

RESEARCH ARTICLE

Open Access



# An unstructured immersed finite element method for nonlinear solid mechanics

Thomas Rüberg<sup>1,2\*</sup>, Fehmi Cirak<sup>3</sup> and José Manuel García Aznar<sup>1</sup>

\*Correspondence:

thomas.rueberg@gmx.de

<sup>1</sup>Multiscale in Mechanical and Biological Engineering (M2BE), University of Zaragoza, María de Luna 3, 50018 Zaragoza, Spain  
Full list of author information is available at the end of the article

## Abstract

We present an immersed finite element technique for boundary-value and interface problems from nonlinear solid mechanics. Its key features are the implicit representation of domain boundaries and interfaces, the use of Nitsche's method for the incorporation of boundary conditions, accurate numerical integration based on marching tetrahedrons and cut-element stabilisation by means of extrapolation. For discretisation structured and unstructured background meshes with Lagrange basis functions are considered. We show numerically and analytically that the introduced cut-element stabilisation technique provides an effective bound on the size of the Nitsche parameters and, in turn, leads to well-conditioned system matrices. In addition, we introduce a novel approach for representing and analysing geometries with sharp features (edges and corners) using an implicit geometry representation. This allows the computation of typical engineering parts composed of solid primitives without the need of boundary-fitted meshes.

**Keywords:** Immersed finite elements, Nonlinear solid mechanics, Nitsche's method, CSG modelling, Cut-element stabilisation, Implicit geometry

## Background

Conventional finite element methods (FEM) are an irreplaceable tool for the numerical analysis of a variety of physical and engineering problems. They rely on a conforming mesh which approximately matches the domain boundary and material interfaces. For this reason, mesh generation is an essential part of the workflow in FEM-based analyses [1]. Although the procedure is well-established, often the use of a boundary-conforming mesh can be limiting or even prohibitive. Fluid-structure interaction, large elastic deformations and shape optimisation are some applications where mesh entanglement can cause severe difficulties for conventional FEM.

In the last two decades or so, a number of finite element-based numerical methods have been introduced in order to eliminate the need for boundary-conforming meshes. Here, we restrict ourselves to immersed methods, also known as embedded or fictitious domain methods, that operate with a geometry-independent mesh, in the line of [2–6]. Since the mesh of an immersed domain method does not conform with the boundary of the physical domain, one of these methods' main difficulties is the application of boundary conditions. Here, we choose Nitsche's method [7] for the weak enforcement of Dirichlet boundary conditions because it gives optimal convergence rates without incurring major

implementation difficulties. Moreover, the use Lagrange multipliers together with its numerical intricacies, such as the fulfilment of the LBB-condition [8], are avoided. For alternative approaches, see [4, 9–14] among others.

A major difficulty of non-body-fitted methods is the accurate integration of the arising volume and surface integrals. Here, we make use of a tessellation concept which allows to incorporate standard, Gauß quadrature schemes. In the course of this development, a technique is presented which enables the representation of sharp domain features by performing constructive solid geometry (CSG) modelling directly on the embedding mesh. This approach poses a clear advantage in comparison to the conventional methods of geometry resolution because these sharp features are accurately reproduced and not chamfered even on coarse meshes.

Another pitfall of immersed finite element methods is the loss of numerical stability in cases where the intersection of a shape function support with the physical domain becomes very small. This issue has been successfully addressed in the context of b-spline finite elements [6, 12, 15]. In this work, we build up on this concept of constraining critical degrees of freedom and apply it to Lagrangian basis functions on unstructured meshes. Note that Burman et al. [16] introduced an alternative approach, the so-called *ghost-penalty stabilisation method*, which is based on an augmented bilinear form. Strongly related to stability are the method's parameters and we show how to choose these parameters in the context of the introduced stabilisation techniques.

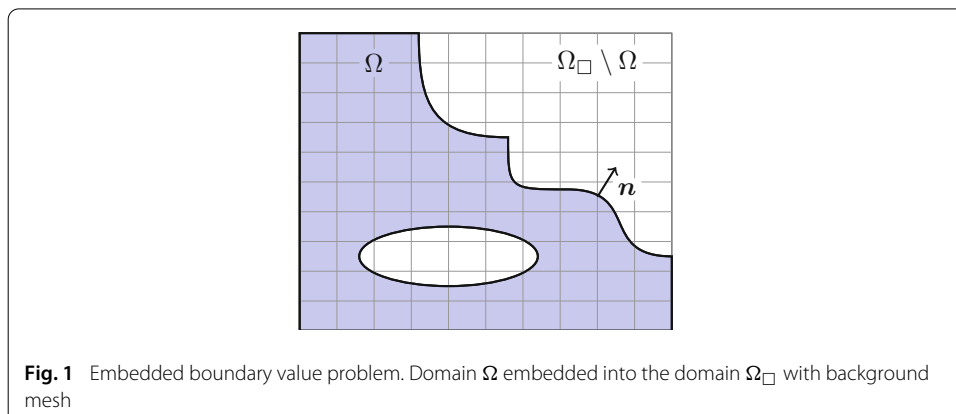
The method we present here is based on our previous works [6, 17–19] and related to [2, 3, 16, 20, 21]. Although, as shown in the cited works, the method can be transferred to many physical applications, we focus on the problem class of nonlinear elasticity.

### Weak enforcement of boundary and interface conditions

At first, we present the derivation of the proposed immersed finite element method as applied to boundary value problems from nonlinear solid mechanics. Using a weighted residual technique, we obtain the weak form of the problem and give its linearisation. Similarly, the expressions for material interface problems are subsequently derived.

### Boundary value problems of nonlinear solid mechanics

Consider the boundary value problem for nonlinear elasticity in the reference domain  $\Omega \in \mathbb{R}^{n_d}$ ,  $n_d = 2$  or  $n_d = 3$  (Fig. 1)



**Fig. 1** Embedded boundary value problem. Domain  $\Omega$  embedded into the domain  $\Omega_{\square}$  with background mesh

$$\begin{aligned}
 -\operatorname{Div} \mathbf{P}(\mathbf{u}) &= \mathbf{f} \quad \text{in } \Omega \\
 \mathbf{u} &= \bar{\mathbf{u}} \quad \text{on } \Gamma_D \\
 \mathbf{t}(\mathbf{u}) &= \bar{\mathbf{t}} \quad \text{on } \Gamma_N,
 \end{aligned} \tag{1}$$

where  $\mathbf{u}$  denotes the unknown displacement field,  $\mathbf{P}$  is the first Piola-Kirchhoff stress tensor,  $\operatorname{Div}$  is the divergence with respect to reference domain coordinates, and  $\mathbf{t}(\mathbf{u}) = \mathbf{P}(\mathbf{u})\mathbf{n}$ , with  $\mathbf{n}$  the outward unit normal vector to  $\Omega$ , is the boundary traction. The prescribed data are the volume force  $\mathbf{f}$ , the prescribed displacement  $\bar{\mathbf{u}}$  and the prescribed traction  $\bar{\mathbf{t}}$ . The boundary of  $\Omega$ , denoted by  $\Gamma$ , is composed of disjoint sets, the Dirichlet boundary  $\Gamma_D$  and the Neumann boundary  $\Gamma_N$ , where the respective data are given.

In order to construct a weak form of the boundary value problem (1), a weighted residual approach is taken with the test function  $\mathbf{v}$ . In mathematical terms, we operate with the Sobolev space  $\mathbf{H}^1(\Omega)$ , i.e. the vector fields whose components are all in  $H^1(\Omega)$ , see, among others, [8] for the precise definition. Different from conventional FEM, we do not employ a constrained subspace with essential boundary conditions. The weighted residual method thus becomes:

$$\begin{aligned}
 \text{Find } \mathbf{u} &\in \mathbf{H}^1(\Omega) \\
 R(\mathbf{u}, \mathbf{v}) &= a(\mathbf{u}, \mathbf{v}) - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega - \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma - \int_{\Gamma_D} \mathbf{t}(\mathbf{u}) \cdot \mathbf{v} d\Gamma = 0 \\
 \forall \mathbf{v} &\in \mathbf{H}^1(\Omega),
 \end{aligned} \tag{2}$$

with

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{S}(\mathbf{u}) : \dot{\mathbf{E}}(\mathbf{v}) d\Omega. \tag{3}$$

Here,  $\dot{\mathbf{E}}$  denotes the variation of the Euler-Green strain tensor ( $\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I})$  with the deformation gradient  $\mathbf{F}$ ) and  $\mathbf{S} = \mathbf{F}^{-1} \mathbf{P}$  the second Piola-Kirchhoff stress tensor [22, 23]. In the applications section, we work with a compressible Neo-Hooke material with given energy density  $W(\mathbf{E})$  and for this hyperelastic case, the stress tensor becomes

$$\mathbf{S} = \frac{\partial W}{\partial \mathbf{E}}. \tag{4}$$

Using a Newton method to solve the nonlinear Eq. (2), the  $k$  th iteration takes the form

$$DR(\mathbf{u}^{(k)}, \mathbf{v})[\Delta \mathbf{u}] = -R(\mathbf{u}^{(k)}, \mathbf{v}) \quad \text{and} \quad \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \Delta \mathbf{u}, \tag{5}$$

where  $D(\cdot)[\Delta \mathbf{u}]$  denotes the derivative in direction of the increment  $\Delta \mathbf{u}$ , which reads

$$DR(\mathbf{u}, \mathbf{v})[\Delta \mathbf{u}] = Da(\mathbf{u}, \mathbf{v})[\Delta \mathbf{u}] - \int_{\Gamma_D} D\mathbf{t}(\mathbf{u})[\Delta \mathbf{u}] \cdot \mathbf{v} d\Gamma \tag{6}$$

with

$$Da(\mathbf{u}, \mathbf{v})[\Delta \mathbf{u}] = \int_{\Omega} (\operatorname{Grad} \mathbf{v}) : \hat{\mathbf{C}}(\mathbf{u}) : (\operatorname{Grad} \Delta \mathbf{u}) d\Omega. \tag{7}$$

In this expression,  $\hat{\mathbf{C}}$  denotes the *effective* elasticity tensor [22]. For simplicity, it is assumed here that the prescribed volume and surface forces,  $\mathbf{f}$  and  $\bar{\mathbf{t}}$ , are independent of the displacement  $\mathbf{u}$  (dead load case). If these assumptions do not hold, the directional derivative of  $R(\mathbf{u}, \mathbf{v})$  contains the derivatives of the applied force and traction terms. So far, expressions (2) and, consequently, (5) do not take into account the displacement boundary condition  $\mathbf{u} = \bar{\mathbf{u}}$  on  $\Gamma_D$ . Therefore, the approach initially introduced by Nitsche [7] is adapted here and the following two terms are added to (5)

$$- \int_{\Gamma_D} D\mathbf{t}(\mathbf{u})[\mathbf{v}] \cdot (\Delta \mathbf{u} - \bar{\mathbf{u}}) d\Gamma \quad \text{and} \quad \gamma \int_{\Gamma_D} (\Delta \mathbf{u} - \bar{\mathbf{u}}) \cdot \mathbf{v} d\Gamma \tag{8}$$

with the predictor  $\tilde{\mathbf{u}} = \hat{\mathbf{u}}$  in the first iteration (or the appropriate value in the first iteration of every load step) and  $\tilde{\mathbf{u}} = \mathbf{0}$  otherwise, corresponding to a displacement-controlled Newton method. The scalar  $\gamma > 0$ , necessary for numerical stability, is discussed in “Numerical stability” section. In summary, the Newton step (5) including Nitsche’s approach to incorporate displacement boundary conditions reads

$$\begin{aligned}
 Da(\mathbf{u}, \mathbf{v})[\Delta \mathbf{u}] &- \int_{\Gamma_D} D\mathbf{t}(\mathbf{u})[\Delta \mathbf{u}] \cdot \mathbf{v} d\Gamma - \int_{\Gamma_D} D\mathbf{t}(\mathbf{u})[\mathbf{v}] \cdot \Delta \mathbf{u} d\Gamma \\
 &+ \gamma \int_{\Gamma_D} \Delta \mathbf{u} \cdot \mathbf{v} d\Gamma = -a(\mathbf{u}, \mathbf{v}) + \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma \\
 &+ \int_{\Gamma_D} \mathbf{t}(\mathbf{u}) \cdot \mathbf{v} d\Gamma - \int_{\Gamma_D} D\mathbf{t}(\mathbf{u})[\mathbf{v}] \cdot \tilde{\mathbf{u}} d\Gamma + \gamma \int_{\Gamma_D} \tilde{\mathbf{u}} \cdot \mathbf{v} d\Gamma,
 \end{aligned} \tag{9}$$

where the iteration counter has been omitted for sake of legibility. The added terms (8) are zero for the exact solution and therefore the method is consistent by construction. Moreover, for hyperelastic materials expression (9) is symmetric and positive for the right choice of  $\gamma$  and non-softening material behaviour, see “Numerical stability” section. In the following, we abbreviate (9) by

$$A(\mathbf{u}; \Delta \mathbf{u}, \mathbf{v}) = \ell(\mathbf{u}; \mathbf{v}). \tag{10}$$

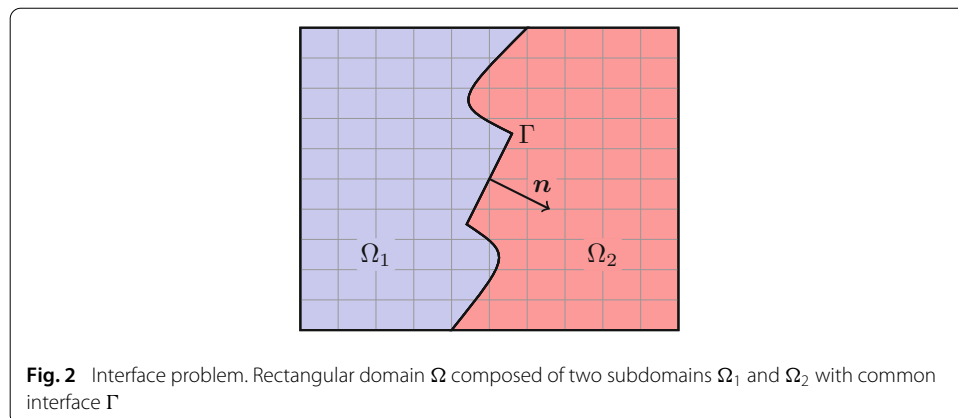
**Material interfaces**

The formalism presented above for the weak incorporation of displacement boundary conditions can be generalised to interface problems, see also [2,3,13]. For simplicity, let the reference domain be composed of two subdomains,  $\Omega = \Omega_1 \cup \Omega_2$ , and let us ignore the Dirichlet boundary conditions on  $\partial\Omega$ . The treatment of such conditions is here essentially the same as in “Boundary value problems of nonlinear solid mechanics” section. We focus only on the conditions imposed on the material interface  $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ , see Fig. 2. In each subdomain  $\Omega_i$  the local equilibrium reads

$$-\text{Div } \mathbf{P}_i(\mathbf{u}_i) = \mathbf{f}_i \quad \text{in } \Omega_i. \tag{11}$$

