma} 2\mu^f \boldsymbol{\varepsilon}(\mathbf{u}) \mathbf{n}^f \cdot \mathbf{u} \, d\Gamma + \gamma \int_{\Gamma} \mathbf{u} \cdot \mathbf{u} \, d\Gamma \\ &\geq \left(\sqrt{A(\mathbf{u}, \mathbf{u})} - C_I \|\mathbf{u}\|_{\Gamma} \right)^2 + (\gamma - C_I^2) \|\mathbf{u}\|_{\Gamma}^2 > 0, \quad \mathbf{u} \neq \mathbf{0}, \gamma > C_I^2. \end{aligned} \quad (32)$$

This estimate is based on the Cauchy-Schwarz inequality, Korn’s inequality [57] to ensure the positivity of $A(\mathbf{u}, \mathbf{u})$, and a special inverse inequality that reads

$$C_I^2 A(\mathbf{u}, \mathbf{u}) \geq \int_{\Gamma} (2\mu^f \boldsymbol{\varepsilon}(\mathbf{u}) \mathbf{n}^f)^2 \, d\Gamma. \quad (33)$$

Details on the derivation of the estimate (32) and the inverse inequality (33) can be found in, among others, [37,38]. Estimate (32) assures that for the choice $\gamma > C_I^2$ the matrix block corresponding to $A(\mathbf{u}, \delta\mathbf{u})$ has a stable inverse and in combination with the inf-sup condition a stable solution of the linear system is guaranteed. But the exact value of C_I from the inverse estimate (33) is not obvious. Ways to find estimates for this value are discussed in [38]. Especially attractive is the element-wise approach in which the problem is considered locally taking into account that all elements which are strictly in side the domain only have the elliptic $A(\mathbf{u}, \delta\mathbf{u})$ as a contribution, but no interface terms, and need therefore not to be considered. Following the steps of [38], the estimate for C_I becomes under the assumption of piece-wise linear finite element shape functions for \mathbf{u} for an element Ω_e

$$C_{I,e}^2 > 2\mu^f \frac{|\Omega_e \cap \Gamma|}{|\Omega_e \cap \Omega^f|}. \quad (34)$$

Note that there are possible geometric configurations in which this value is unbounded and therefore not suited to guarantee numerical stability. But the derivation of (34) does not consider the stabilisation technique that is employed here. The use of the linear constraints (30) effectively augment the support of shape functions near the boundary and a value of the constant of the order

$$\gamma > C_I^2 = \frac{\gamma_0 \mu^f}{h}, \quad (35)$$

where h is a measure of the element size, is sufficient even for relatively small numbers γ_0 . See [36] for a numerical study of this parameter in the context of fluid flow around moving boundaries. In our numerical results in “Example applications” section, a value of $\gamma_0 = 1$ has been chosen unless noted otherwise and no stability issues are encountered.

Interface update and particle tracking

The Updated Lagrangian Method, as outlined in section “Updated Lagrangian method and linearisation”, is now reconsidered. The aim is to change the solid configuration, but maintain a fixed mesh throughout the simulation. Instead of using a material mesh as in a standard finite element method for large deformations which moves along with the solid, only the solid surface (the interface) representation is relocated. This is achieved by the following steps

- Extract a surface mesh $\tilde{\Gamma}_n$ from the current level set data dist_{Γ_n} (Eq. (28) and the construction in Fig. 4) which delivers an explicit description of the interface
- Evaluate at every node \mathbf{x}_i^{Γ} of this surface mesh the current displacement increment and update the coordinate of that node.

$$\mathbf{x}_i^{\Gamma} \leftarrow \mathbf{x}_i^{\Gamma} + (\mathbf{d}_{n+1} - \mathbf{d}_n)(\mathbf{x}_i^{\Gamma}). \quad (36)$$

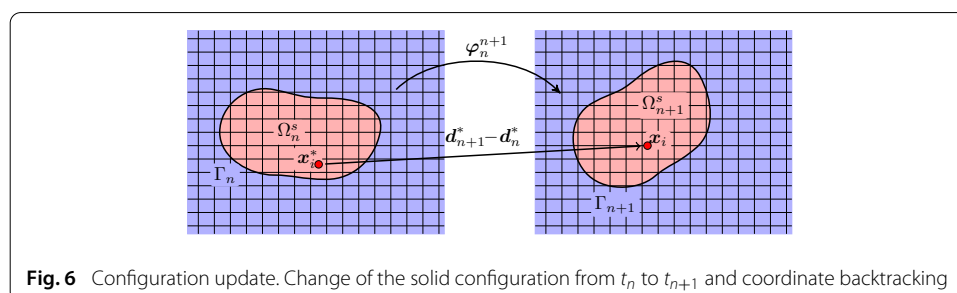
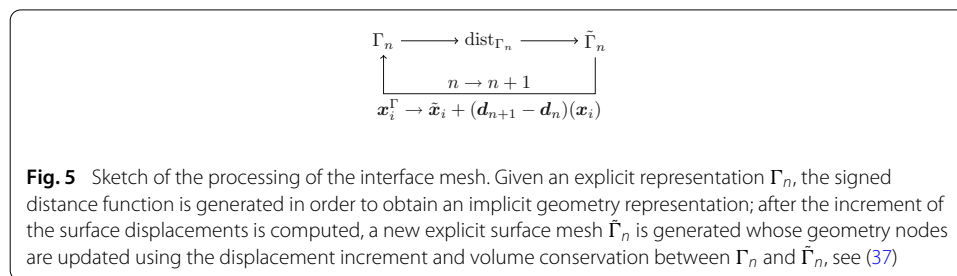
- The thereby updated surface mesh becomes Γ_{n+1} and is used for the the signed distance function (29) in the next step.

These surface operations require some comments. In principle it is possible to work with one surface mesh throughout the simulation. But the surface Γ_n^h as seen by the immersed finite element solver (refer to Fig. 4) does not coincide with the original surface mesh Γ . This implies that a displacement solution might not be available at the location of a node of the original surface mesh. Moreover, the use of a newly generated mesh from the level set data liberates the method from the surface mesh size: the thus generated surface mesh has always the same resolution as the domain mesh. A pure level set approach, on the other hand, works only with an implicit geometry description, but requires an additional advection equation to be solved in every time step [48]. Such equation poses numerical difficulties and would add another field of unknowns to our system. For this reason, we have chosen to combine both approaches. This approach of switching between explicit and implicit surface descriptions is depicted in Fig. 5. But the conversion from explicit to implicit and back to an explicit surface representation (as in the top line of this diagram) affects the volume enclosed by Γ and can lead to undesirable shrinking effects, especially in case of relatively coarse meshes. For this reason, an artificial coordinate \tilde{x}_i is introduced before the surface location is updated which is defined as

$$\tilde{x}_i = c + \alpha(x_i^\Gamma - c) \quad \text{with} \quad \alpha = \sqrt[D]{V_n/\tilde{V}_n} \tag{37}$$

with the enclosed volumes V_n of Γ_n and \tilde{V}_n of $\tilde{\Gamma}_n$, and c the centre of the domain Ω^s ; recall that $D = 2$ or $D = 3$ denotes the spatial dimension. Note that this coordinate scaling only affects the gain or loss of volume throughout the conversions between explicit and implicit surface representations. The volume changes due to the elastic deformation of the compressible material are not inhibited.

In order to maintain the deformation history of the solid, at every grid node in the new configuration its previous location has to be determined. As shown in Fig. 6, the grid point x_i in the new configuration at time t_{n+1} results from its previous location x_i^* and the displacement increment from t_n to t_{n+1} at that location, i.e.



$$\mathbf{x}_i = \mathbf{x}_i^* + (\mathbf{d}_{n+1} - \mathbf{d}_n)(\mathbf{x}_i^*). \tag{38}$$

Note that this expression is the same as in a standard Updated Lagrangian Method [44]. But the main difference is that here the new location \mathbf{x}_i is given and the right hand side is sought for. Using a finite element geometry representation and the displacement trial as in (27), this equation becomes

$$\mathbf{x}(\xi_i^*) + \mathbf{d}_{n+1}(\xi_i^*) - \mathbf{d}_n(\xi_i^*) - \mathbf{x}_i = \mathbf{0}, \tag{39}$$

where ξ represents the local element coordinates and ξ_i^* is the solution to this nonlinear equation. In order to solve equation (39), first the element has to be found in which $\mathbf{x}_i^* = \mathbf{x}(\xi_i^*)$ lies, and then a Newton method is used to obtain the value of ξ_i^* . Searching for elements in a structured grid is a simple task and one can begin with the element that contains $\mathbf{x}_i - (\mathbf{d}_{n+1} - \mathbf{d}_n)(\mathbf{x}_i)$ as an initial guess. If the Newton method does not converge in this element, its neighbours are considered. The Newton method itself consists of the iterations

$$\begin{aligned} \left[\frac{\partial \mathbf{x}}{\partial \xi} + \frac{\partial(\mathbf{d}_{n+1} - \mathbf{d}_n)}{\partial \xi} \right]_{\xi=\xi_i^{(k)}} \Delta \xi &= \mathbf{x}_i - \mathbf{x}(\xi_i^{(k)}) - (\mathbf{d}_{n+1} - \mathbf{d}_n)(\xi_i^{(k)}) \\ \xi_i^{(k+1)} &= \xi_i^{(k)} + \Delta \xi \end{aligned} \tag{40}$$

and converges rapidly to ξ_i^* . Once, the point has been determined, the displacement history is transferred from \mathbf{x}_i^* to \mathbf{x}_i . This kind of particle tracking can be found in [45] and avoids the advection of the solid displacement typical for a fully Eulerian approach [39,40,42,65].