Let  $\mathbf{u}$  denote the compound displacement field, such that  $\mathbf{u}|_{\Omega_i} = \mathbf{u}_i$ , and define the compound test function  $\mathbf{v}$  similarly. Moreover, for any compound function  $\mathbf{g}$ , with  $\mathbf{g}|_{\Omega_i} = \mathbf{g}_i$ , the jump across  $\Gamma$  is denoted with

$$\llbracket \mathbf{g} \rrbracket = \mathbf{g}_1 - \mathbf{g}_2. \tag{12}$$



**Fig. 2** Interface problem. Rectangular domain  $\Omega$  composed of two subdomains  $\Omega_1$  and  $\Omega_2$  with common interface  $\Gamma$

For later use, also a weighted average  $\{\mathbf{g}\}$  is defined on  $\Gamma$  as

$$\{\mathbf{g}\} = \beta \mathbf{g}_1 + (1 - \beta) \mathbf{g}_2, \tag{13}$$

where  $0 \leq \beta \leq 1$  is some weighting parameter yet to be discussed. On the interface  $\Gamma$  the conditions are

$$\llbracket \mathbf{u} \rrbracket = \mathbf{u}^\Gamma \quad \text{and} \quad \llbracket \mathbf{t}(\mathbf{u}) \rrbracket = \mathbf{t}^\Gamma, \tag{14}$$

with prescribed jump functions  $\mathbf{u}^\Gamma$  and  $\mathbf{t}^\Gamma$ . These conditions represent the jump in the solid displacements and the traction equilibrium across the interface. In more complex situations, such as soft interfaces, a cohesive law can be imposed relating the interface traction  $\mathbf{t}^\Gamma$  to the displacement gap  $\llbracket \mathbf{u} \rrbracket$ , see [24]. Here, we assume that  $\mathbf{u}^\Gamma$  and  $\mathbf{t}^\Gamma$  are prescribed and that they are independent of the displacement  $\mathbf{u}$ . Note that in the evaluation of the tractions  $\mathbf{t}_i$  the unique normal vector  $\mathbf{n} = \mathbf{n}_1 = -\mathbf{n}_2$  as shown in Fig. 2 is used, where this choice is arbitrary.

Repetition of the steps as in the single-domain problem above yields the weighted residual method for interface problems

$$\begin{aligned} \text{Find } \mathbf{u}_i \in H^1(\Omega_i) \quad i = 1, 2, \text{ such that} \\ R(\mathbf{u}, \mathbf{v}) = \sum_{i=1,2} \left( a_i(\mathbf{u}_i, \mathbf{v}_i) - \int_{\Omega_i} \mathbf{f}_i \cdot \mathbf{v}_i d\Omega \right) - \int_{\Gamma} \llbracket \mathbf{t}(\mathbf{u}) \cdot \mathbf{v} \rrbracket d\Gamma = 0 \\ \forall \mathbf{v}_i \in H^1(\Omega_i). \end{aligned} \tag{15}$$

Now the integrand of the interface term is rewritten as follows

$$\begin{aligned} \llbracket \mathbf{t}(\mathbf{u}) \cdot \mathbf{v} \rrbracket &= [\beta \mathbf{t}_1(\mathbf{u}_1) + (1 - \beta) \mathbf{t}_2(\mathbf{u}_2)] \cdot \llbracket \mathbf{v} \rrbracket + [(1 - \beta) \mathbf{v}_1 + \beta \mathbf{v}_2] \cdot \llbracket \mathbf{t}(\mathbf{u}) \rrbracket \\ &= \{\mathbf{t}(\mathbf{u})\} \llbracket \mathbf{v} \rrbracket + [(1 - \beta) \mathbf{v}_1 + \beta \mathbf{v}_2] \cdot \mathbf{t}^\Gamma \end{aligned} \tag{16}$$

employing the average term (13) and the interface conditions (14)<sub>2</sub>. Using a Newton method to solve the nonlinear problem (15) with (16) requires the directional derivative

$$DR(\mathbf{u}, \mathbf{v})[\Delta \mathbf{u}] = \sum_{i=1,2} Da_i(\mathbf{u}_i, \mathbf{v}_i)[\Delta \mathbf{u}_i] - \int_{\Gamma} \{\mathbf{Dt}(\mathbf{u})[\Delta \mathbf{u}]\} \cdot \llbracket \mathbf{v} \rrbracket d\Gamma. \tag{17}$$

The interface condition (14)<sub>1</sub> is now incorporated by adding terms akin to (8), namely

$$- \int_{\Gamma} \{\mathbf{Dt}(\mathbf{u})[\mathbf{v}]\} \cdot (\llbracket \Delta \mathbf{u} \rrbracket - \tilde{\mathbf{u}}^\Gamma) d\Gamma \quad \text{and} \quad \gamma \int_{\Gamma} (\llbracket \Delta \mathbf{u} \rrbracket - \tilde{\mathbf{u}}^\Gamma) \cdot \llbracket \mathbf{v} \rrbracket d\Gamma, \tag{18}$$

to expression (17). Again, the parameter  $\gamma > 0$  is yet to be discussed in the Appendix A and we use the predictor  $\tilde{\mathbf{u}}^\Gamma = \mathbf{u}^\Gamma$  in the first iteration and zero afterwards. In summary, a step in a Newton iteration to solve the coupled interface problem reads

$$\begin{aligned} \sum_{i=1,2} Da_i(\mathbf{u}_i, \mathbf{v}_i)[\Delta \mathbf{u}_i] - \int_{\Gamma} \{\mathbf{Dt}(\mathbf{u})[\Delta \mathbf{u}]\} \cdot \llbracket \mathbf{v} \rrbracket d\Gamma - \int_{\Gamma} \{\mathbf{Dt}(\mathbf{u})[\mathbf{v}]\} \cdot \llbracket \Delta \mathbf{u} \rrbracket d\Gamma \\ + \gamma \int_{\Gamma} \llbracket \Delta \mathbf{u} \rrbracket \cdot \llbracket \mathbf{v} \rrbracket d\Gamma = \sum_{i=1,2} \left( \int_{\Omega_i} \mathbf{f}_i \cdot \mathbf{v}_i d\Omega - a_i(\mathbf{u}_i, \mathbf{v}_i) \right) + \int_{\Gamma} \{\mathbf{t}(\mathbf{u})\} \cdot \llbracket \mathbf{v} \rrbracket d\Gamma \\ - \int_{\Gamma} \{\mathbf{Dt}(\mathbf{u})[\mathbf{v}]\} \cdot \tilde{\mathbf{u}}^\Gamma d\Gamma + \int_{\Gamma} \{(1 - \beta) \mathbf{v}_1 + \beta \mathbf{v}_2\} \cdot \mathbf{t}^\Gamma d\Gamma + \gamma \int_{\Gamma} \tilde{\mathbf{u}}^\Gamma \cdot \llbracket \mathbf{v} \rrbracket d\Gamma. \end{aligned} \tag{19}$$

Only the first two terms on the right hand side remain in the case of  $\mathbf{u}^\Gamma = \mathbf{0}$  and  $\mathbf{t}^\Gamma = \mathbf{0}$ . As before, this expression is represented by the equation  $A(\mathbf{u}; \Delta \mathbf{u}, \mathbf{v}) = \ell(\mathbf{u}; \mathbf{v})$ , see (10), and we postpone the discussion of the parameters  $\beta$  and  $\gamma$  to Appendix A.

### Immersed finite element method

#### Finite element discretisation

The linearised weighted residual Eqs. (9) and (19) form the basis of a finite element discretisation. To this end, a domain  $\Omega_{\square}$  of a simple shape, typically rectangular, is defined such that it fully contains the reference domain  $\Omega$ . The following finite element discretisation is based on a triangulation of  $\Omega_{\square}$  instead of a geometry-conforming mesh of  $\Omega$  itself (see Fig. 3). We use piece-wise polynomial basis functions  $\varphi_I(\mathbf{x})$  and write for the approximated displacement field

$$\mathbf{u}^h(\mathbf{x}) = \sum_I \mathbf{u}_I \varphi_I(\mathbf{x}). \tag{20}$$

There is no constraint on the chosen finite element space, but if the surface  $\Gamma$  overlaps with the boundary of the embedding domain (that is if  $\Gamma_{\square} = \Gamma \cap \partial\Omega_{\square} \neq \emptyset$ ), it can be more convenient to use an essential treatment of displacement boundary conditions [8] along this boundary. On the other hand, if non-nodal basis functions (like, for instance, higher-order b-splines) are used as the finite element basis, the above presented weak incorporation of the boundary conditions works perfectly well on this boundary part  $\Gamma_{\square}$  too.

Let the support of the basis function  $\varphi_I$  be denoted by  $\text{supp}(\varphi_I)$ . Now all coefficients  $\mathbf{u}_I$  from the approximation (20) are discarded a priori if  $\text{supp}(\varphi_I) \cap \Omega = \emptyset$ . By  $\mathbb{S}$  we denote the set of the indices of the remaining coefficients and thus  $\{\varphi_I\}_{I \in \mathbb{S}}$  forms the full basis of the immersed finite element method. This basis is in general not stable [25] and requires further attention, which is given in “Numerical stability” section. Using the approximation (20), we reach the final system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{21}$$

with the matrix and vector coefficients

$$\begin{aligned} \mathbf{A}[I n_d + a, J n_d + b] &= A(\mathbf{u}; \mathbf{e}_b \varphi_J, \mathbf{e}_a \varphi_I) \\ \mathbf{x}[J n_d + b] &= (\Delta \mathbf{u}_J) \cdot \mathbf{e}_b \\ \mathbf{b}[I n_d + a] &= \ell(\mathbf{u}; \mathbf{e}_a \varphi_I) \end{aligned} \tag{22}$$

for the zero-based indices  $I, J \in \mathbb{S}$ , and using the coordinate directions  $0 \leq a, b < n_d$  ( $n_d$  being the spatial dimension of the problem) and Cartesian unit vectors  $\mathbf{e}_a$ . Although this immersed finite element method seemingly leads to the same type of linear system as a conventional, geometry-conforming FEM, there are technical differences which will be discussed in the following: the representation of the boundary or interface  $\Gamma$ , the quadrature of elements traversed by this boundary, and the stabilisation of the basis for

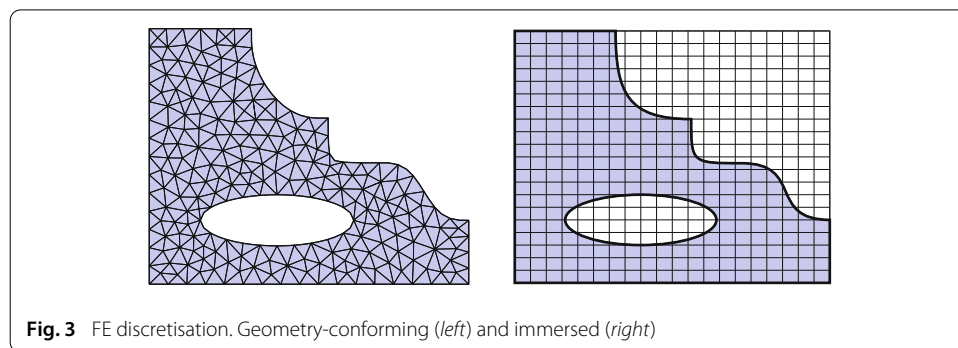


Fig. 3 FE discretisation. Geometry-conforming (left) and immersed (right)

such elements. The choice of the Nitsche parameters  $\gamma$  and  $\beta$  is analysed in the Appendix A.

Above expressions hold analogously for interface problems. The main difference is that the two fields  $\mathbf{u}_1$  and  $\mathbf{u}_2$  are approximated in fashion of (20) independently on the same background mesh of  $\Omega_\square$  which encompasses both sub-domains  $\Omega_1$  and  $\Omega_2$ . Consequently, the elements which are traversed by the material interface approximate both fields since the FE shape functions of the entire element are used even though the fields are only defined up to the interface on their respective side of the domain. Using two sets of shape functions on these elements allows us to represent a discontinuous derivative of the FE solution and can thus be compared to the element enrichment of XFEM [11]. A good illustration of this implementation detail can be found in [2].

**Signed distance functions**

The weak forms introduced in “Weak enforcement of boundary and interface conditions” section allow us to work with a finite element discretisation which is independent of the geometry, but still the volume and surface integrals,  $\int_\Omega (\cdot) d\Omega$  and  $\int_\Gamma (\cdot) d\Gamma$ , need geometry information. To this end, we classify the elements (for instance the quadrilaterals in the right picture of Fig. 3) by their location with respect to the physical domain  $\Omega$ . If  $\tau_I$  denotes any such element, we have the three cases:

- 1  $\tau_I \cap \Omega = \emptyset$ , the element is completely outside of  $\Omega$  and can be ignored,
- 2  $\tau_I \cap \Omega = \tau_I$ , the element is completely inside and its treatment is straightforward as in any geometry-conforming FEM,
- 3  $\tau_I \cap \Gamma \neq \emptyset$ , the element is traversed by the domain’s boundary and requires special consideration.

Note that elements adjacent to the boundary of the embedding mesh (for instance the left or bottom boundaries in the right picture of Fig. 3) technically fall into the third category, but do not pose any difficulty apart from the identification of the element faces which lie on that boundary.