Fluid–solid coupling

In order to finalise this section, the algorithmic steps of the devised method are summarised. Given the latest known configuration at t_n by means of the solid displacements \mathbf{d}_n and the surface mesh Γ_n , the following steps are performed:

Geometry immersion Compute the signed distance function (28) based on the given surface mesh Γ_n .

Fluid–solid balance Solve problem (12) with the interface term (16) by a Newton method as shown in expression (26). After convergence, the new solid displacement \mathbf{d}_{n+1} and the fluid state variables \mathbf{u}_{n+1} and p_{n+1} are known. The spatial discretisation is performed as described in this section.

Geometry update Extract a surface mesh $\tilde{\Gamma}_n$ from the signed distance function. Update the node locations of this mesh based on the displacement increment $(\mathbf{d}_{n+1} - \mathbf{d}_n)$ taking into account the re-scaling as explained above. This yields the new interface location Γ_{n+1} and implies the solid and fluid domain locations.

Backtrack nodal locations For every finite element node in the new solid domain Ω_{n+1}^s , find its previous location in Ω_n^s . Transfer the field variables from the previous to the new location.

With the end of the last step, the new configuration at t_{n+1} is completely determined and the solid displacement history is known at every finite element node in Ω_{n+1}^s .

Example applications

In all examples, the elastic behaviour of the solid is a compressible Neo-Hookean material [53] with strain energy expressed in the deformation gradient \mathbf{F}

$$W(\mathbf{F}) = \frac{\lambda}{2} (\log J)^2 - \mu^s \log J + \frac{\mu^s}{2} (\text{tr } \mathbf{C} - 3), \quad J = \det \mathbf{F} \text{ and } \mathbf{C} = \mathbf{F}^\top \mathbf{F}. \tag{41}$$

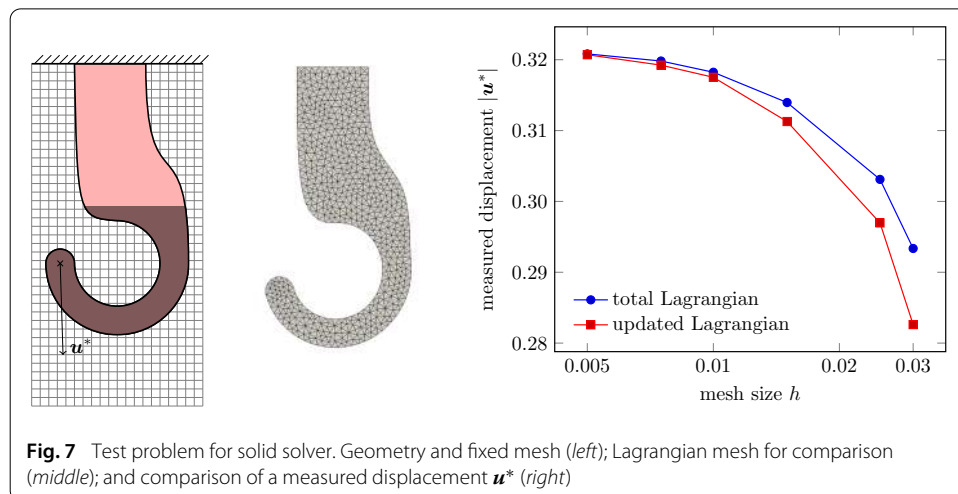
The material parameters λ and μ^s are the Lamé parameters in E and ν , that is $\lambda = \frac{E\nu}{(1-2\nu)(1+\nu)}$ and $\mu^s = \frac{E}{2(1+\nu)}$. The fluid is incompressible Newtonian according to (5) with the dynamic viscosity μ^f .

For the finite element method, we use lowest-order Taylor-Hood elements (Q_2/Q_1) for the fluid (velocity/pressure) and Q_1 elements for the solid displacement [57]. An Euler backward time integration is used with constant step size Δt . The coupling parameter γ is chosen as $\gamma = \alpha\mu^f/h$, where h is the characteristic element size, see Eq (35). If not indicated otherwise, $\alpha = 1$ is the default choice. The convergence criterion for the Newton method (26) is the ℓ^2 norm of the vector representing the displacement increment $\Delta \mathbf{d}$ divided by the number of solid degrees of freedom. The tolerance in all computations is chosen as 10^{-10} and at most three iterations are observed. The same tolerance is used for the Newton method (40) in the coordinate backtracking. Note that in the examples the effective mesh size can be arbitrarily small, while the interface moves through the finite element mesh. But due to the stabilisation, as presented in “Cut elements” section, the size of the intersection of the shape function support with the integration domain does not shrink to zero but remain of the order of h . For this reason, the use of the element size h of the embedding domain grid is a valid mesh characteristic.

Before focusing on the fluid–solid applications, a pure solid example is considered. The aim of this example is to demonstrate the viability and accuracy of the devised Updated Lagrangian method for the solid domain based on the immersed finite element method with particle backtracking.

Updated Lagrangian method

In a first, preliminary example, the accuracy of the updated Lagrangian method, as introduced in section “Updated Lagrangian method and linearisation”, together with the particle backtracking of “Interface update and particle tracking section” is assessed. Therefore, a problem only consisting of a solid domain without fluid interaction is devised. The undeformed geometry of the solid domain is depicted in the left picture of Fig. 7. Other than in the derivation of the presented fluid–solid coupling method and the remaining examples, the boundary of the solid domain Ω^s partially overlaps with the boundary $\partial\Omega$ of the embedding domain. Along this overlap the displacements are set to zero by employing



essential boundary conditions. Moreover, the initial solid domain Ω_0^s has subdomain $\tilde{\Omega}_0^s$ in which a constant downward body force $\mathbf{f}_0 = -\frac{n}{2}\mathbf{e}_2$ is applied with $0 \leq n \leq 4$ as the number of the load step. The region of the applied body force is darker in the picture. The solid is hyperelastic according to (41) with parameters $E = 100$ and $\nu = 0.3$.

In order to assess the quality of the solution of this problem, a pure Lagrangian approach with a body-conforming mesh is used as comparison and the middle picture in Fig. 7 shows one of the meshes. Since the computational domain changes throughout the computation in the Updated Lagrangian approach, the applied body force in the latest known configuration Ω_n^s becomes

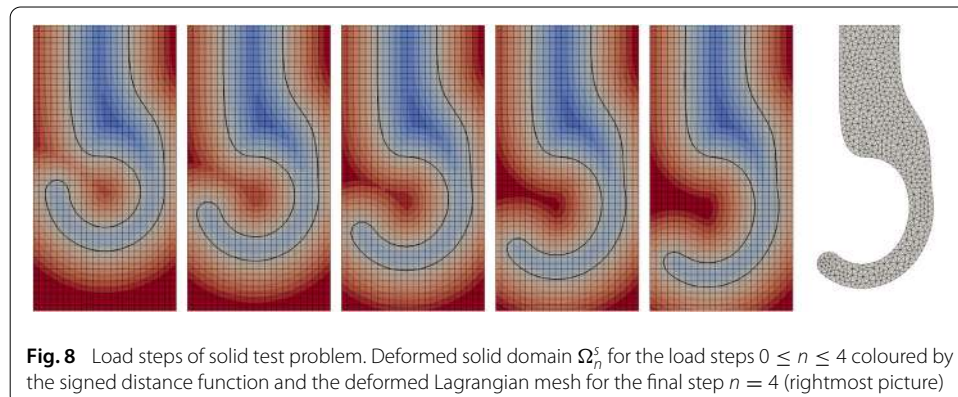
$$\mathbf{f}_n = \det \mathbf{F}_0^n \mathbf{f}_0. \tag{42}$$

Figure 8 shows the deformed solid domain for all load steps and for visual comparison the deformed Lagrangian mesh for the final step. Without noticeable difference between the two approaches, the boundary of the solid domain freely moves throughout the fixed mesh.