For above classification it is sufficient to have an oriented representation of the surface  $\Gamma = \partial\Omega$ . Therefore, the surface is either closed or assumed to be extended beyond the boundaries of  $\Omega_\square$ . Here, we assume that  $\Gamma$  is either given analytically or is approximated by means of a surface mesh composed of surface elements  $\sigma_J$ ,

$$\Gamma \approx \Gamma^h = \bigcup_J \sigma_J. \tag{23}$$

In order to avoid the tedious task of intersecting volume elements  $\tau_I$  with surface elements  $\sigma_J$ , an *implicit* geometry representation is introduced. Therefore, the signed distance function [26] is used which is defined as

$$\text{dist}_\Gamma(\mathbf{x}) = s(\mathbf{x}) \min_{y \in \Gamma} |\mathbf{x} - \mathbf{y}|, \quad \text{with} \quad s(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega \\ -1 & \text{if } \mathbf{x} \notin \Omega. \end{cases} \tag{24}$$

In case of interface problems as introduced in “Material interfaces” section, the above definition of  $s(\mathbf{x})$  refers to  $\Omega_1$  and  $\Omega_2$  instead of  $\Omega$  and its complement  $\Omega_\square \setminus \Omega$ . If  $\Gamma$  is represented by a mesh  $\Gamma^h$ , the signed distance function  $\text{dist}_{\Gamma^h}$  with respect to this mesh is used instead. Moreover, only a piece-wise polynomial approximation of this function is used

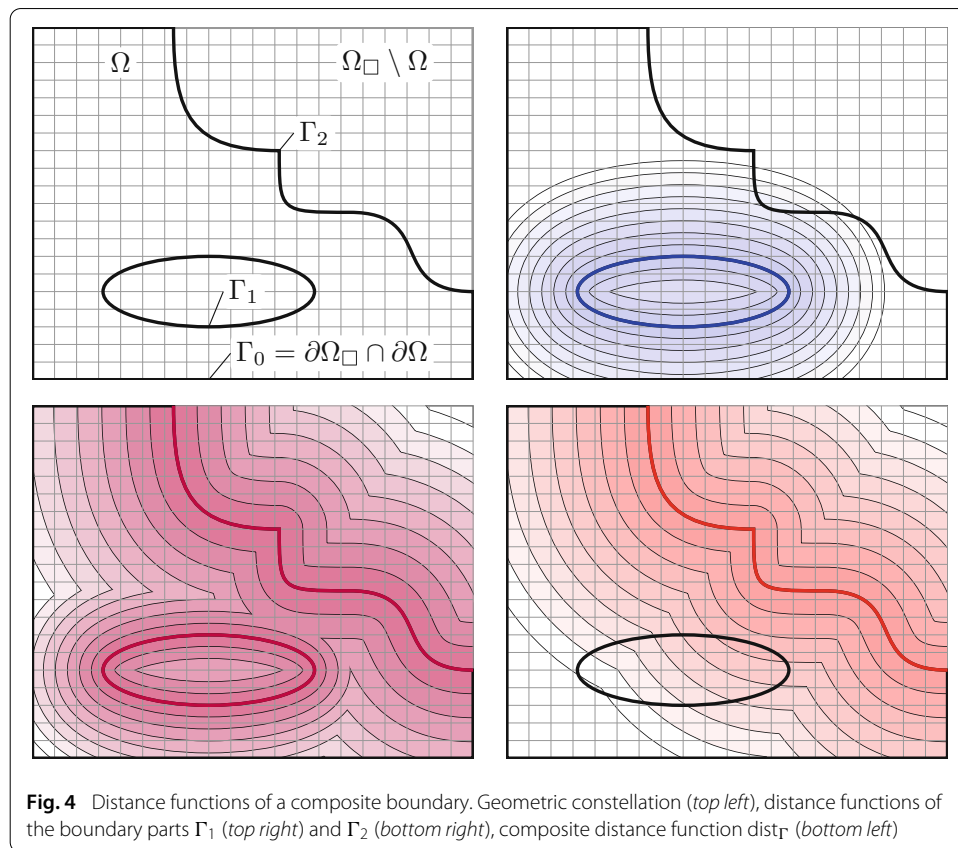
$$\text{dist}_\Gamma^h(\mathbf{x}) = \sum_K d_K \varphi_K(\mathbf{x}) \quad \text{with } d_K = \text{dist}_\Gamma(\mathbf{x}_K), \tag{25}$$

where  $\varphi_K$  are the nodal finite element shape functions (not necessarily the same as in the approximation (20)) and the coefficients  $d_K$  represent the value of the signed distance at the finite element nodes  $\mathbf{x}_K$ .

The representation (23) can be of higher polynomial degree, given by NURBS patches [1, 14, 27] or subdivision surfaces [28, 29]. But the computation of the coefficients  $d_K$  in (25) and the quadrature described below are non-trivial tasks if the  $\sigma_j$  have a degree higher than linear simplex elements (straight lines in two or flat triangles in three dimensions). In that case, the computation of the distances  $d_K$  requires the solution of nonlinear equations, see, for instance, [30]. In the rest of this work, the  $\sigma_j$  are always linear ( $n_d - 1$ )-simplex elements. Moreover, once only piece-wise linear elements are used for the surface representation (23), the optimal convergence rate of any higher-order method is impeded by this geometry approximation error, see [8].

Figure 4 shows a two-dimensional example where the boundary is composed of three parts:  $\Gamma_0$  is the part of the boundary of  $\Omega$  that coincides with the box boundary  $\partial\Omega_\square$  and does not require any special attention;  $\Gamma_1$  and  $\Gamma_2$  are separated parts which are immersed in the background grid. For the computation of the distance function  $\text{dist}_\Gamma$ , it is convenient to treat  $\Gamma_1$  and  $\Gamma_2$  separately as shown in the figure. The final distance function is then composed as the minimal value of these distances,

$$\text{dist}_\Gamma(\mathbf{x}) = \min(\text{dist}_{\Gamma_1}(\mathbf{x}), \text{dist}_{\Gamma_2}(\mathbf{x})). \tag{26}$$





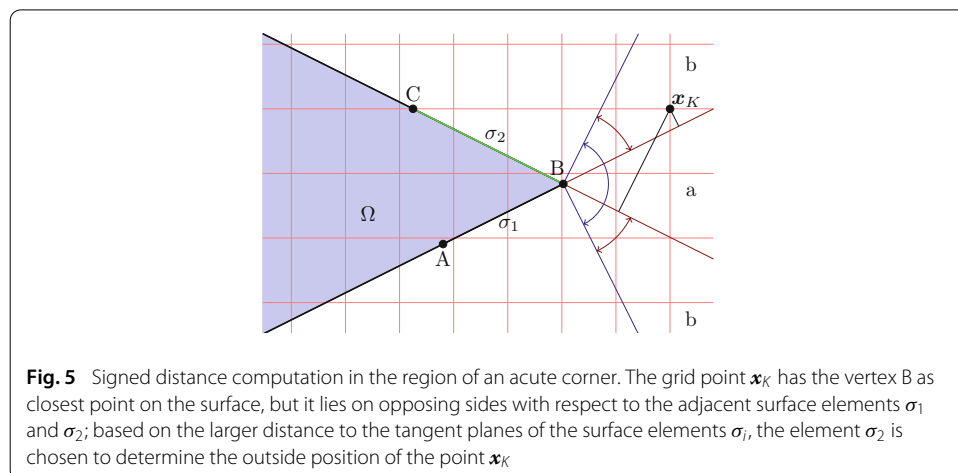
See also [31] for arithmetic with distance functions. Figure 4 shows the iso-curves of the individual distance functions  $\text{dist}_{\Gamma_i}$  as well as of the composite function  $\text{dist}_{\Gamma}$ . The extension of this approach to a larger number of immersed surfaces is straightforward.

Once the function  $\text{dist}_{\Gamma}$  has been determined, the above classification of volume elements  $\tau_I$  is carried out by means of the nodal values  $d_K$  of the distance function: if all  $d_K$  of the element  $\tau_I$  are strictly positive (negative), the element is inside (outside) of the domain. If a change in sign of the  $d_K$  occurs,  $\tau_I$  is traversed by the immersed boundary  $\Gamma$ .

It remains to outline how the coefficients  $d_K$  for a given surface are computed. In case of an analytic surface representation by an implicit function, these coefficients are calculated directly. In case of an immersed surface mesh, one needs to find the surface element  $\sigma_K^*$  which contains the point  $\mathbf{x}_K^*$  closest to  $\mathbf{x}_K$ , see for instance [32] for such basic primitive tests as the closest point on a triangle to a point. With the knowledge of the closest element  $\sigma_K^*$ , it can be decided if  $\mathbf{x}_K$  lies on the positive or the negative side of this element in order to determine the sign  $s(\mathbf{x}_K)$  as defined in (24). This decision is based on the premise that the surface mesh is well oriented. Note that, when the closest point falls on an edge or a vertex, ambiguities can arise for the decision if a point is inside or outside the surface mesh [26], see the case shown in Fig. 5.

At the acute corner in the figure, the region of points whose closest point is the vertex B, is delimited by the outer cone. For all points in this cone,  $\sigma_1$  and  $\sigma_2$  are possible choices as closest surface element. The cone contains the region 'a' in which the points are all outside with respect to both elements. The points in region 'b' are outside with respect to one of the possible closest surface elements and inside with respect to the other. Hence, for this region the mentioned ambiguity can occur. One solution to this problem is to introduce angle-weighted vertex normal vectors [26], but this requires extra data structures. Here we choose the simpler approach shown in Fig. 5: the point  $\mathbf{x}_K$  has a larger distance to the extension plane of  $\sigma_2$  than to the extension plane of  $\sigma_1$ . This distance is given by the inner product of the element normal vector and the distance vector between the considered grid point and the closest surface point (here, B). Choosing the element with a larger value of this distance resolves the ambiguity. The method is also used in three dimensions with the only difference being a larger set of candidates as closest elements.

Finally, we consider the numerical complexity of the distance function computation. If there are  $N_{\Gamma}$  elements in the surface mesh and  $N_{\Omega}$  nodes in the volume mesh, a brute-



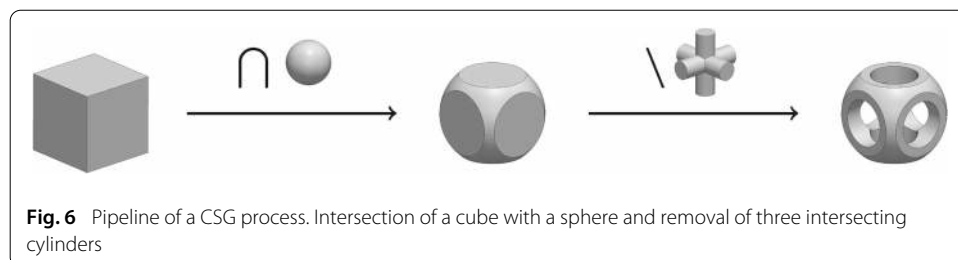
force approach requires  $N_\Gamma \times N_\Omega$  closest point computations. In many cases, this number can be substantially reduced by precomputing a bounding box [32] of the surface  $\Gamma^h$  and assigning a default value for the  $d_K$  of nodes outside of this box, but the essential complexity remains of order  $\mathcal{O}(N_\Gamma \times N_\Omega)$ . Complexity reduction is possible by generation of a hierarchy of bounding boxes [32] or using so-called *marching* methods, see e.g. [33].

**Constructive solid geometry modelling**

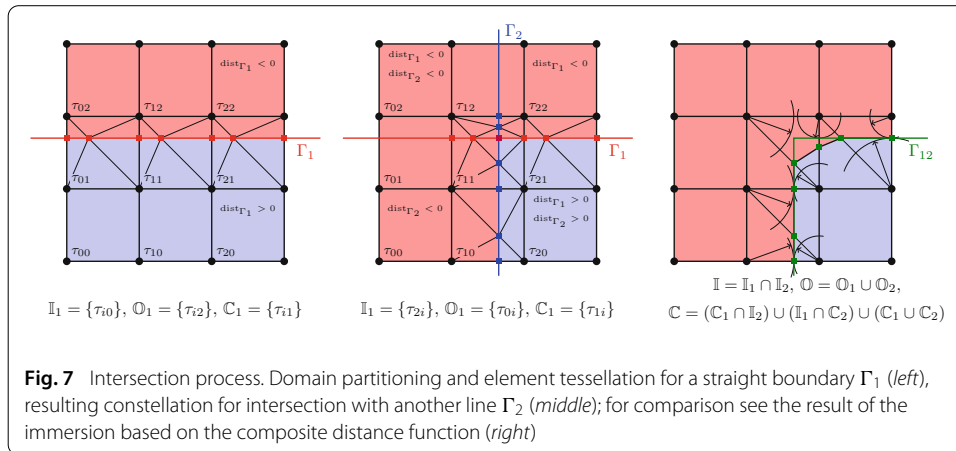
Now, we consider a different approach for integrating finite element analysis with geometric design, similar to the ideas presented in [34]. Specifically, we consider the construction of a three-dimensional geometry by means of CSG, see, for instance, [35,36]. An example of such a modelling process is given in Fig. 6, where one begins with a cube as a workpiece and performs set operations with other geometric primitives until the desired geometry is obtained. These operations are commonly union  $\cup$ , intersection  $\cap$ , subtraction  $\setminus$  and the set complement  $()^c$ . Based on De Morgan’s laws [36], it suffices to work with the canonical operations intersection and complement, and represent the other two as compositions thereof, more precisely  $A \cup B = (A^c \cap B^c)^c$  and  $A \setminus B = A \cap B^c$ .

The conventional finite element approach is to work through such a CSG pipeline, export a geometry representation and use a mesh generation software to create a body-fitted volume mesh for the numerical analysis. The direct modification for an immersed finite element method is to export a surface representation of the geometry and embed this into the mesh by the methods described in “Immersed finite element method” section. Here, a third way is suggested in which the set operations are directly applied to the embedding (non-conforming) volume mesh. As outlined above, it suffice to provide the complement and intersection operations only. The former is trivially achieved: the use of a signed distance function generates an in- and an outside partition of the mesh, reversing these partitions gives the complement. For this reason, all that need be explained is the intersection operation.