For a more quantitative comparison, the right graph in Fig. 7 shows the modulus of the measured displacement \mathbf{u}^* for the two approaches and various mesh sizes. Clearly, both approaches converge to very similar numerical values of the measured displacement. Note that the presented method for the solid part is tailored towards the analysis large deformation problems. In a linearised setting, the distinction between the configurations as shown in Fig. 2 does not make sense and there would be no need for the update of a configuration. For this reason, analytic solutions for the convergence study are highly complicated and we therefore rely on numerical reference solutions. Moreover, the methods we compare in this section operate in different configurations such that we are restricted to compare the results at individual points.

Shear flow–convergence analysis

Whereas the previous section shows the convergence of the results for a problem with only a solid domain, we aim to assess here the convergence behaviour of the method for a fluid–solid interaction problem. To this end, we compare the numerical solution from various grid sizes with the outcome of a highly refined grid. Due to the lack of an analytical solution, we use this fine-grid result as the reference solution in order to quantify the convergence. This concept of assessing the accuracy of the method has been employed by other authors, see, for instance, [33].



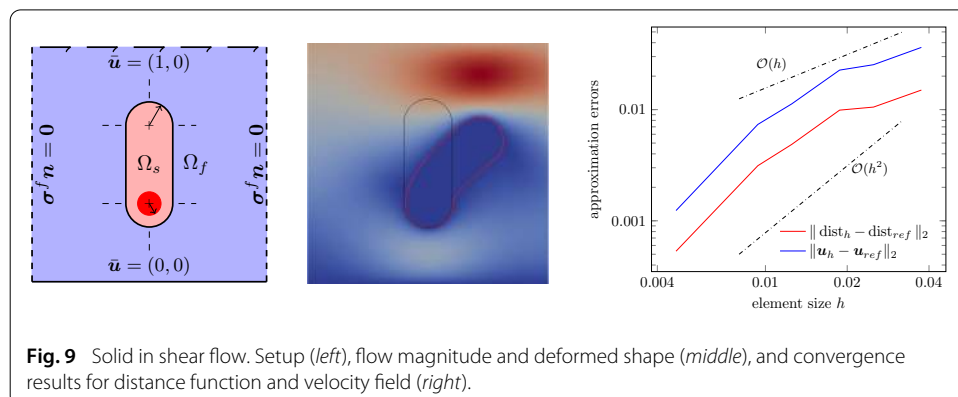
To this end, we consider the problem as described in Fig. 9, a fully immersed hyperelastic solid domain Ω^s of depicted shape whose material points in the dark red circle are held fixed. The radius of the upper and lower circular boundaries of the solid is 0.15, whereas the circle of fixed material points is half as big with a radius of 0.075. The distance between the centres of the circular boundaries is 0.5 and the whole fluid box has the dimensions 1.5×1.5 . The surrounding fluid box is subject to a prescribed shear flow and leads to a bending of the solid body. The fluid is Newtonian with viscosity $\mu^f = 1$ and the parameters for the hyperelastic material as in (41) are $E = 1000$ and $\nu = 0.3$. For the simulation 10 time steps with a step size $\Delta t = 0.2$ are used which yield the deformed shape as shown in the middle picture of Fig. 9 and which corresponds to a quasi-static state.

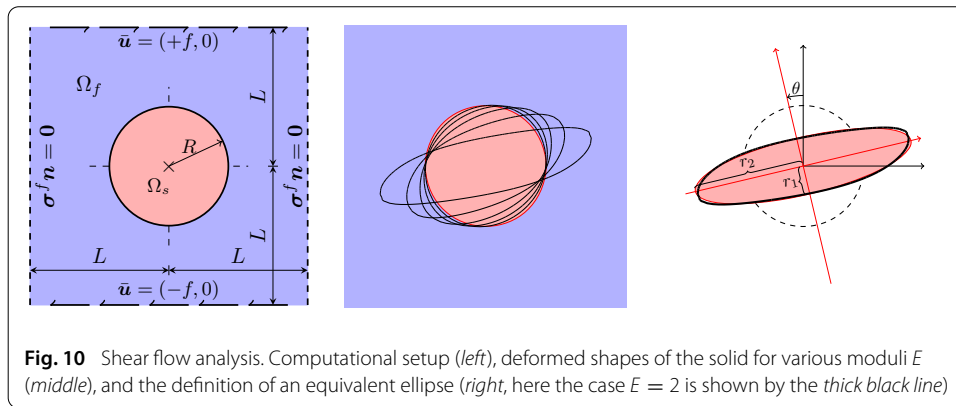
As reference solution a fine grid of 480×480 elements, i.e. $h = 0.003125$, is used. The right graphic in Fig. 9 shows the L_2 -norm errors of the distance function dist_Γ and the velocity field \mathbf{u} for various grid sizes h with respect to the chosen reference at the final step of the simulation at $t = 2$. One can see the both measured errors behave similarly. They have an approximately linear decay for coarse grids followed by a order higher than linear. The final order, which appears to be more than quadratic, is clearly owed to the choice of reference solution and does not claim to be a characteristic of the method. Due to the choice of the time stepping method, see (17), the convergence order is impeded and does not reach the quadratic behaviour as linear finite elements for linear static problems commonly exhibit [57]. Moreover, one has to bear in mind that the solution of nonlinear systems in every time step by the Newton method (26) and the particle backtracking (40) contribute to the overall error of the method.

Shear flow–parameter study

Consider the setup as depicted in the left picture of Fig. 10. An initially circular solid object with radius R is placed at the centre of a fluid box of size $2L \times 2L$. The fluid has the viscosity μ^f and the solid the material parameters E and ν . At the top and bottom boundaries a horizontal velocity is prescribed in positive and negative directions, respectively. The entire computational domain $\Omega = \Omega^f \cup \Omega^s$ is discretised by $N \times N$ elements leading to a constant mesh size of $h = 2L/N$. The initial situation is the undeformed circle as shown in the picture and at $t > 0$ the velocity boundary condition $\bar{\mathbf{u}}$ is applied. All simulations cover the time interval $0 \leq t \leq 20$.

In the following, most of the problem parameters related to discretisation, material behaviour, geometry and boundary conditions are varied and their respective influence





on the solution is studied. The choice of parameters is given in Table 1, where the standard value and the possible variations of each parameter are shown. To get an image of various outcomes the right picture in Fig. 10 shows the shapes of the immersed interface Γ at $t = 20$ for various solid stiffness parameters E while keeping all other parameters at their standard value.

As it can be seen from the deformed shapes in Fig. 10, the circle converts into an ellipse-like shape with the deformation obviously greater for the lower material stiffness E . In the following analysis of the influence of the parameters of Table 1, the shape of the solid domain is used. Therefore, the moments of inertia of the solid domain,

$$I_{ij} = \int_{\Omega^s} x_i x_j \, d\Omega, \tag{43}$$

are computed. In this two-dimensional analysis, the I_{ij} form a symmetric 2×2 -matrix whose eigenvalues I_1 and I_2 are the principal moments of inertia. We assume $I_1 \leq I_2$ in the following. Since the observed shapes are almost like a tilted ellipse, these principal moments of inertia allow to determine the minor and major radius, r_1 and r_2 , of an equivalent ellipse. With these radii, the eccentricity e and a shape parameter D_{12} [28] are defined

$$e = \sqrt{1 - \frac{r_1^2}{r_2^2}} = \sqrt{1 - \frac{I_1}{I_2}} \quad \text{and} \quad D_{12} = \frac{r_2 - r_1}{r_2 + r_1} = \frac{\sqrt{I_2} - \sqrt{I_1}}{\sqrt{I_2} + \sqrt{I_1}}. \tag{44}$$

Moreover, the angle θ by which the ellipse deviates from a horizontal orientation, is also studied.

At first the method parameters N , Δt and α are considered. It turned out that the multiplier α of the boundary term as in the paragraph leading to Equation (35) did not

Table 1 Parameters used in two-dimensional shear flow problem

Parameter	Standard value	Variations
N	50	25, 75, 100, 200
Δt	0.1	0.05, 0.2, 0.5
α	1	0.01, 0.1, 10, 100
E	50	2, 5, 10, 20
ν	0.3	—
μ^f	1	0.1, 0.5, 2, 4
f	1	0.2, 0.5, 2, 5, 10
R	0.6	0.2, 0.4, 0.8, 1.0
L	1.4	1.0, 1.8, 2.4, 3.0

show any noticeable influence on the monitored quantities, and is thus omitted from the rest of the discussion. The left picture of Fig. 11 shows the eccentricity e plotted over the time of analysis $0 \leq t \leq 20$ for the variation of N , the number of elements per direction. One can clearly see that there is a fluctuation of the values throughout time and the amplitude of this fluctuation diminishes with increasing values of N . This becomes more clear, when looking at the time-averaged values of e and θ in the right picture. Here \bar{a} denotes the average of the quantity a over the time interval $5 \leq 20$. The values of the average angle and the average eccentricity tend towards a specific value with increasing N , see Fig. 11. Similar observations are made for the time step variations.

Now, the influence of the solid material's stiffness parameter E on the outcome of the solution is considered. The deformed shapes at $t = 20$ are already shown in Fig. 10. A decreasing value of E leads to a more pronounced flattening of the solid object and a lower angle of inclination θ . These observations are confirmed by the plots in Fig. 12, where the evolution of e and θ are shown for all considered values of E .

The deformation of red blood cells in simple shear flow has been studied in [28] for various membrane stiffness. This study is based on the common model of these cells as a liquid-filled membrane. Nevertheless, there are strong similarities between the findings in [28] and the results shown in Fig. 12 in this work. In both cases, the shape deformation from initially spherical to elliptical and the angle variation take mainly place for $t <$

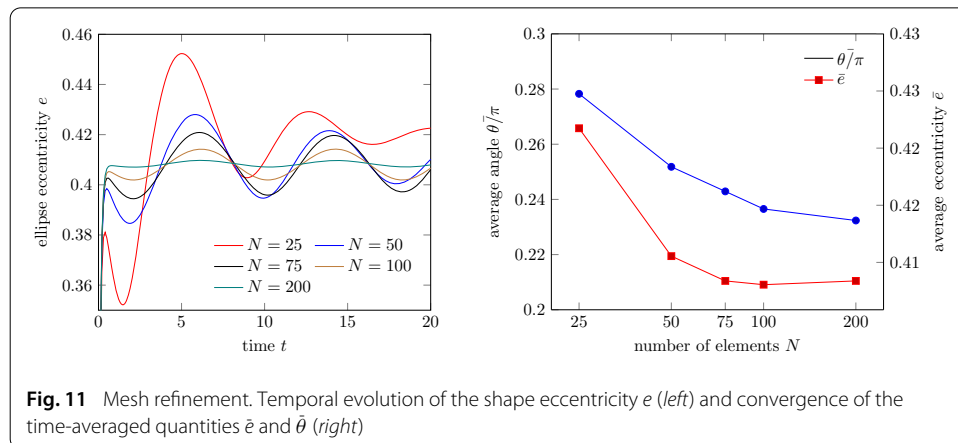


Fig. 11 Mesh refinement. Temporal evolution of the shape eccentricity e (left) and convergence of the time-averaged quantities \bar{e} and $\bar{\theta}$ (right)

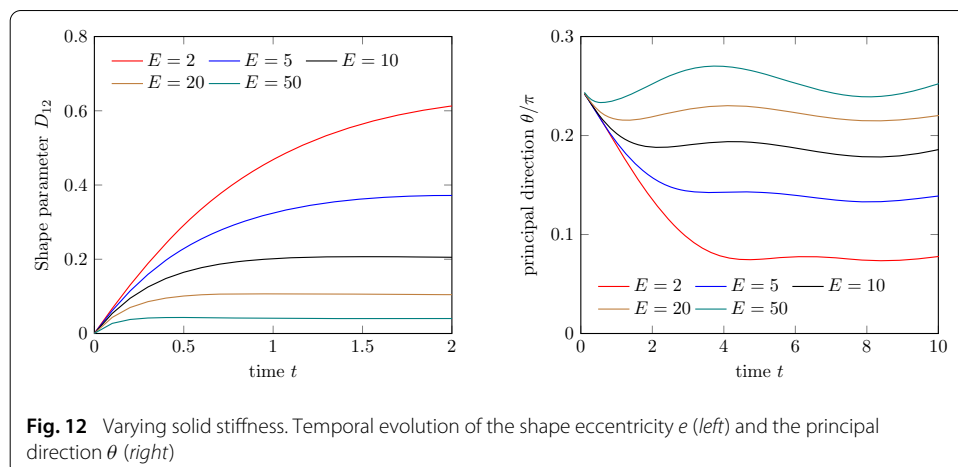
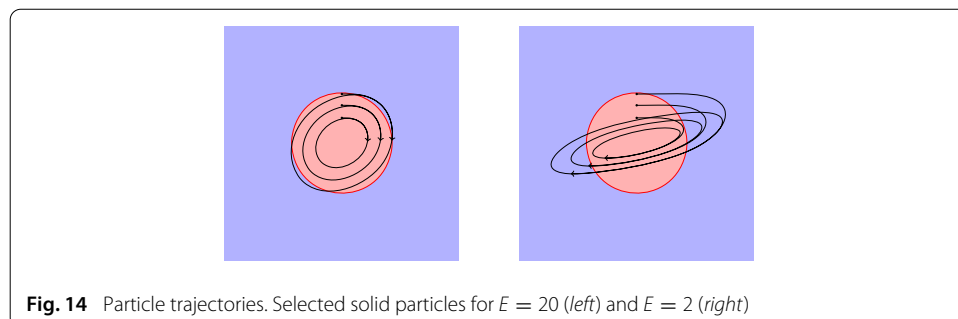
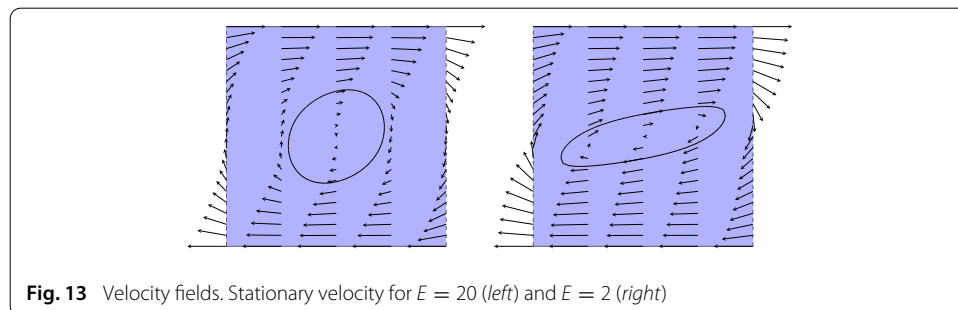


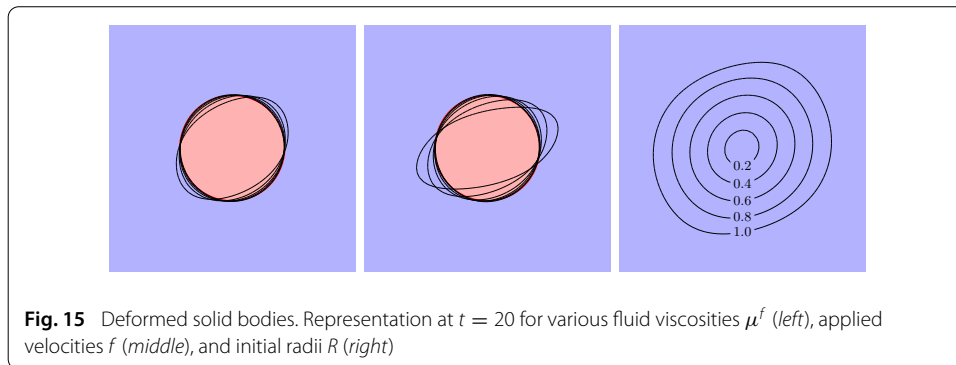
Fig. 12 Varying solid stiffness. Temporal evolution of the shape eccentricity e (left) and the principal direction θ (right)

3. Moreover, a decrease of the material stiffness (membrane stiffness in [28] and solid stiffness E here) leads to an increase in shape eccentricity and a deviation of the principal direction from $\pi/4$. Despite the different constitutive models (solid vs. membrane), the qualitative outcome is very similar.

Even though the fluctuations of e diminish for larger times, this state of equilibrium is dynamic. This means that the immersed body experiences a non-zero velocity along its surface Γ and thus is constantly moving in circumferential direction. This type of motion is referred to as tank-treading and typical for vesicles [66]. Snapshots of the velocity fields in fluid and solid at $t = 20$ are shown in Fig. 13 for two different choices of E . Clearly, the solid body is subject to a rotating flow field and responds with internal rotation. For a lower material stiffness, this rotational motion is increased. It is important to note that this is not a rigid body rotation, because the solid domain's outer shape, which is non-circular, remains fixed while the material moves. The tank-treading is also confirmed by marking individual material particles and tracing their path throughout the simulation. Fig. 14 shows the trajectories for three material points and the same choices of E as in the previous figure. These particles move to their almost elliptical orbits during the initial deformation and then stay on this path throughout the considered time.