A simple two-dimensional example in Fig. 7 demonstrates the intersection operation: first the intermediate domain  $\Omega_1$  is given via the distance function of a straight line  $\Gamma_1$ , afterwards a second distance function to the line  $\Gamma_2$  yields the final domain  $\Omega = \{x \in \Omega_\square : \text{dist}_{\Gamma_1}(x) > 0 \text{ and } \text{dist}_{\Gamma_2}(x) > 0\}$ . For sake of clarity, let us discuss the individual steps in this picture. First the line  $\Gamma_1$  is embedded into the shown  $3 \times 3$ -grid which fills out the square domain  $\Omega_\square$ . The elements  $\tau_{i0}$  are strictly inside the intermediate domain  $\Omega_1 = \{x \in \Omega_\square : \text{dist}_{\Gamma_1}(x) > 0\}$  and form the set  $\mathbb{I}_1$ . The elements  $\tau_{i2}$  are strictly outside and form the set  $\mathbb{O}_1$ . Now, the remaining elements form the set  $\mathbb{C}_1$  and are triangulated such that the embedded boundary is approximated by triangle edges. The squares are first



**Fig. 6** Pipeline of a CSG process. Intersection of a cube with a sphere and removal of three intersecting cylinders



subdivided into two triangles each and then every such triangle is intersected by means of the nodal values of the signed distance function  $\text{dist}_{\Gamma_1}$  [37]. The resulting outcome is the left picture in Fig. 7 where the red square-shaped marks indicate the location of the intersection points.

In the second embedding step, the distance function  $\text{dist}_{\Gamma_2}$  is used which gives rise to the element sets  $\mathbb{I}_2, \mathbb{O}_2$  and  $\mathbb{C}_2$ . All elements which belong to the outside are directly assigned to the complementary domain  $\Omega_{\square} \setminus \Omega$ , that is  $\mathbb{O} = \mathbb{O}_1 \cup \mathbb{O}_2$ . On the other hand, all elements of  $\mathbb{I}_1$ , which also belong to  $\mathbb{I}_2$ , are inside the final domain  $\Omega$ , hence  $\mathbb{I} = \mathbb{I}_1 \cap \mathbb{I}_2 = \{\tau_{20}\}$ . Finally, there are the intersection cases. Elements belonging to  $\mathbb{C}_1$  and  $\mathbb{I}_2$  ( $\tau_{21}$ ) keep their status and sub-division. Elements from  $\mathbb{I}_1$  and  $\mathbb{C}_2$  ( $\tau_{10}$ ) are subject to the same decomposition methods as  $\mathbb{C}_1$ . It remains to discuss the situation of the elements which belong to  $\mathbb{C}_1 \cap \mathbb{C}_2$ ; the ones which are intersected by both boundaries, and in our example of Fig. 7 this is the element  $\tau_{11}$ . In this case, simply the composing triangles are intersected with  $\Gamma_2$  as if they were elements of their own. Proper categorisation of these simplex shapes defines the final domain  $\Omega$  and its complement  $\Omega_{\square} \setminus \Omega$ , see the middle picture of Fig. 7.

The advantage of this approach becomes clear when looking at the right picture of Fig. 7. Shown is the result for the same target domain  $\Omega$ , but first the composition of the individual distance functions  $\text{dist}_{\Gamma_i}$  is computed according to expression (26) and then the element intersections are constructed. Clearly, in the right picture the corner is chamfered whereas in the above outlined approach this geometric feature is preserved. This is the distinctive characteristic of the presented idea: by successively embedding the geometry primitives into the mesh, the sharp features at the primitive intersections are preserved. It is important to remark that the boundaries  $\Gamma_i$  have been represented exactly in this example, but this is solely owed to the fact that they are straight lines. In the more general situation of curved boundaries, they are again represented on the finite element mesh by piece-wise linear simplex elements. But, even though these surrogate boundaries do not exactly reproduce the given geometry, the here presented approach still allows to represent corners or edges at the intersection locations of the original primitives which lie inside of the finite elements.

The presented method for CSG modelling based on finite element meshes is straightforward to extend to three dimensions. In the plane case outlined so far, the rectangular

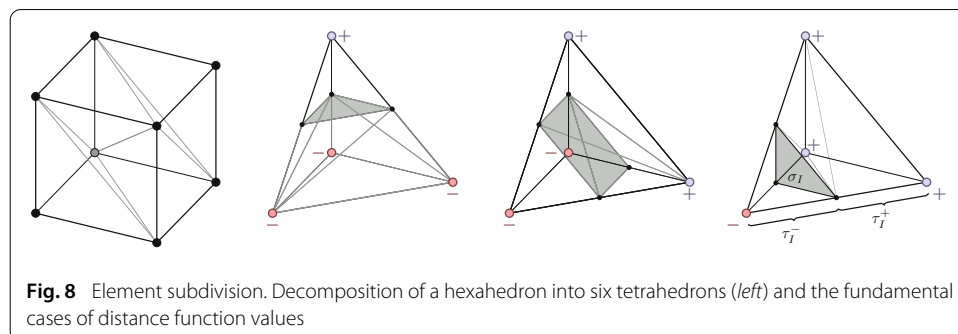
elements are subdivided into two triangles which themselves are triangulated in order to recover the implicit surface in the form of triangle edges. This approach is akin to a two-dimensional version of marching cubes and in three dimensions we make use of a similar technique. The used three-dimensional element shapes are either tetrahedrons or hexahedrons. Figure 8 shows how a hexahedron is decomposed into six tetrahedrons such that it remains to consider this shape only. Given a tetrahedron with values of the signed distance function at its vertices we can classify the cases shown in the right part of the figure. Based on linear interpolation along the edges the zeros of the distance function are recovered and give rise to two volume tessellations  $\tau_I^-$  and  $\tau_I^+$  whose common faces form the triangulated surface  $\sigma_I$ . Using these decompositions, the above outlined intersection operations of two geometry primitives can be carried out analogously in three dimensions.

Alternative approaches for increasing the quality of implicit geometry representations in the vicinity of sharp features (such as edges and corners) exist. In [38] the operations of surface reconstruction by means of marching cubes and the distance function computation are combined in order to generate a so-called directed distance field allowing for a better resolution of surface features. On the other hand, enriched distance functions are presented in [39] where additional edge and vertex descriptors augment the distance geometry representation. Although both approaches are promising concepts in the context of immersed finite element methods, they are not further considered in this work.

We conclude this paragraph by noting that the here used tessellation techniques also help to construct numerical integration schemes for the elements that are traversed by the boundary or interface. The cut elements are general polytopes for which quadrature rules are not easily obtained. There are many techniques that address this problem, such as moment-fitting [40], surface-only integration [41], and adaptive decomposition of the integration region [14,42]. But since we have a tessellation in simplex shapes already available, we use composite Gauß type quadrature rules, see e.g., [11].

**Numerical stability**

Up to now, it has been shown how to derive an immersed finite element method for boundary value and interface problems, see (9) and (19), and how to compute the matrix coefficients of the linear system of equations. But the stable solution of this final system of Eq. (21) remains to be discussed, especially in view of the method’s parameters  $\gamma$  (for boundary value and interface problems) and  $\beta$  (for interface problems only).



**Sources of instability**

As an illustrative example, consider a one-dimensional problem

$$\begin{aligned}
 -\alpha u'' &= f & x \in (0, x_\varepsilon) \\
 u &= 0 & x = 0 \\
 \alpha u' &= 0 \text{ or } u = 0 & x = x_\varepsilon
 \end{aligned}
 \tag{27}$$

for the domain  $\Omega = (0, x_\varepsilon)$  and a constant material parameter  $\alpha$ . The boundary conditions are a prescribed value of  $u = 0$  at the left end and either a zero derivative (homogeneous Neumann) or a zero function value (homogeneous Dirichlet) at the right end. Let  $\Omega_\square = (0, 2h)$  be the embedding domain and two linear finite elements of size  $h$  are used for the discretisation, see Fig. 9. First, we consider the case with a homogeneous Neumann boundary condition at the right end. The left-side boundary condition is going to be incorporated essentially and the system matrix becomes

$$K_N = \frac{\alpha}{h} \begin{pmatrix} 1 + \varepsilon & -\varepsilon \\ -\varepsilon & \varepsilon \end{pmatrix}.
 \tag{28}$$

Obviously, for  $\varepsilon \rightarrow 1$  this matrix recovers the standard finite element matrix for this problem with its known properties. The eigenvalues of this matrix have the values

$$\lambda_{1,2} = \frac{\alpha}{2h} \left[ 1 + 2\varepsilon \pm \sqrt{4\varepsilon^2 + 1} \right].
 \tag{29}$$

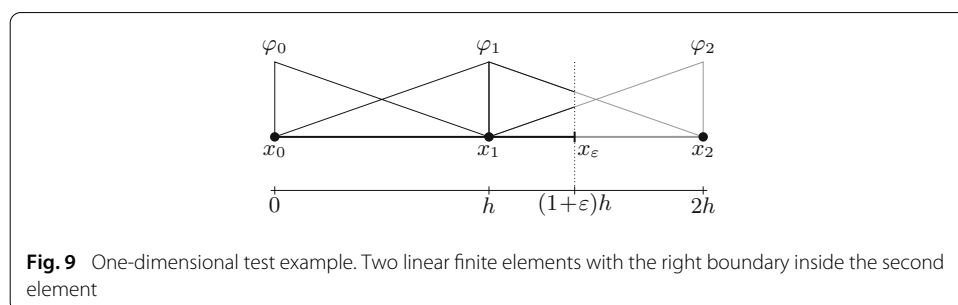
Clearly, the smaller eigenvalue goes to zero for the limit  $\varepsilon \rightarrow 0$ , that is the case of a vanishing cut element. As expected, the matrix  $K_N$  is ill-conditioned for this limit.

We now turn to the Dirichlet case and evaluate the left-hand-side of expression (7) for this simple test problem. The resulting stiffness matrix has the form (replacing the surface integrals by point evaluation at  $x_\varepsilon$ )

$$\begin{aligned}
 \frac{h}{\alpha} K_D &= \begin{pmatrix} 1 + \varepsilon & -\varepsilon \\ -\varepsilon & \varepsilon \end{pmatrix} - \begin{pmatrix} \varepsilon - 1 & 1 - \varepsilon \\ -\varepsilon & \varepsilon \end{pmatrix} - \begin{pmatrix} \varepsilon - 1 & -\varepsilon \\ 1 - \varepsilon & \varepsilon \end{pmatrix} \\
 &+ \gamma \frac{h}{\alpha} \begin{pmatrix} (1 - \varepsilon)^2 & \varepsilon(1 - \varepsilon) \\ \varepsilon(1 - \varepsilon) & \varepsilon^2 \end{pmatrix}.
 \end{aligned}
 \tag{30}$$

Note that the expression for the system matrix has been multiplied by the factor  $\frac{h}{\alpha}$ . The expressions of the eigenvalues of  $K_D$  are not easily determined, but the condition  $\det(K_D) > 0$  is more workable. Note that since the trace of the matrix is positive and equals  $\lambda_1 + \lambda_2$ , the condition of a positive determinant (recall  $\det(K_D) = \lambda_1 \lambda_2$ ) is sufficient for positive definiteness. One gets

$$\det(K_D) = \frac{1}{h^2} \left( \frac{\varepsilon \gamma h}{\alpha} - 1 \right) (1 + \varepsilon) > 0 \Rightarrow \gamma > \frac{\alpha}{h\varepsilon}.
 \tag{31}$$



**Fig. 9** One-dimensional test example. Two linear finite elements with the right boundary inside the second element

Fulfilment of this condition guarantees that the matrix is positive definite for a fixed mesh size  $h$ , but unfortunately it implies  $\gamma \rightarrow \infty$  for  $\varepsilon \rightarrow 0$ . The use of a very large value for  $\gamma$  can lead to undesired numerical problems.

In the case of the interface problems and formulation (19), the situation is slightly better. The extra parameter  $\beta$  can be adjusted in a smart way such that a finite value of  $\gamma$  is always achievable. Such a choice is proposed in [43] where  $\beta$  depends on the material parameters of the subdomains and the sizes of the cut elements,  $|\tau_I \cap \Omega_i|$ . Using this approach, the system matrix has always positive eigenvalues (for the considered problem class) with a finite value of  $\gamma$ . Nevertheless, the minimal eigenvalue goes to zero for vanishing sizes of the cut elements. Even though the parameter choices by [43] show a good performance in terms of the quality of the numerical results, the matrix condition number still cannot be bounded for a fixed mesh and arbitrary interface locations.

### **Stabilisation**

The above indicated sources of numerical instability all stem from the same situation that for some degrees of freedom, the intersection of the support of their associated shape functions with the physical domain becomes very small,

$$s_I = |\text{supp}(\varphi_I) \cap \Omega| \ll h, \quad (32)$$

where  $\text{supp}(\varphi_I)$  denotes the support of shape function  $\varphi_I$  and  $h$  is a measure of the mesh size on  $\Omega_\square$ . In all above cases, Neumann, Dirichlet, or interface problem, this leads to severe ill-conditioning of the final system matrix. To solve this problem, the following approaches have been proposed, among others,

- S-1 Discarding all degrees of freedom with support intersection below a certain threshold,  $s_I < \varepsilon h$ ;
- S-2 Adding a face-based stabilisation term [16];
- S-3 Constraining degenerate degrees of freedom [6,12].

As reported, among others, in [12], the approach S-1 leads to a loss of approximation order. Although appealing due to its simplicity, this drawback can be prohibitive in some applications. An ad-hoc approach to remedy the stability problem is to locally adapt the finite element mesh in order to avoid the problem of too small values of  $s_I$ . Even though simple at first sight, a robust realisation of this idea in three dimensions is not straightforward and mesh entanglement needs to be avoided. The support size  $s_I$  is increased if specific nodes are moved away from the surface  $\Gamma$ , but there is an interesting alternative in which the points are snapped to the surface thereby generating a conforming mesh, see [44] for two-dimensional analysis of this idea.

Another approach, S-2, is proposed in [16] where the jump of the function gradients across certain element faces is added to the weak form in order to guarantee stability of the method. Other than the result (31), the system matrix stays well-conditioned for small values of  $\gamma$  in the limit  $\varepsilon \rightarrow 0$ . This approach requires to evaluate surface integrals over interior mesh faces, a technicality which requires additional data structures in many codes, but does not add much to the overall difficulty of implementing an immersed finite element method. Nevertheless there is a drawback with this approach, since it introduces another weighting factor whose adjustment is not straightforward: for too small values of

this factor the stabilisation effect disappears and for too large values the method’s accuracy is affected [16].