At last, the deformed shapes at $t = 20$ are displayed for variations of the fluid viscosity μ^f , the applied velocity f and the radius of the circular solid domain at $t = 0$. For the chosen values of μ^f , see Table 1, one observes that a higher viscosity increases the deformation of the solid body. Clearly, the viscous forces of the fluid flow are increased and have a stronger effect on the solid, see the left picture in Fig. 15. Obviously the same happens for an increase of the applied velocity, as shown in the middle picture. The right picture of Fig. 15 shows the deformed shapes for various sizes of the solid body. Whereas the smallest solid with $R = 0.2$ poses almost no obstruction to the fluid flow, there is an increasing deformation (i.e. deviation from the initial circle) visible for larger values of



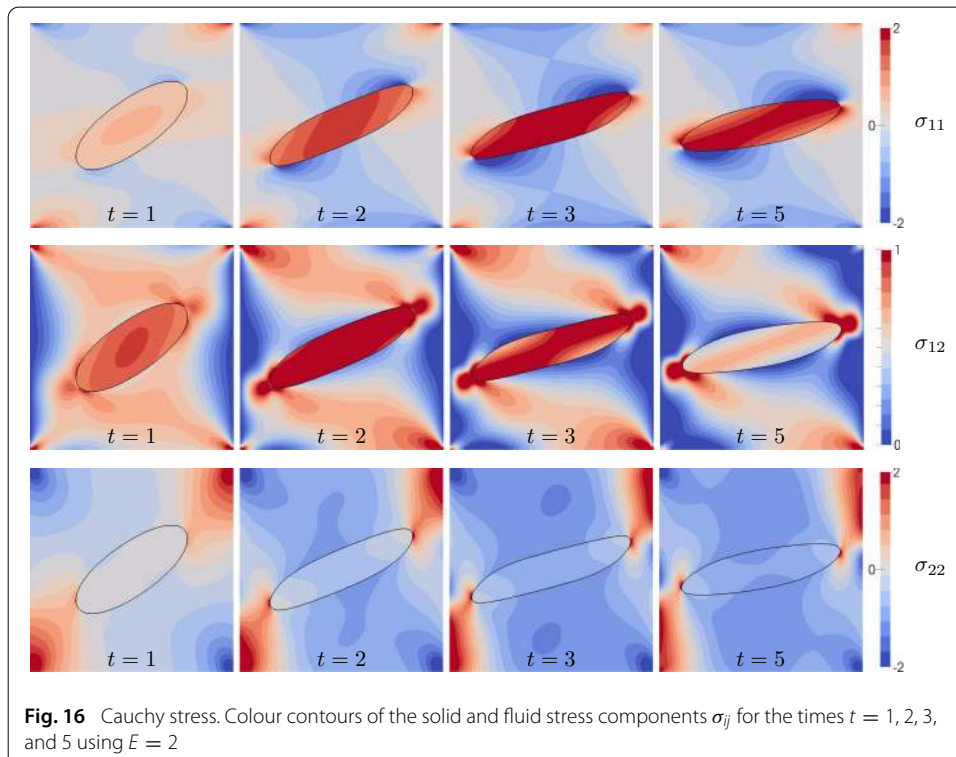


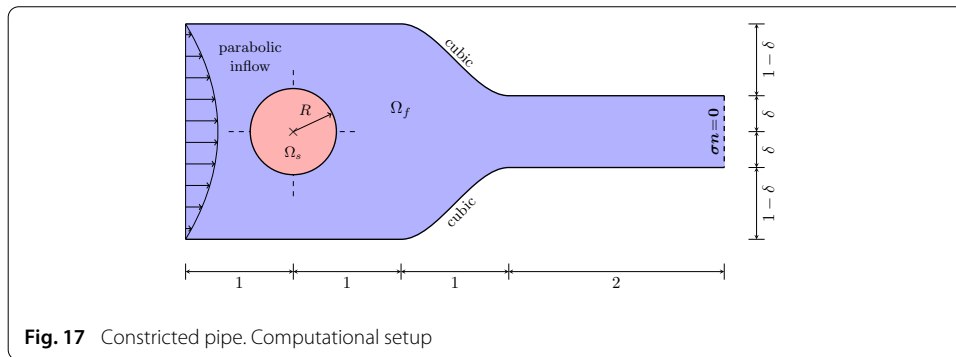
R . The variation of the size of the fluid box, namely the half-width parameter L , did not reveal any significant alterations in the solution.

Finally, we consider for the softest solid with $E = 2$ the variation of the stress components σ_{ij} in the fluid and the solid domains at some time instants. Figure 16 displays contour plots of the stress components for four different times. It has to be emphasised the plotted stress components refer to the Cartesian coordinate axis and not the principal axes of the deformed solid. Note that there are strong indications that stresses regulate substantial biological processes in living cells [5]. The proposed method allows for future applications in which a detailed stress analysis is required for a deeper insight in such processes.

Flow in a narrowed tube

A circular solid object is placed in a pipe with a geometric constriction and subject to a forced flow. The geometric dimensions are shown in Fig. 17 and the boundary conditions





are a parabolic inflow from the left with average velocity $\bar{\mathbf{u}} = (0.1, 0)$, an open boundary at the right end, and no-slip walls at the top and bottom boundaries. The constriction is defined by a cubic polynomial reduces the pipes diameter from 2 down to 2δ with different values of δ as given in Table 2. The mesh used in all simulations is a structured 250×100 -grid deformed to fit into the constricted pipe. Therefore, the elements have a constant length in x_1 -direction of $h_1 = 0.02$. They are of square shape in the left (the wide) part of the pipe and squeezed correspondingly compressed in x_2 -direction in the right (the constricted) part. Hence, the mesh size in the vertical direction shrinks from $h_2 = 0.02$ down to a value between 0.005 ($\delta = 0.25$) and 0.008 ($\delta = 0.4$). The size of the time steps is constant with $\Delta t = 0.025$. The fluid viscosity is set to $\mu^f = 1$, the solid's Poisson ratio $\nu = 0.3$ and the stiffness modulus assumes the values from Table 2. The radius of the solid circle is always $R = 0.4$.

Since $\delta \geq R$ for all chosen values of δ , the solid cannot pass undeformed through the narrow pipe section. Figure 18 shows snapshots of the deformed solid at 12 different time instants between $t = 0$ and $t = 13.75$ with a constant time difference 1.25. In this picture, the most narrow case of $\delta = 0.25$ is shown. One can observe that the compressible solid initially shrinks, which is due to the hydrostatic pressure of the fluid environment, and moves with the fluid motion towards the constriction. The closer the solid gets towards narrow section, the more it deforms to a tampon-like shape, just wide enough to fit through the pipe.

Table 2 Parameter variations in constricted pipe flow

Parameter	Value	Variation
δ	0.33	0.25, 0.3, 0.4
E	100	50, 200, 500

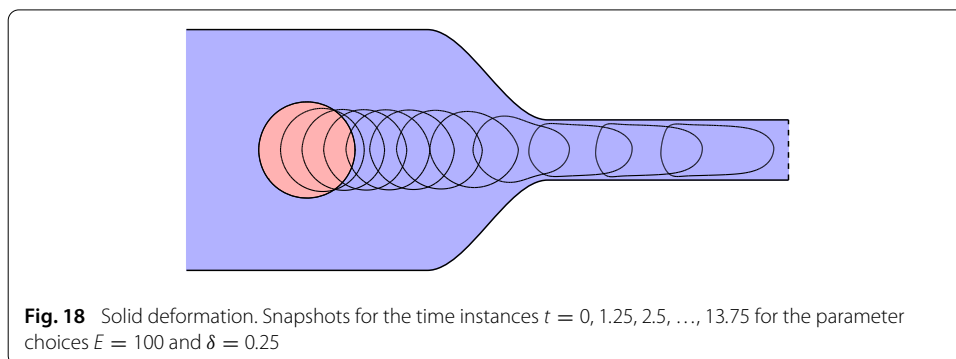


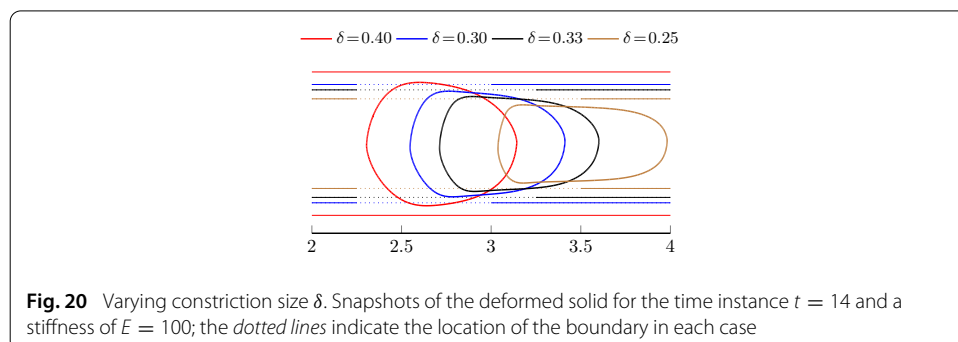
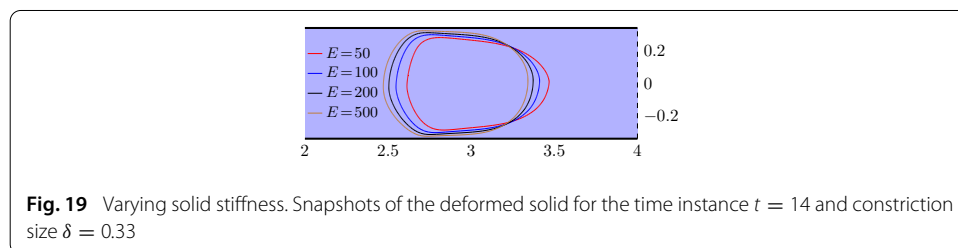
Fig. 18 Solid deformation. Snapshots for the time instances $t = 0, 1.25, 2.5, \dots, 13.75$ for the parameter choices $E = 100$ and $\delta = 0.25$

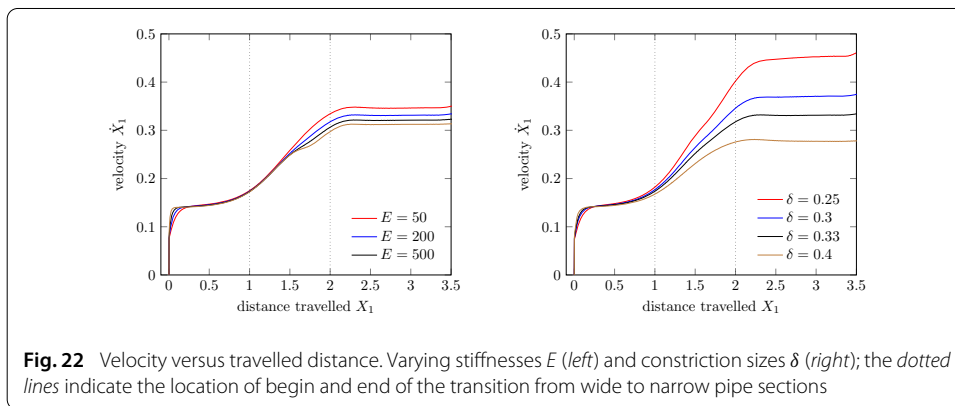
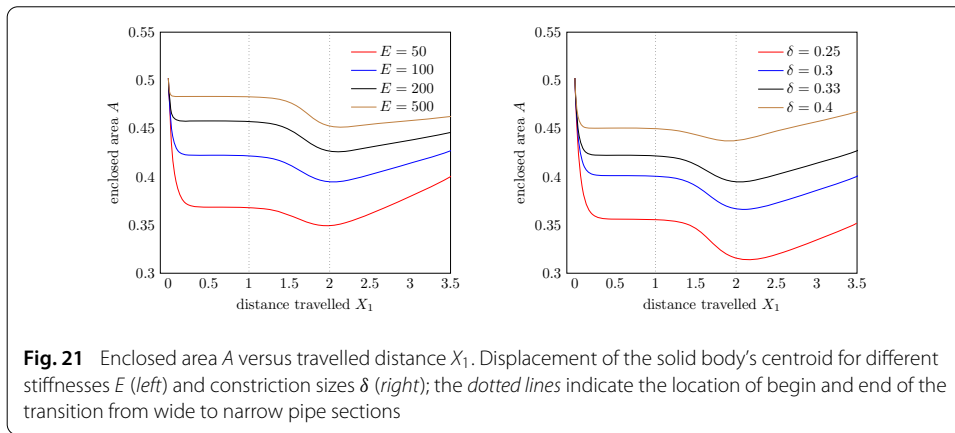
Figures 19, 20 show the deformed solid at the time instant $t = 14$. At this moment, the solid is located completely inside the narrow part of the pipe. Figure 19 shows the different deformed shapes for the various stiffness parameters E . Clearly, the softest material leads to narrower state of deformation and is slightly more advanced since it poses less disturbance to the fluid flow. In Fig. 20 the deformed shapes for a constant stiffness but different constriction sizes are shown. Obviously, a narrower pipe requires a larger deformation of the solid. Moreover, the fluid velocity is higher in the narrower section due to the mass balance of the incompressible fluid. Therefore, the solid has advanced more in the narrower pipe.

Above observations are quantified in Figs. 21, 22, where the enclosed area A of the solid body and the velocity are plotted versus the current position and for all parameter variations. The initial shrinkage due to the fluid pressure is clear visible in Fig. 21 and more pronounced for lower values of E (softer material) and lower values of δ (higher fluid pressure). The area then stays constant while the solid travels towards the narrower section. When entering the transition region it begins to shrink more until reaching a minimum size when approximately entering the narrow section. The travelled distance is measured by the location of the centroid of the solid body with respect to its initial location. When inside the narrow section, the solid gets stretched and increases in area.

The velocity of the solid body's centroid is shown in Fig. 22 for all parameter choices. At the begin of the simulation the solid catches the velocity of the surrounding fluid and this increases when approaching the transition to the narrower section. Once entered this final part, the velocity stays approximately constant.

In [2], a three-dimensional experiment of the flow of liquid-filled capsules through a narrowed pipe have been carried out. Although the presented example is two-dimensional and a compressible solid rather than a capsule is subject to the fluid flow, the observed shapes [2] are similar to Figs. 19, 20. In the experiments, it has been found that with increasing capillary number Ca the rear end of the capsule becomes flatter and eventually, above some value of Ca buckles inwards. Here the values of Ca are $1/E$ and do not fall into the range of inward buckling. Yet a increasing flattening with lower value of E (i.e.,

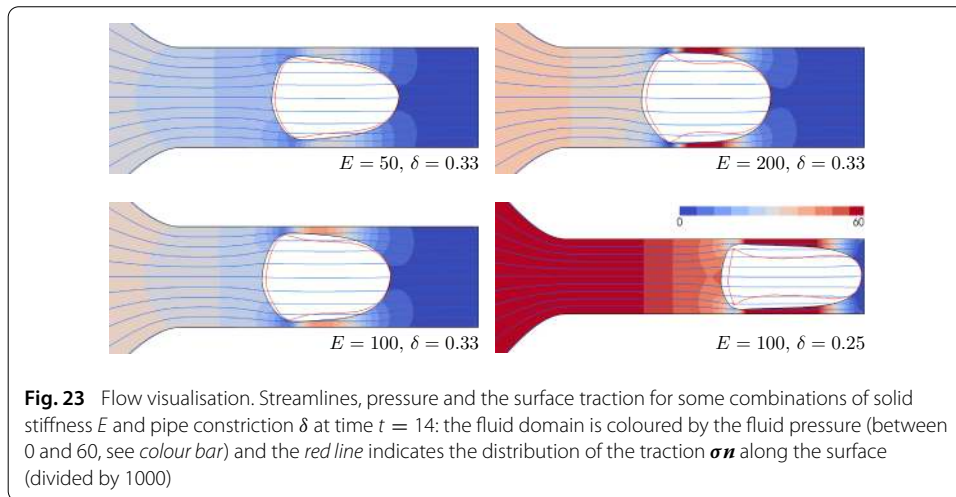




higher value of Ca) is visible in Fig. 19. Note that our model does not allow for contact between the immersed solid and the domain boundaries. As the solid squeezes through the narrowed section of the pipe it never touches the boundaries but there remains a fluid gap between the interface and the boundary, see Fig. 18. This gap is an outcome of the simulation and not subject to artificially imposed distance constraints. Via the stabilisation techniques of “Cut elements” section, this implies a restriction on the mesh size. If too many neighbouring fluid elements have degenerate degrees of freedom (a geometric situation similar to a cusp), the employed stabilisation technique fails.

Finally, in Fig. 23 for a few parameter combinations the velocity streamlines are shown together with the fluid pressure and the surface traction σn . For the computation of the streamlines, the discrete velocity according to (17) has been used inside the solid domain. Clearly, the flow pattern do not differ significantly among the displayed images. But the fluid pressure is higher in case of a larger solid stiffness E as it is necessary in order to sufficiently deform the solid body. In case of a smaller pipe diameter the fluid pressures are obviously larger. Accordingly, the distribution of the surface traction becomes higher for larger values of E and smaller values of δ .