Finally, we consider S-3 which relies on the concept of coupling degrees of freedom with too small supports to other degrees of freedom from the interior of the domain. In order to outline this approach, the degrees of freedom shall first be classified according to the size  $s_I$  of the intersection of their support with the domain, as defined in (32). In “Finite element discretisation” section, the set  $\mathbb{S}$  has been introduced which contains all indices of shape functions for which  $s_I$  is larger than zero. Introducing a threshold  $\hat{s}$ , the set  $\mathbb{S}$  is now decomposed into the disjoint index sets,  $\mathbb{A}$  and  $\mathbb{B}$  with definition

$$\mathbb{A} = \{I \in \mathbb{S}: s_I \geq \hat{s}\} \quad \text{and} \quad \mathbb{B} = \{I \in \mathbb{S}: s_I < \hat{s}\}. \tag{33}$$

The threshold  $\hat{s}$  used in this classification has to depend on the mesh size  $h$  and should not be larger than one typical element size. The basic idea of Höllig et al. [12] is to constrain degrees of freedom from the set  $\mathbb{B}$  to suitably chosen degrees of freedom from  $\mathbb{A}(J)$ , a subset of  $\mathbb{A}$ ,

$$\forall J \in \mathbb{B}: \quad \mathbf{u}_J = \sum_{I \in \mathbb{A}(J) \subset \mathbb{A}} c_{IJ} \mathbf{u}_I \tag{34}$$

where the coefficients  $c_{IJ}$  will be discussed further below. These constraints give rise to the modified shape function basis

$$\begin{aligned} \mathbf{u}^h(\mathbf{x}) &= \sum_{I \in \mathbb{A}} \mathbf{u}_I \varphi_I(\mathbf{x}) + \sum_{J \in \mathbb{B}} \mathbf{u}_J \varphi_J(\mathbf{x}) \\ &= \sum_{I \in \mathbb{A}} \mathbf{u}_I \varphi_I(\mathbf{x}) + \sum_{J \in \mathbb{B}} \left( \sum_{K \in \mathbb{A}(J)} c_{KJ} \mathbf{u}_K \right) \varphi_J(\mathbf{x}) \\ &= \sum_{I \in \mathbb{A}} \mathbf{u}_I \left( \varphi_I(\mathbf{x}) + \sum_{J \in \mathbb{B}(I)} c_{IJ} \varphi_J(\mathbf{x}) \right) = \sum_{I \in \mathbb{A}} \mathbf{u}_I \tilde{\varphi}_I(\mathbf{x}). \end{aligned} \tag{35}$$

In this reordering of the finite element approximation (20) a new set  $\mathbb{B}(I)$  is used which contains all indices  $J$  from  $\mathbb{B}$ , such that  $I \in \mathbb{A}(J)$ . For the implementation of this stabilisation method, it is sufficient to work with expression (34), but the result of (35) demonstrates that effectively a modified shape function basis  $\{\tilde{\varphi}_I\}_{I \in \mathbb{A}}$  is generated and illustrates the notion of *extended splines* as given in [12]. Note also that  $\mathbb{B}(I) = \emptyset$  for all degrees of freedom that are not in the vicinity of the boundary and in that case  $\tilde{\varphi}_I = \varphi_I$ , so that most shape functions are not affected. Since the support size of the basis functions  $\tilde{\varphi}_I$  is larger, the bandwidth increases for these degrees of freedom. Therefore, it has to be remarked that only degrees of freedom in the vicinity of the boundary are affected. Moreover, the storage requirement of the final system matrix is of course not larger than it would be for the original (unstable) basis functions  $\varphi_I$ .

There are two open questions when using this approach: (i) the choice of the index set  $\mathbb{A}(J)$  associated to  $J$  and (ii) the values of the constraint weights  $c_{IJ}$ . The origin of this approach, as introduced in [12], is to stabilise b-spline discretisations. In this particular situation, the underlying mesh is logically Cartesian and an explicit expression of the coefficients  $c_{IJ}$  can be given as a function of the multi-indices used to label that grid. See also [6] for a more intuitive interpretation of the arising extrapolation of Lagrange polynomials and its efficient implementation. The aim of this stabilisation procedure is to maintain the convergence order of the method and therefore to not lose the polynomial



approximation quality of the approximation (20) due to the constraints (34). In other words, the modified basis functions  $\tilde{\varphi}_I$  introduced in (35) have to represent the same polynomials as the  $\varphi_I$  themselves.

In order to outline the procedure for obtaining  $\mathbb{A}(J)$  and the corresponding coefficients  $c_{IJ}$ , consider the situation depicted in Fig. 10. The degree of freedom  $\mathbf{u}_J$ ,  $J \in \mathbb{B}$ , resides at node  $\mathbf{x}_J$  and the size of the intersection of the support (hatched in the picture) with the domain  $\Omega$  is below the threshold  $\hat{s}$ . Searching through the elements in the vicinity of  $\mathbf{x}_J$ , one finds the element  $\tau_{K(J)}$  whose connected degrees of freedom all belong to  $\mathbb{A}$ . Any element entirely inside the domain  $\Omega$  fulfils this condition. Normally, many such elements can be found and the closest is selected, where the distance between the element middle point and  $\mathbf{x}_J$  is a possible way to measure the proximity. The selected element  $\tau_{K(J)}$  gives rise to the index set  $\mathbb{A}(J) \subset \mathbb{A}$  associated with  $\mathbf{u}_J$ . Formally, we can write

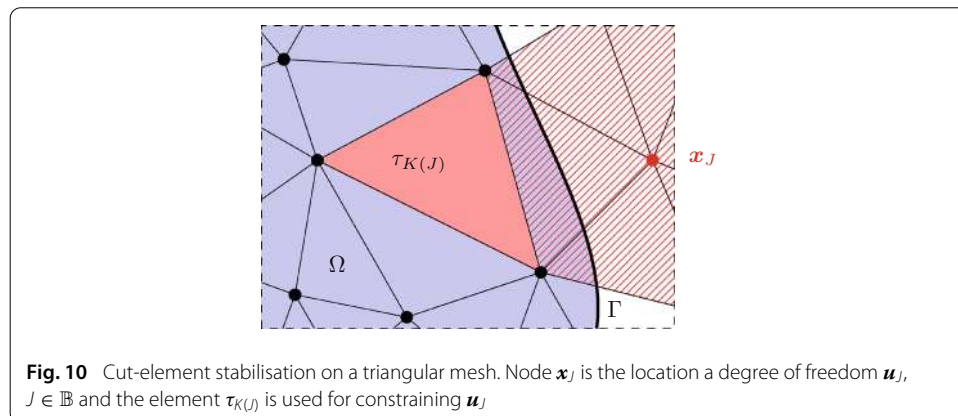
$$\mathbb{A}(J) = \{I \in \mathbb{A} : \text{supp}(\varphi_I) \cap \tau_{K(J)} \neq \emptyset\}. \tag{36}$$

Once this set is defined, the weights  $c_{IJ}$  are calculated by evaluation of the basis of  $\tau_{K(J)}$  at the node  $\mathbf{x}_J$ ,

$$\forall I \in \mathbb{A}(J) : c_{IJ} = \varphi_I(\mathbf{x}_J). \tag{37}$$

This choice of weights is an extension of the idea given in [15] where the weights are defined for non-uniform b-splines as dual functionals applied to the polynomials in a chosen grid element. Here the point evaluation of (37) is the corresponding dual functional of Lagrange polynomials [8]. Note that  $\mathbf{x}_J \notin \tau_{K(J)}$  and thus  $c_{IJ}$  represents an extrapolation of the polynomial basis spanned in  $\tau_{K(J)}$  to the outside point  $\mathbf{x}_J$ , see also Fig. 10. The stabilisation procedure can be summarised as follows

- 1 categorise  $\mathbb{A}$  and  $\mathbb{B}$  using a threshold  $\hat{s}$ , see (32) and (33)
- 2 for all  $J \in \mathbb{B}$ 
  - find  $\tau_{K(J)}$  with all degrees of freedom from  $\mathbb{A}$  that is close to  $\mathbf{x}_J$ ,
  - define the constraint coefficients as  $c_{IJ} = \varphi_I(\mathbf{x}_J)$  for all  $I \in \mathbb{A}(J)$
- 3 assemble the final system of equations using the constraint equations (34) applied to test and trial spaces
- 4 after solving the global system, calculate the constrained degree of freedom  $\mathbf{u}_J$  with  $J \in \mathbb{B}$  according to (34).



**Fig. 10** Cut-element stabilisation on a triangular mesh. Node  $\mathbf{x}_J$  is the location a degree of freedom  $\mathbf{u}_J$ ,  $J \in \mathbb{B}$  and the element  $\tau_{K(J)}$  is used for constraining  $\mathbf{u}_J$



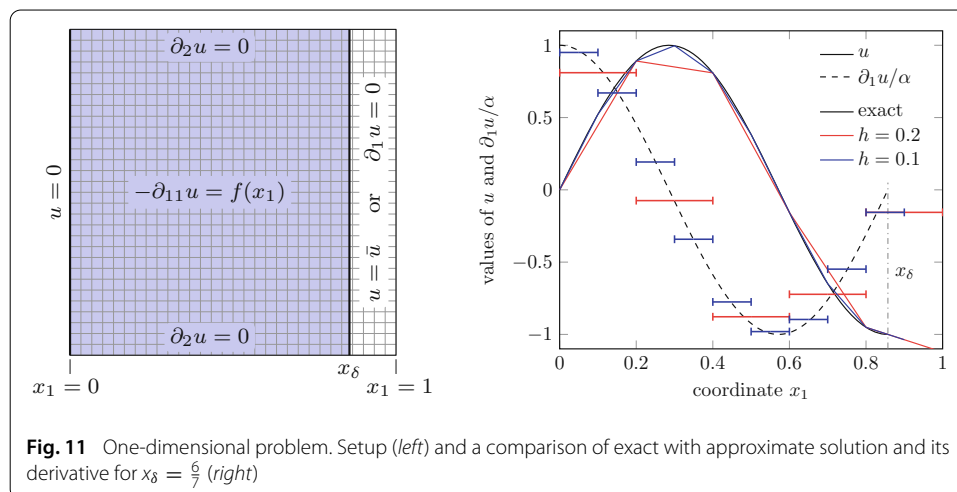
With respect to the implementation a few remarks have to be made. The code has to be able to search the elements in the neighbourhood of a given node. For instance, the element  $\tau_{K(j)}$  in Fig. 10 does not lie in the support of  $\varphi_j$  but in the ring of elements around that support. Theoretically, for very extreme shapes of  $\Gamma$  the nearest  $\tau_{K(j)}$  to  $\mathbf{x}_j$  could lie far away, but here we assume that the mesh is fine enough such that there is always an element nearby. Cusp-shaped domains are excluded from the onset. In addition, one has to evaluate the shape functions of  $\tau_{K(j)}$  at  $\mathbf{x}_j$  and this requires to find first the reference coordinate  $\xi_j$  (outside of the reference element) such that the geometry representation of the chosen element represents  $\mathbf{x}_j$  when evaluated at this coordinate, that is  $\mathbf{x}_{K(j)}(\xi_j) = \mathbf{x}_j$ . Here we restrict ourselves to meshes in which all elements are an affine transformation of the reference element. Higher-order geometry representations of the volume mesh are excluded, but they are also not necessary since the mesh, by design of the immersed method, need not conform to the geometry of  $\Omega$ .

### Numerical examples

At last, a few numerical examples are presented in order to study and demonstrate the performance of the immersed finite element method as presented here. Unless indicated otherwise, the spatial discretisation of all problems is carried out with linear finite elements. As shown in the appendix, the Nitsche parameter is chosen as  $\gamma = \gamma_0 \frac{\alpha}{h}$  with the mesh width  $h$ , the representative material parameter  $\alpha$  and a dimensionless scalar  $\gamma_0$ . The default choices for this parameter is  $\gamma_0 = 10$  and for interface problems the additional parameter is chosen as  $\beta = 0.5$ . The threshold  $\hat{s}$  used to distinguish between the degree of freedom sets  $\mathbb{A}$  and  $\mathbb{B}$  in the stabilisation of “Stabilisation” section is set to the size of one element.