This example is concluded by a comment on the advantage of immersed finite elements. The most common technique for the fully-coupled analysis of fluid–solid interaction is the *Arbitrary Lagrangian-Eulerian* (ALE) technique [26] in which the fluid mesh is deformed in order to accommodate for the solid deformation and to maintain a usable analysis mesh. Although a powerful method, it is expected that the here presented example is not

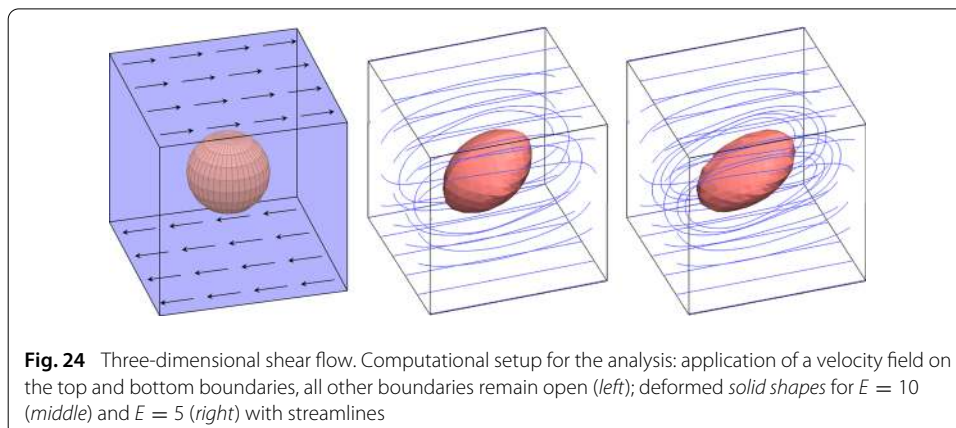


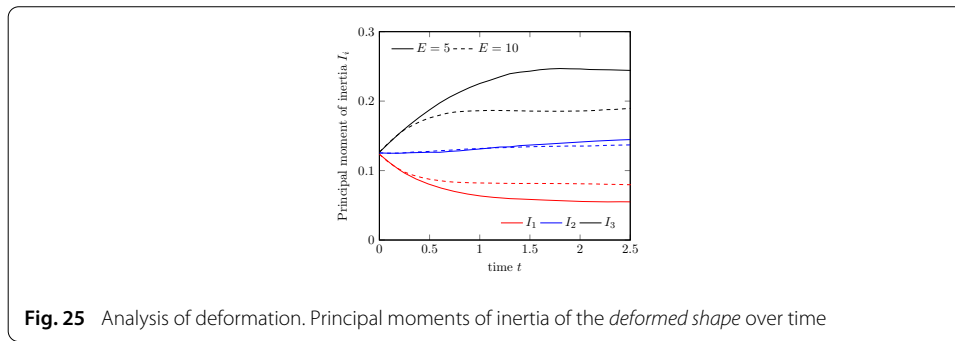
directly accessible by an ALE method. Figs. 18, 19, 20 clearly show that the initial fluid mesh would be highly distorted when the solid body enters the narrow section of the pipe. Tedious re-meshing and solution mapping techniques are required in an ALE approach.

Three-dimensional shear flow

At last, a three-dimensional example is considered. Here the setup is similar to the shear flow of section “Shear flow – parameter study” but extended to the third dimension. Fig. 10 shows the initial configuration: a sphere of radius $R = 0.6$ located at the centre of a fluid box of dimension $[-1.4, 1.4]^3$. All parameters are chosen as the standard values of Table 1 apart from the material stiffness parameter, which assumes the values $E = 5$ or $E = 10$, and the spatial discretisation is carried out by 20^3 elements of size $h = 0.14$.

Figure 24 displays the initial configuration and the final deformed shapes for the two chosen material parameters. As in the two-dimensional case, the solid assumes an elliptic shape that is inclined in the plane of the shear flow. In Fig. 25, the principal moments of inertia, computed from the 3×3 -matrix with coefficients (43), are shown for a time interval $0 \leq t \leq 2.5$ and the two choices of E . The spheres flatten as discussed above and rotate about the axis perpendicular to the plane of shear by approximate angles 0.20π for $E = 10$ and 0.17π for $E = 5$. As seen in Fig. 25, the moment of inertia around this





out-of-plane axis stays almost constant throughout the simulation and is similar for both stiffnesses. The other two moments clearly deviate from their initial values as the elliptic shape in the plane of shear is formed.

Conclusion

A new approach for the numerical analysis of the interaction between viscous fluid flow and highly deformable solids has been presented. The method builds up on previous works, such as [8,36] for the analysis of fluid flow around moving and highly flexible boundaries. Derived from the basic balance equations for the quasi-static equilibrium of solid and fluid, the interface conditions are incorporated weakly and a global solid-fluid balance law is obtained. The formalism of an updated Lagrangian method is used for the description of the solid constituent. Its equilibrium is thus expressed in the latest known configuration and the deformation history maintained by particle tracking between the newly computed and the previous state of the solid. By means of this choice, the advection equations and the shape derivatives in the linearisation of a fully Eulerian method are avoided. The analysis is carried out on one mesh which stays fixed in space and time.

For the spatial discretisation an immersed finite element method is employed with the mesh independent of the location of the solid–fluid configuration. By using a signed distance function, the location of the interface is given implicitly to the finite element solver. The difficulty of an accurate quadrature of the elements which are crossed by the interface is as much addressed as the possible ill-conditioning of the system of equations due to the small support of shape functions on such elements. Once the nonlinear iterations of the fully coupled fluid–solid system are converged and the new equilibrium has been found, the configuration is updated. To this end, an explicit representation of the surface is recovered from the level set and this is updated by means of the displacement increments along the surface. Finally, every mesh node of the solid domain is tracked back to its previous location in order to transfer the displacement history.

Due to the choice of a monolithic fluid–solid coupling, the method is unconditionally stable. Moreover, the full linearisation in the Updated Lagrangian framework leads to a fast convergence within every time step. Several example applications in two and three spatial dimensions are presented and the influence of all parameters of the method are studied. Being tailored towards the analysis of cell motility and microfluidic experimentation, we consider shear flow examples which reveal the basic characteristics of liquid-filled vesicles, such as tumbling and tank-treading behaviour. Moreover, the passing of a deformable object through a narrowed tube of diameter smaller than the body is analysed and the

trajectories of an elastic solid in the vortex of a driven cavity flow. Especially, the former application provides an important step in the direction of computational analysis of cell migration in confined spaces [67].

In view of these examples, we highlight the following features of the devised method. Based on simple balance laws, virtually any material law of solid and fluid constituents can be incorporated. Especially, active behaviour, like growth or self-propulsion, is a feasible extension. Moreover, the restriction to a quasi-static equilibrium is not essential and an adaptation of the method to fully dynamic solid–fluid interaction is straightforward. The developed immersed finite element method operates with a single analysis mesh for solid and fluid that is not subject to any deformation. This is particularly useful for the analysis of the narrowed tube and the driven cavity examples in which the use of a body-fitted mesh is not possible in any standard way. Finally, we aim to emphasise that the here presented approach is virtually mesh-free: even though a volume finite element mesh is employed it is not subject to any geometrical restrictions and its generation is a trivial task.

Authors' contributions

TR derived the mathematical model for the immersed FE method and carried out the numerical implementation. JMGA defined the conception of the underlying physical models. Both authors were fully involved in the preparation of this manuscript and the interpretation of the results. All authors read and approved the final manuscript.

Author details

¹Multiscale in Mechanical and Biological Engineering (M2BE), University of Zaragoza, María de Luna 3, 50018 Zaragoza, Spain, ²Fundación ARAID, María de Luna 11, 50018 Zaragoza, Spain.

Acknowledgements

The support of the European Research Council (ERC), through project ERC-2012-StG 306751, and of the Spanish Ministry of Economy and Competitiveness, through project DPI2012-38090-C03-01 (partly financed by the European Union through the European Regional Development Fund), is gratefully acknowledged.

Competing interests

The authors declare that they have no competing interests.

Received: 17 December 2015 Accepted: 12 February 2016

Published online: 15 March 2016

References

1. Tarbell JM, Weinbaum S, Kamm RD. Cellular fluid mechanics and mechanotransduction. *Ann Biomed Eng.* 2005;33:1719–23.
2. Risso F, Collé-Paillot F, Zagzoule M. Experimental investigation of a bioartificial capsule flowing in a narrow tube. *J Fluid Mech.* 2006;547:149–73.
3. Tezduyar TE, Sathe S, Cragin T, Nanna B, Conklin BS, Pausewang J, Schwaab M. Modelling of fluid-structure interactions with the space-time finite elements: Arterial fluid mechanics. *Int J Numer Methods Fluids.* 2007;54:901–22.
4. Keller SR, Skalak R. Motion of a tank-treading ellipsoidal particle in a shear flow. *J Fluid Mech.* 1982;120:27–47.
5. Diamond S, Eskin S, McIntire L. Fluid flow stimulates tissue plasminogen activator secretion by cultured human endothelial cells. *Science.* 1989;243:1483–5.
6. Polacheck WJ, Li R, Uzel SG, Kamm RD. Microfluidic platforms for mechanobiology. *Lab Chip.* 2013;13:2252–67.
7. van der Meulen MC, Huiskes R. Why mechanobiology?: A survey article. *J Biomech.* 2002;35:401–14.
8. Rüberg T, Cirak F. A fixed-grid b-spline finite element technique for fluid-structure interaction. *Int J Numer Methods Fluids.* 2014;74:623–60.
9. Tian FB, Dai H, Luo H, Doyle JF, Rousseau B. Fluid-structure interaction involving large deformations: 3d simulations and applications to biological systems. *J Computational Phys.* 2014;258:451–69.
10. Lim C, Zhou E, Quek S. Mechanical models for living cells—a review. *J Biomech.* 2006;39:195–216.
11. Ohayon R, Felippa C. Advances in computational methods for fluid-structure interaction. *Comput Methods Appl Mech Eng.* 2001;190:2977–3292.
12. Bazilevs Y, Takizawa K, Tezduyar TE. Special issue on computational fluid mechanics and fluid-structure interaction. *Comput Mech.* 2011;48:245–348.
13. Tezduyar TE, Bazilevs Y. Advances in computational fluid mechanics and fluid-structure interactions: A tribute to Yoichiro Matsumoto on the occasion of his 60th birthday. *Int J Numer Method Fluids.* 2011;65:1–340.
14. Bazilevs Y, Takizawa K, Tezduyar TE. *Computational fluid-structure interaction: methods and applications.* Hoboken: Wiley; 2012.
15. Heil M. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Comput Method Appl Mech Eng.* 2004;193:1–23.