### Convergence and robustness analysis

At first, the method’s performance under variation of various parameters is assessed. For this purpose, an essentially one-dimensional Poisson problem is used as depicted in Fig. 11, left, with a forcing function  $f(\mathbf{x}) = \alpha^2 \sin(\alpha x_1)$  and  $\alpha = \frac{3\pi}{2x_\delta}$ . The resulting exact solution is then  $u(\mathbf{x}) = \sin(\alpha x_1)$ . The signed distance function is  $\text{dist}_\Gamma(\mathbf{x}) = x_\delta - x_1$  and the boundary is represented exactly. At first, this problem is analysed using a structured mesh as shown in the figure. Figure 11, on the right, shows the analytic solution (solid



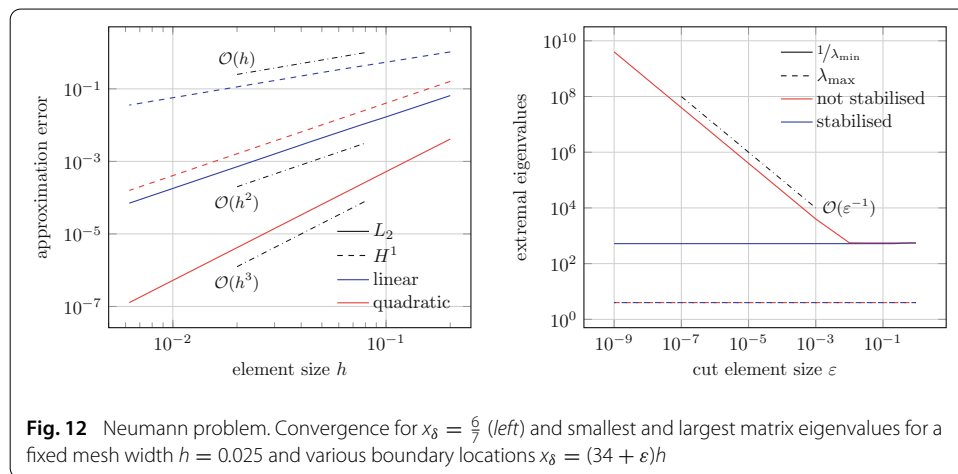
**Fig. 11** One-dimensional problem. Setup (left) and a comparison of exact with approximate solution and its derivative for  $x_\delta = \frac{6}{7}$  (right)

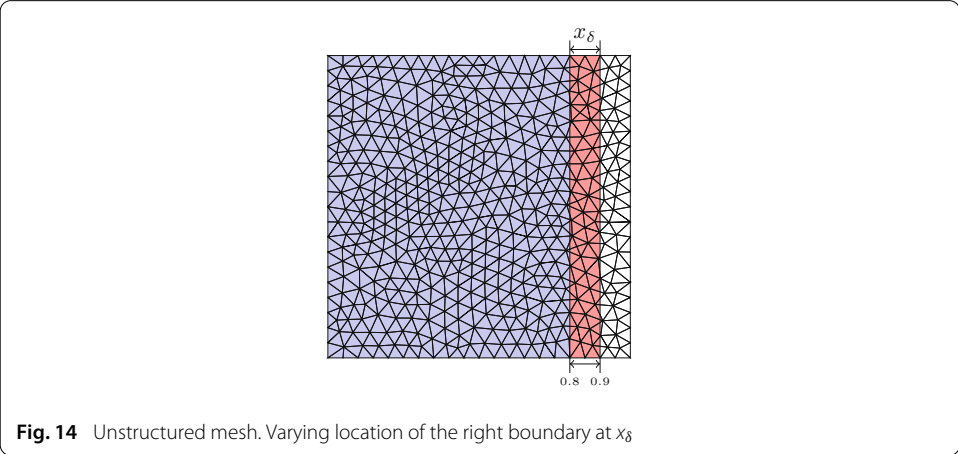
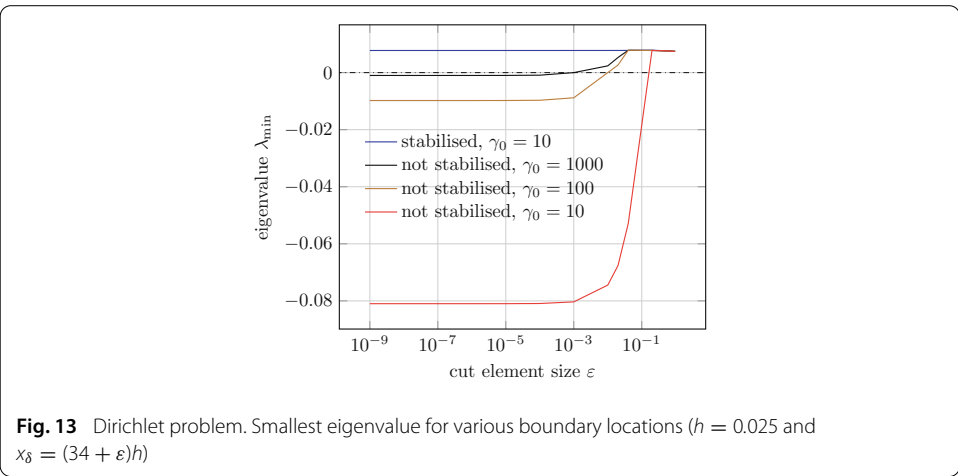
black line) along the  $x_1$ -axis and its derivative (dashed line) for a boundary location at  $x_\delta = \frac{6}{7}$ . The approximations  $u^h$  and  $\partial_1 u^h$  for a mesh with  $5 \times 5$  elements (red) and a  $10 \times 10$  element (blue) are also displayed. One can see that the numerical approximation  $u^h$  coincides with the analytic solution at the finite element nodes and, moreover, at the boundary location at  $x_\delta$ .

The convergence of the method is shown in the left of Fig. 12, where the numerical errors in  $L_2$ -norm and  $H^1$ -seminorm are shown for an approximation with linear and a quadratic Lagrange polynomials. These results exhibit the expected optimal convergence rates [8]. In this graph, the boundary location is held fixed at  $x_\delta = \frac{6}{7}$  and the mesh width  $h$  is decreased. On the other hand, the right side of Fig. 12 shows the smallest and largest eigenvalues of the system matrix in dependence of the boundary location for a non-stabilised implementation and for the stabilisation presented in “Stabilisation” section. Here, a fixed  $40 \times 40$  mesh is used and the location of the boundary is at  $x_\delta = (34 + \varepsilon)h$  with the parameter  $0 \leq \varepsilon \leq 1$ . A Neumann boundary condition at  $x_\delta$  is considered and, hence, one has always  $a(u^h, u^h) > 0$  and  $\lambda_{\min} > 0$ . One can clearly see that  $\lambda_{\min} \in \mathcal{O}(\varepsilon)$  for small  $\varepsilon$  and for the non-stabilised case (note that the figure shows in fact the inverse  $\frac{1}{\lambda_{\min}}$ ). Clearly, the matrix condition number grows without bound. The stabilisation as outlined in “Stabilisation” section, however, guarantees a constant value of  $\lambda_{\min}$  well above zero. The largest eigenvalues coincide for both cases.

Now we turn to the problem with a Dirichlet boundary condition at  $x_\delta$ . Using the same variation of the location of this boundary as above, Fig. 13 shows the smallest eigenvalue  $\lambda_{\min}$  for the stabilised method and for the non-stabilised method for various values of  $\gamma_0$  (recall that  $\gamma = \frac{\gamma_0}{h}$ ). One can see that without stabilisation the considered minimal eigenvalue changes sign for decreasing values of  $\varepsilon$  rendering the system matrix indefinite (and singular when the zero is crossed). In order to force  $\lambda_{\min} > 0$  one can increase the value of  $\gamma_0$ , but for  $\varepsilon \rightarrow 0$  this value grows without bound and one gets effectively  $\lambda_{\max} \rightarrow \infty$  which likewise deteriorates the condition number of the matrix, as already discussed in “Sources of instability” section.

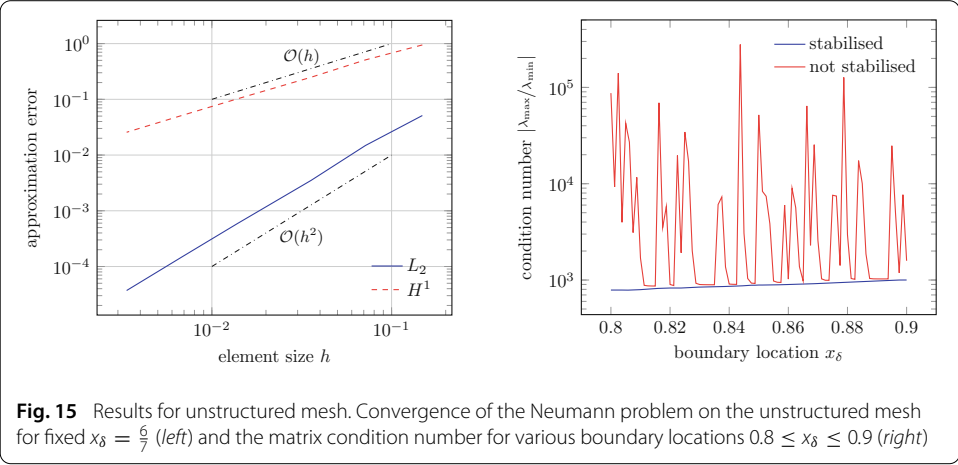
Next, the stabilisation technique is applied to an unstructured mesh as shown in Fig. 14. Note that this case is not covered by the original idea of this technique as given by [12] which was only designed for b-spline basis functions on structured meshes. The left of Fig. 15 shows the convergence of the stabilised method for linear triangle elements and

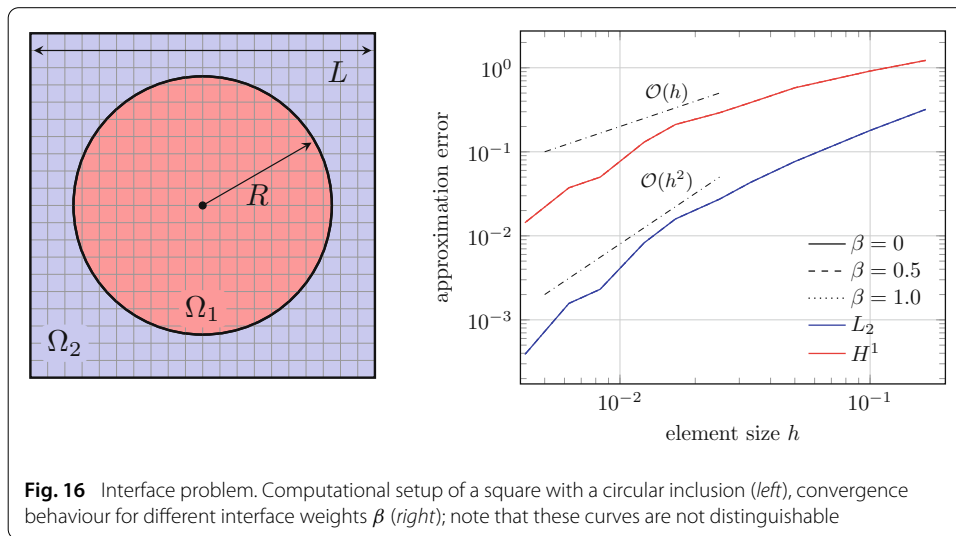




a fixed boundary location. Finally, the location of the boundary is varied again and the condition number for a Neumann problem is considered in the right of Fig. 15. Whereas in the non-stabilised case this value shows a very erratic behaviour with large peaks, the condition number for the stabilised method is almost constant at a low value.

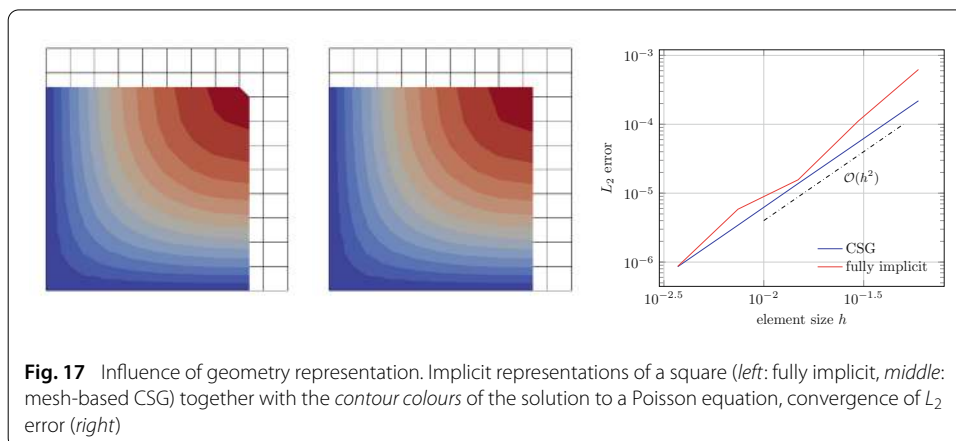
Now, an interface problem is considered. Figure 16 shows the computational domain that is composed of a circular domain  $\Omega_1$  embedded in a square domain  $\Omega_2$ . On this





domain, the Poisson problem  $-\alpha_i \Delta u = 4$  with material parameters  $\alpha_1 = 1$  and  $\alpha_2 = 1000$  is solved, subject to Dirichlet boundary conditions on the outer boundary  $\partial\Omega_2$ . The geometric parameters are chosen as  $R = 0.75$  and  $L = 2$ , respectively. This problem together with its analytic solution is taken from [2]. In the right graph of Fig. 16 the convergence behaviour is shown for different values of the interface weight factor  $\beta$ . For the three considered values  $\beta = 0, 0.5$  and  $1$ , the curves are indistinguishable. Also optimal convergence rates are achieved for mesh sizes smaller than  $h \approx 0.02$ .

At last, we consider the influence of the geometry representation. As outlined in “Constructive solid geometry modelling” section, we have two approaches available: the use of a signed distance function representing the entire embedded surface and the successive embedding of the geometry primitives that form the final model. For simplicity, consider a square that coincides on two of its edges with the mesh boundary whereas the other two are represented implicitly. Figure 17 shows the effect of the introduced two approaches in the left and middle images, respectively. Clearly, the upper right corner is chamfered off in the first approach, but represented exactly in the second. As a numerical problem we have chosen  $-\Delta u = 1$  on a unit square subject to  $u = 0$  on the lower and left boundaries and  $\partial u / \partial \mathbf{n} = 0$  on the other two boundaries. An analytic solution to this problem is



available, for instance in [45] in the context of Poiseuille flow in a rectangular channel. The right graph in Fig. 17 shows the convergence rates for the two types of geometry modelling. Clearly, optimal convergence rates are obtained for both cases. Nevertheless the exact representation of the corner leads to a much smoother outcome with lower approximation errors for coarse mesh sizes.

**Mesh-embedded CSG**

Here the domain as obtained by the CSG process of Fig. 6 is reconsidered, see also the left of Fig. 18. The embedding domain is  $\Omega_{\square} = (0, 1)^3$  and equipped with a uniform hexahedral mesh. Following the mesh-based Boolean operations as introduced in “Constructive solid geometry modelling” section, the immersed geometry is obtained by

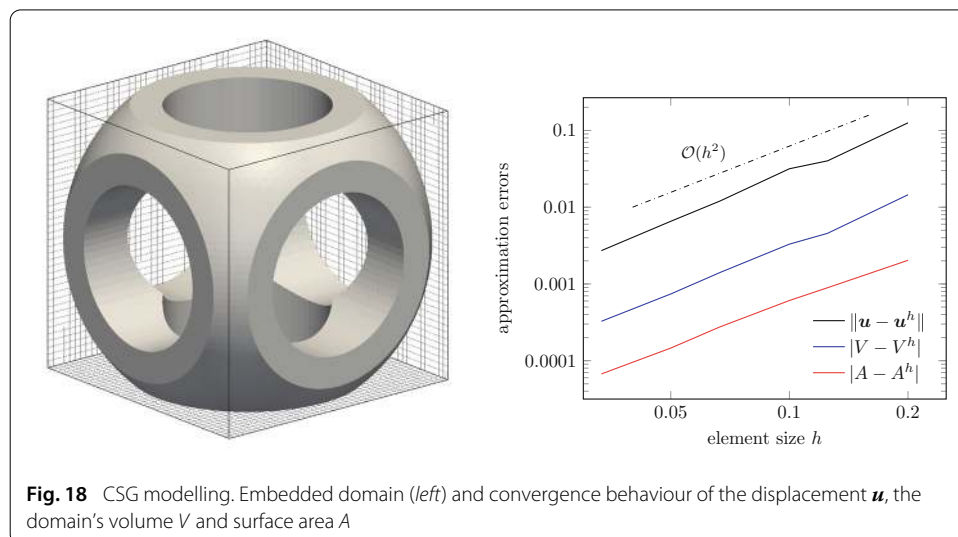
- 1 *intersection* with a sphere of radius 0.65 and centred at (0.5, 0.5, 0.5), and
- 2 successive *subtraction* of cylinders around the same centre with radius 0.3 and in the directions of the  $x_i$ -coordinate axes.