16. Burman E, Fernández M. Stabilization of explicit coupling in fluid-structure interaction involving fluid incompressibility. *Comput Method Appl Mech Eng.* 2009;198:766–84.
17. Felippa CA, Park K, Farhat C. Partitioned analysis of coupled mechanical systems. *Comput Method Appl Mech Eng.* 2001;190(24):3247–70.
18. Küttler U, Wall WA. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Comput Mech.* 2008;43:61–72.
19. Mittal R, Iaccarino G. Immersed boundary methods. *Ann Rev Fluid Mech.* 2005;37:239–61.
20. Tezduyar TE. Finite element methods for flow problems with moving boundaries and interfaces. *Arch Comput Method Eng.* 2001;8:83–130.
21. Takizawa K, Tezduyar TE. Multiscale space-time fluid-structure interaction techniques. *Comput Mech.* 2011;48:247–67.
22. Küttler U, Gee M, Förster C, Comerford A, Wall W. Coupling strategies for biomedical fluid-structure interaction problems. *Int J Numer Method Biomed Eng.* 2010;26:305–21.
23. Takizawa K, Wright S, Moorman C, Tezduyar TE. Fluid-structure interaction modeling of parachute clusters. *Int J Numer Method Fluids.* 2011;65:286–307.
24. Kramer R, Cirak F, Pantano C. Fluid-structure interaction simulation of an inflatable aerodynamic tension-cone decelerator. *AIAA J.* 2010; 4608.
25. Bazilevs Y, Hsu MC, Scott M. Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Comput Method Appl Mech Eng.* 2012;249:28–41.
26. Hirt C, Amsden AA, Cook J. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J Comput Phys.* 1974;14:227–53.
27. Peskin C. The immersed boundary method. *Acta Numer.* 2002;11:479–517.
28. Eggleton CD, Popel AS. Large deformation of red blood cell ghosts in a simple shear flow. *Phys Fluids.* 1998;10:1834–45.
29. Zhang L, Gerstenberger A, Wang X, Liu WK. Immersed finite element method. *Comput Method Appl Mech Eng.* 2004;193:2051–67.
30. Pozrikidis C. Interfacial dynamics for stokes flow. *J Comput Phys.* 2001;169:250–301.
31. Veerapaneni SK, Gueyffier D, Zorin D, Biro G. A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2d. *J Comput Phys.* 2009;228(7):2334–53.
32. Veerapaneni SK, Rahimian A, Biro G, Zorin D. A fast algorithm for simulating vesicle flows in three dimensions. *J Comput Phys.* 2011;230(14):5610–34.
33. Valkov B, Rycroft CH, Kamrin K. Eulerian method for fluid-structure interaction and submerged solid-solid contact problems. 2014.
34. Burman E, Fernández MA, et al. An unfitted nitsche method for incompressible fluid-structure interaction using overlapping meshes. *Comput Method Appl Mech Eng.* 2014;279:497–514.
35. Boffi D, Gastaldi L. A finite element approach for the immersed boundary method. *Comput Struct.* 2003;81(8):491–501.
36. Rüberg T, Cirak F. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Comput Method Appl Mech Eng.* 2011;209–212:266–83.
37. Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsche’s method, for elliptic interface problem. *Comput Method Appl Mech Eng.* 2002;191:5537–52.
38. Dolbow J, Harari I. An efficient finite element method for embedded interface problems. *Int J Numer Method Eng.* 2009;78:229–52.
39. Laadhari A, Ruiz-Baier R, Quarteroni A. Fully eulerian finite element approximation of a fluid-structure interaction problem in cardiac cells. *Int J Numer Method Eng.* 2013;96:712–38.
40. Richter T, Wick T. Finite elements for fluid-structure interaction in ale and fully eulerian coordinates. *Comput Method Appl Mech Eng.* 2010;199:2633–42.
41. He P, Qiao R. A full-eulerian solid level set method for simulation of fluid-structure interactions. *Microfluid Nanofluid.* 2011;11(5):557–67.
42. Dunne T. An eulerian approach to fluid-structure interaction and goal-oriented mesh adaptation. *Int J Numer Method Fluid.* 2006;51(9–10):1017–39.
43. Wick T. Fully eulerian fluid-structure interaction for time-dependent problems. *Comput Method Appl Mech Eng.* 2013;255:14–26.
44. Bathe KJ, Ramm E, Wilson EL. Finite element formulations for large deformation dynamic analysis. *Int J Numer Method Eng.* 1975;9:353–86.
45. Armero F, Love E. An arbitrary lagrangian-eulerian finite element method for finite strain plasticity. *Int J Numer Method Eng.* 2003;57:471–508.
46. Fernández MÁ, Moubachir M. A Newton method using exact jacobians for solving fluid-structure coupling. *Comput Struct.* 2005;83:127–42.
47. Nitsche J. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, vol. 36, Springer, 1971; 9–15.
48. Sethian JA. Theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numer.* 1996;5:309–95.
49. Batchelor GK. *An introduction to fluid dynamics.* Cambridge: University Press; 2000.
50. Ogden RW. *Non-linear elastic deformations.* New York: Courier Dover Publications; 1997.
51. Fernández M, Gerbeau JF, Grandmont C. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *Int J Numer Method Eng.* 2007;69:794–821.
52. LeTallec P, Mouro J. Fluid structure interaction with large structural displacements. *Comput Method Appl Mech Eng.* 2001;190:3039–67.
53. Bonet J. *Nonlinear continuum mechanics for finite element analysis.* Cambridge: University Press; 1997.
54. Stenberg R. On some techniques of approximating boundary conditions in the finite element method. *J Comput Appl Math.* 1995;63:139–48.

55. Annavarapu C, Hautefeuille M, Dolbow JE. A robust Nitsche's formulation for interface problems. *Comput Method Appl Mech Eng.* 2012;225:44–54.
56. Bangerth W, Rannacher R. Finite element approximation of the acoustic wave equation: Error control and mesh adaptation. *EastWest J Numer Math.* 1999;7:263–82.
57. Ern A, Guermond JL. *Theory and practice of finite elements.* New York: Springer; 2004.
58. Mauch S. Efficient algorithms for solving static hamilton-jacobi equations. PhD Thesis, California Institute of Technology; 2003.
59. Hughes TJ. *The finite element method: linear static and dynamic finite element analysis.* New York: Courier Dover Publications; 2012.
60. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *Int J Numer Method Eng.* 1999;46:131–50.
61. Massing A, Larson MG, Logg A. Efficient implementation of finite element methods on nonmatching and overlapping meshes in three dimensions. *SIAM J Sci Comput.* 2013;35:C23–47.
62. Zeng X, Farhat C. A systematic approach for constructing higher-order immersed boundary and ghost fluid methods for fluid-structure interaction problems. *J Comput Phys.* 2012;231(7):2892–923.
63. Höllig K. *Finite element methods with B-splines.* SIAM Front Appl Math. 2003.
64. Schott B, Wall W. A new face-oriented stabilized XFEM approach for 2D and 3D incompressible Navier-Stokes equations. *Comput Method Appl Mech Eng.* 2014;276:233–65.
65. Richter T. A fully eulerian formulation for fluid-structure-interaction problems. *J Comput Phys.* 2013;233:227–40.
66. Barthès-Biesel D. Motion of a spherical microcapsule freely suspended in a linear shear flow. *J Fluid Mech.* 1980;100:831–53.
67. Wolf K, te Lindert M, Krause M, Alexander S, te Riet J, Willis AL, Hoffman RM, Figdor CG, Weiss SJ, Friedl P. Physical limits of cell migration: control by ecm space and nuclear deformation and tuning by proteolysis and traction force. *J Cell Biol.* 2013;201:1069–84.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