Thus the domain  $\Omega$  is obtained as shown in Fig. 18 and we assume that it is occupied by a hyperelastic solid. In a first analysis, linearised elasticity is assumed and the convergence is studied by using fundamental solution of elasticity  $\mathbf{U}(\mathbf{x}, \mathbf{y})$  (see, for instance, [46]) as an imposed analytic solution with a source point  $\mathbf{y}$  located outside of the domain. Therefore, on the bottom ( $x_3 = 0$ ) the boundary displacements  $\bar{\mathbf{u}}(\mathbf{x}) = \mathbf{U}(\mathbf{x}, \mathbf{y})$  are prescribed and the remaining boundaries are subject to the Neumann condition  $\bar{\mathbf{t}} = \mathbf{t}(\mathbf{U})(\mathbf{x}, \mathbf{y})$ . For simplicity, the material parameters are chosen as  $\lambda = 28.85$  and  $\mu = 19.23$ . The right of Fig. 18 shows the convergence of the displacement solution and of the computed volume and surface area of the embedded domain. Quadratic convergence is observed for all considered quantities.

Next, a compressible Neo-Hookean material model [22,23] with large deformations is used, based on the strain energy density

$$W(\mathbf{F}) = \frac{\lambda}{2}(\log J)^2 - \mu \log J + \frac{\mu}{2}(\text{tr } \mathbf{C} - 3), \quad J = \det \mathbf{F} \quad \text{and} \quad \mathbf{C} = \mathbf{F}^T \mathbf{F}, \quad (38)$$

with the deformation gradient  $\mathbf{F} = \mathbf{I} + \text{Grad } \mathbf{u}$ . The material parameters are the same as in the linearised case above. For the example, the bottom boundary is held fixed and a



**Fig. 18** CSG modelling. Embedded domain (left) and convergence behaviour of the displacement  $\mathbf{u}$ , the domain’s volume  $V$  and surface area  $A$

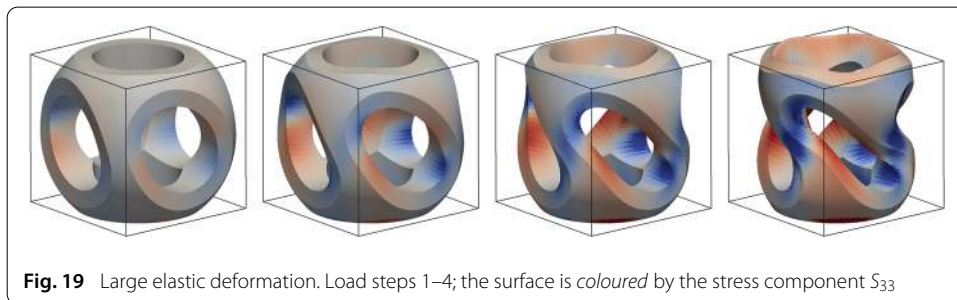
twisting traction field is applied to the top surface with value  $\vec{t} = 10(x_2 - 0.5, 0.5 - x_1, 0)$ . The load is applied in 4 steps and within each step a Newton method is used to obtain the equilibrium state. The deformed geometry for these four load steps is shown in the images of Fig. 19 for a  $40^3$  grid of linear hexahedron elements.

**Composite material**

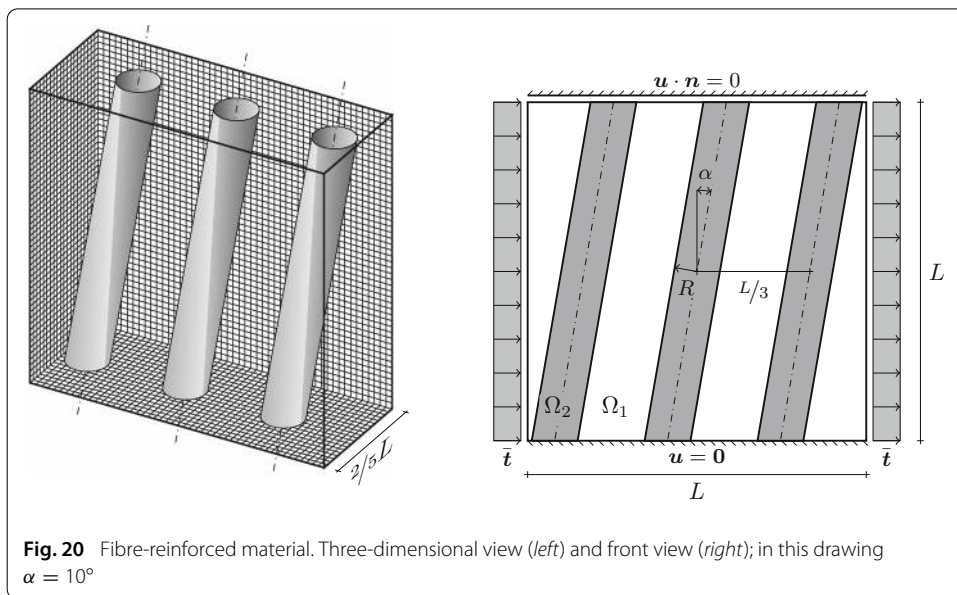
As a last example, the elastic deformation of a fibre-reinforced block of elastic material is considered. A block of dimension  $L \times \frac{2}{5}L \times L$  is reinforced by inclined fibres placed with a main axis separation of  $\frac{L}{3}$ . The fibres are represented by cylinders with radius  $\frac{L}{15}$ . The three-dimensional setup is shown in the left of Fig. 20 and on the right a two-dimensional view of the problem is depicted. The bottom surface is held fixed and the top surface is constrained in normal direction. The left and right surfaces are subject to a constant traction field  $\vec{t}$  in normal direction. The discretisation is carried out by a fixed mesh of dimension  $50 \times 20 \times 50$  as indicated on the back faces of the three-dimensional view. For comparison, we monitor the average horizontal displacement

$$U_1 = \frac{1}{|\Omega|} \int_{\Omega} u_1(x) d\Omega \tag{39}$$

throughout the composite body for a variety of fibre angles  $-35^\circ \leq \alpha \leq 35^\circ$ . Both domains  $\Omega_1$  and  $\Omega_2$  have the hyperelastic material law according to the energy (38) and

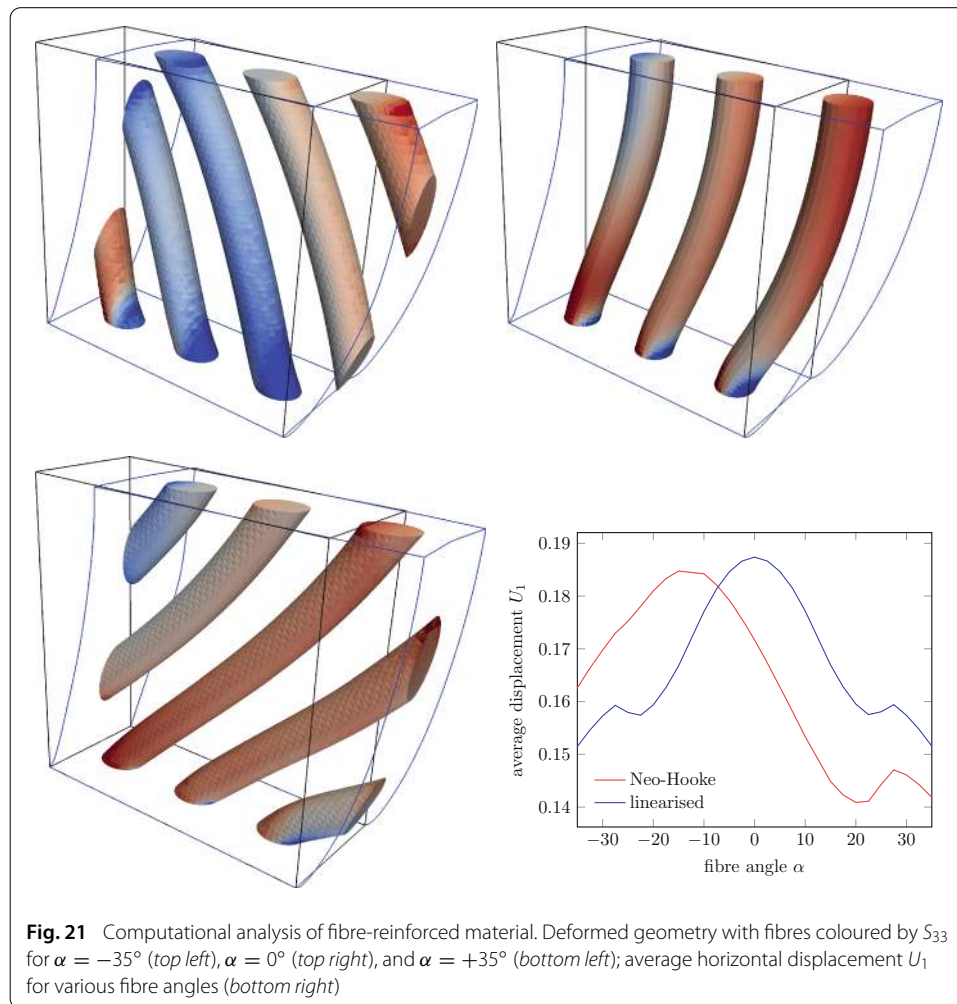


**Fig. 19** Large elastic deformation. Load steps 1–4; the surface is coloured by the stress component  $S_{33}$



**Fig. 20** Fibre-reinforced material. Three-dimensional view (left) and front view (right); in this drawing  $\alpha = 10^\circ$





the computations are carried out with large deformations. For comparison, a linearised situation is also considered.

The model parameters are chosen as  $L = 1$  and  $\vec{t} = (1, 0, 0)$ . The materials are represented by the Lamé parameters  $\lambda_1 = 5.769$ ,  $\mu_1 = 3.846$ ,  $\lambda_2 = 10\lambda_1$  and  $\mu_2 = 10\mu_1$ . Figure 21 shows the deformed configuration for fibre angles  $\alpha = -35^\circ$ ,  $\alpha = 0^\circ$  and  $\alpha = +35^\circ$ . In addition, the analysis of the average horizontal displacement  $U_1$  as a function of the considered fibre angles  $\alpha$  is shown for the Neo-Hooke material and linearised elasticity. Although there are similarities between the large-deformation analysis and the linearised version, striking differences can be observed too. Most of all, the linear variant is completely symmetric with respect to the sign of  $\alpha$  and has its largest value for  $\alpha = 0^\circ$ . In the large-deformation variant, on the other hand the result is a larger deformation for the negative fibre angles and smaller for positive angles. Overall, the body behaves less flexibly in the nonlinear analysis, but with a strong bias to an increased flexibility for negative fibre angles.

## Conclusions

Immersed finite element methods, that do not rely on a body-fitted mesh, are a promising alternative to conventional FEM for many applications. Especially in the case of complex three-dimensional geometries, moving interfaces, or design optimisation such methods

allow for more flexible geometry processing and remove the repeated interaction with mesh generation software. Here, we present an immersed FEM for the problem class of nonlinear elasticity, based on a weak incorporation of Dirichlet boundary conditions and interface conditions with Nitsche's method, an implicit geometry representation and accurate integration of the arising cut elements. We place emphasis on the implementation details such as the robust computation of the signed distance function and quadrature by means of tessellation. A common pitfall of non-body-fitted FEM, the loss of numerical stability in situations with degenerate function support, is analysed and we provide a stabilisation technique that is robust without affecting the convergence behaviour. Moreover, the choice of the parameters in the context of Nitsche's method are thoroughly discussed.

We demonstrate a way to incorporate sharp features such as edges and vertices in our method by means of successively embedding the geometry primitives into the analysis mesh in a similar way as in constructive solid geometry modelling. Based on this idea, geometry modelling is directly integrated in the finite element analysis and there is no need for a mesh generation tool. The presented applications emphasise the potential of this approach, where large deformation analyses are carried out based on a trivial Cartesian background mesh.

A present shortcoming of the introduced approach is the restriction to linear approximation orders. Although the field approximation used in this FEM can be of arbitrary order, a gain in convergence order would be impeded by the geometry representation based on linear facets. In principle, the use of more accurate signed distance functions and the subsequent adaptation on the quadrature level to account for embedded higher-order surface representations is feasible.

Finally, we note that the presented method is ideally suited for the incorporation of  $h$ -adaptivity. A combination of this immersed FEM with hierarchical refinement techniques as shown, for instance, in [47] would render a powerful analysis toolbox, which yields accurate numerical predictions based only on the input of geometry primitives.

#### Authors' contributions

The presented ideas are joint contribution of the three authors. TR implemented the numerical techniques and wrote the initial draft of the paper. JMGA and FC contributed to the revision of the draft. All authors read and approved the final manuscript.

#### Author details

<sup>1</sup>Multiscale in Mechanical and Biological Engineering (M2BE), University of Zaragoza, María de Luna 3, 50018 Zaragoza, Spain. <sup>2</sup>Institut für Baumechanik, TU Graz, Technikerstraße 4, 8010 Graz, Austria, <sup>3</sup>Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK.

#### Acknowledgements

This work was partially supported by the EPSRC (second author, Grant #EP/G008531/1), by the European Research Council (third author, Grant #ERC-2012-StG 306751), and by the Spanish Ministry of Economy and Competitiveness (third author, Grant #DPI2015-64221-C2-1-R).

#### Competing interests

The authors declare that they have no competing interests.

## Appendix A: method parameters $\gamma$ and $\beta$

In the following, estimates for the parameters  $\gamma$  (Penalty weighting parameter first appearing in (8)) and  $\beta$  (interface weighting parameter first appearing in (13)) are derived. These parameters only make sense with the Finite Element discretisation as the final goal. With a slight abuse of notation, the functions used in the following have to be understood as discrete FE solutions, but for simplicity the superscripts, for instance of  $\mathbf{u}^h$ , have been omitted.



**Dirichlet boundary conditions**

We develop an estimate for the parameter  $\gamma$  appearing in the linearised weighted residual equation (9). The material behaviour is assumed to be such that  $Da(\mathbf{u}, \mathbf{v})[\Delta \mathbf{u}]$  is an elliptic bilinear form. The aim is to show that  $A(\mathbf{u}; \mathbf{w}, \mathbf{w}) > 0$  for any  $\mathbf{w} \neq \mathbf{0}$ , where  $\mathbf{u}$  is the current displacement solution. This gives

$$\begin{aligned} A(\mathbf{u}; \mathbf{w}, \mathbf{w}) &= Da(\mathbf{u}, \mathbf{w})[\mathbf{w}] - 2 \int_{\Gamma_D} \mathbf{Dt}(\mathbf{u})[\mathbf{w}] \cdot \mathbf{w} d\Gamma + \gamma \int_{\Gamma_D} \mathbf{w} \cdot \mathbf{w} d\Gamma \\ &\geq Da(\mathbf{u}, \mathbf{w})[\mathbf{w}] - 2 \|\mathbf{Dt}(\mathbf{u})[\mathbf{w}]\|_{\Gamma_D} \|\mathbf{w}\|_{\Gamma_D} + \gamma \|\mathbf{w}\|_{\Gamma_D}^2 \\ &\geq Da(\mathbf{u}, \mathbf{w})[\mathbf{w}] - 2C \sqrt{Da(\mathbf{u}, \mathbf{w})[\mathbf{w}]} \|\mathbf{w}\|_{\Gamma_D} + \gamma \|\mathbf{w}\|_{\Gamma_D}^2 \\ &= \left( \sqrt{Da(\mathbf{u}, \mathbf{w})[\mathbf{w}]} - C \|\mathbf{w}\|_{\Gamma_D} \right)^2 + (\gamma - C^2) \|\mathbf{w}\|_{\Gamma_D}^2. \end{aligned} \tag{40}$$

Here,  $\|\mathbf{w}\|_{\Gamma_D}$  is the  $L_2$ -norm over the Dirichlet boundary  $\Gamma_D$ . The Cauchy-Schwarz inequality has been used from the first to the second line and, most importantly, the third line is based on the inverse inequality

$$C^2 Da(\mathbf{u}, \mathbf{w})[\mathbf{w}] \geq \|\mathbf{Dt}(\mathbf{u})[\mathbf{w}]\|_{\Gamma_D}^2. \tag{41}$$

This type of estimate is also presented in [3] for the case of Poisson’s equation. Knowledge of the constant  $C$  gives rise to the choice  $\gamma > C^2$  which renders the last line in (40) positive. Inserting the finite element trial functions (20) into (41) leads to the condition

$$C^2 Da(\mathbf{u}, \mathbf{e}_a \varphi_I)[\mathbf{e}_b \varphi_J] \geq \int_{\Gamma_D} \mathbf{Dt}(\mathbf{u})[\mathbf{e}_a \varphi_I] \cdot \mathbf{Dt}(\mathbf{u})[\mathbf{e}_b \varphi_J] d\Gamma \tag{42}$$

for all coordinate directions  $\mathbf{e}_a$ ,  $0 \leq a < n_d$ , and all shape functions  $\varphi_I$ . Obviously, this condition is only non-trivial if the supports of  $\varphi_I$  and  $\varphi_J$  overlap each other and intersect with the Dirichlet boundary  $\Gamma_D$ . Therefore, we use the following abbreviations for these domain and boundary intersections

$$\Omega_{IJ} = \Omega \cap (\text{supp}(\varphi_I) \cap \text{supp}(\varphi_J)) \quad \text{and} \quad \Gamma_{IJ} = \Gamma_D \cap (\text{supp}(\varphi_I) \cap \text{supp}(\varphi_J)). \tag{43}$$

The left-hand side of (42) can be re-written as

$$Da(\mathbf{u}, \mathbf{e}_a \varphi_I)[\mathbf{e}_b \varphi_J] = \int_{\Omega_{IJ}} \mathbf{Grad}(\mathbf{e}_b \varphi_J) : \hat{\mathbf{C}}(\mathbf{u}) : \mathbf{Grad}(\mathbf{e}_a \varphi_I) d\Omega, \tag{44}$$

and the right-hand side becomes

$$\begin{aligned} &\int_{\Gamma_{IJ}} [(\hat{\mathbf{C}}(\mathbf{u}) : \mathbf{Grad}(\mathbf{e}_a \varphi_I)) \mathbf{n}] \cdot [(\hat{\mathbf{C}}(\mathbf{u}) : \mathbf{Grad}(\mathbf{e}_b \varphi_J)) \mathbf{n}] d\Gamma \\ &= \int_{\Gamma_{IJ}} (\hat{\mathbf{C}}(\mathbf{u}) : \mathbf{Grad}(\mathbf{e}_a \varphi_I)) (\mathbf{n} \otimes \mathbf{n}) : (\hat{\mathbf{C}}(\mathbf{u}) : \mathbf{Grad}(\mathbf{e}_b \varphi_J))^\top d\Gamma \\ &\leq \int_{\Gamma_{IJ}} (\hat{\mathbf{C}}(\mathbf{u}) : \mathbf{Grad}(\mathbf{e}_a \varphi_I)) : (\hat{\mathbf{C}}(\mathbf{u}) : \mathbf{Grad}(\mathbf{e}_b \varphi_J)) d\Gamma. \end{aligned} \tag{45}$$

Abbreviating the integrands in (44) and (45) with  $f_{abI}^\Omega$  and  $f_{abJ}^\Gamma$ , we obtain for the inverse inequality

$$C^2 \int_{\Omega_{IJ}} f_{abI}^\Omega d\Omega \geq \int_{\Gamma_{IJ}} f_{abJ}^\Gamma d\Gamma. \tag{46}$$

Assuming continuity of the integrands, the mean value theorem of integration states that two points  $\xi \in \Omega_{IJ}$  and  $\eta \in \Gamma_{IJ}$  exist such that the estimate becomes

$$C^2 \geq \frac{f_{abJ}^\Gamma(\eta) |\Gamma_{IJ}|}{f_{abI}^\Omega(\xi) |\Omega_{IJ}|}. \tag{47}$$

Without assumptions on the shape functions  $\varphi_I$  and the material behaviour, we cannot further reduce the ratio of the evaluated integrands. Nevertheless, if  $\alpha$  denotes a characteristic material behaviour (for instance the bulk modulus), it is clear from (44) and (45) that this ratio scales like  $\frac{\alpha^2}{\alpha} = \alpha$ . Here, we focus on controlling the second part of the right hand side of the estimate (47). Using a standard finite element basis, there are constellations in which  $\Omega_{IJ} \rightarrow 0$ , whereas  $\Gamma_{IJ}$  does not decrease: see, for instance the *sliver* test case in [16].

Employing the proposed stabilisation technique from “Stabilisation” section, where critical shape functions are essentially joined with neighbouring ones that have a non-degenerate support, the above geometric ratio can be safely estimated as

$$\frac{|\Gamma_{IJ}|}{|\Omega_{IJ}|} \approx \frac{h^{n_d-1}}{h^{n_d}} = \frac{1}{h}, \tag{48}$$

because the support sizes of the stabilised basis functions  $\tilde{\varphi}_I$  never fall below the stabilisation threshold  $\hat{s}$  which is of order  $h^{n_d}$ . In conclusion, we can state that there exists a constant  $\gamma_0$  independent of the material and the mesh size  $h$ , such that

$$\gamma = \gamma_0 \frac{\alpha}{h} > C^2 \tag{49}$$

and, in turn,  $A(\mathbf{u}; \mathbf{w}, \mathbf{w}) > 0$ , see estimate (40).

**Interface conditions**

Next, we assess the choice of parameters in the linearised weighted residual equation (19) for the interface problems. To this end, the steps of (40) are repeated in an analogous manner with the function  $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$  and yield the estimate

$$\begin{aligned} A(\mathbf{u}; \mathbf{w}, \mathbf{w}) &\geq \left( \sqrt{Da_1(\mathbf{u}_1, \mathbf{w}_1)[\mathbf{w}_1] + Da_2(\mathbf{u}_2, \mathbf{w}_2)[\mathbf{w}_2]} - D\|\llbracket \mathbf{w} \rrbracket\|_{\Gamma} \right)^2 + (\gamma - D^2)\|\llbracket \mathbf{w} \rrbracket\|_{\Gamma}^2, \end{aligned} \tag{50}$$

which is positive for  $\mathbf{w}_i \neq \mathbf{0}$  and  $\gamma > D^2$ . Here,  $D$  is the constant of the inverse estimate

$$\begin{aligned} D^2 (Da_1(\mathbf{u}_1, \mathbf{w}_1)[\mathbf{w}_1] + Da_2(\mathbf{u}_2, \mathbf{w}_2)[\mathbf{w}_2]) &\geq \int_{\Gamma} \{ \beta D \mathbf{t}(\mathbf{u}_1)[\mathbf{w}_1] + (1 - \beta) D \mathbf{t}(\mathbf{u}_2)[\mathbf{w}_2] \}^2 d\Gamma. \end{aligned} \tag{51}$$

Using the “Peter-Paul inequality” with some  $\delta > 0$  (that is, the estimate  $2ab \leq \delta a^2 + \delta^{-1}b^2$ ) one gets

$$\begin{aligned} D^2 (Da_1(\mathbf{u}_1, \mathbf{w}_1)[\mathbf{w}_1] + Da_2(\mathbf{u}_2, \mathbf{w}_2)[\mathbf{w}_2]) &\geq (1 + \delta)\beta^2 \int_{\Gamma} (D \mathbf{t}(\mathbf{u}_1)[\mathbf{w}_1])^2 d\Gamma + (1 + \delta^{-1})(1 - \beta)^2 \int_{\Gamma} (D \mathbf{t}(\mathbf{u}_2)[\mathbf{w}_2])^2 d\Gamma. \end{aligned} \tag{52}$$

Let  $C_i$  denote the constant of (41) for the subdomain  $\Omega_i$ , choose  $\delta = ((1 - \beta)^2 C_2^2) / (\beta^2 C_1^2)$ , and insert the inverse inequality (41) into the last expression in order to get

$$D^2 \geq \beta^2 C_1^2 + (1 - \beta)^2 C_2^2. \tag{53}$$

Based on the discussion above we know that the stabilisation technique given in “Stabilisation” section keeps the values of  $C_i$  bounded: therefore  $D$  according to (53) is well-behaved, independent of the choice of  $0 \leq \beta \leq 1$ . Nevertheless, in [43] a parameter choice for  $\beta$  is presented which keeps the value of  $D$  bounded even for a non-stabilised finite element basis. Their choice is, adapted to the notation used here,  $\beta = C_1^{-1} / (C_1^{-1} + C_2^{-1})$  and this

value controls nicely the right-hand-side of expression (53) for unbounded values of  $C_i$  (note that in case of geometric constellations for which  $C_1$  is large the counterpart  $C_2$  is small, and vice versa). But it has to be remarked that in certain types of applications (e.g. fluid-structure interaction [48]) it is convenient to freely choose the parameter  $\beta$  without stability restrictions. Moreover, this special choice of  $\beta$  maintains  $A(\boldsymbol{u}; \boldsymbol{w}, \boldsymbol{w}) > 0$  for a finite value of  $\gamma$ , but does not prevent unbounded values of the condition number of the system matrix. With the here presented stabilisation technique, the choice of  $\beta$  does not affect stability and  $\gamma = \gamma_0(\alpha_1 + \alpha_2)/h$  provides a safe choice for the characteristic material parameters  $\alpha_i$ .

Received: 29 March 2016 / Accepted: 7 July 2016

Published online: 22 July 2016

## References

1. Cottrell JA, Hughes TJ, Bazilevs Y. *Isogeometric analysis: toward integration of CAD and FEA*. New York: Wiley; 2009.
2. Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsche's method, for elliptic interface problem. *Comput Methods Appl Mech Eng*. 2002;191:5537–52.
3. Dolbow J, Harari I. An efficient finite element method for embedded interface problems. *Int J Numer Methods Eng*. 2009;78:229–52.
4. Sanches R, Bornemann P, Cirak F. Immersed b-spline (i-spline) finite element method for geometrically complex domains. *Comput Methods Appl Mech Eng*. 2011;200(13):1432–45.
5. Rüberg T, Cirak F. An immersed finite element method with integral equation correction. *Int J Numer Methods Eng*. 2011;86:93–114.
6. Rüberg T, Cirak F. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Comput Methods Appl Mech Eng*. 2011;209–212:266–83.
7. Nitsche J. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. In: *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, vol. 36. Berlin: Springer; 1971. p. 9–15.
8. Ern A, Guermond JL. *Theory and practice of finite elements*, vol. 159. Berlin: Springer; 2004.
9. Courant R. Variational methods for the solution of problems of equilibrium and vibrations. *Bull Am Math Soc*. 1943;49(1):1–23.
10. Babuška I. The finite element method with Lagrangian multipliers. *Numerische Mathematik*. 1973;20(3):179–92.
11. Fries T-P, Belytschko T. The extended/generalized finite element method: an overview of the method and its applications. *Int J Numer Methods Eng*. 2010;84(3):253–304.
12. Höllig K, Reif U, Wipperfurth J. Weighted extended B-spline approximation of Dirichlet problems. *SIAM J Numer Anal*. 2001;39:442–62.
13. Legrain G, Allais R, Cartraud P. On the use of the extended finite element method with quadtree/octree meshes. *Int J Numer Methods Eng*. 2011;86(6):717–43.
14. Schillinger D, Dedè L, Scott MA, Evans JA, Borden MJ, Rank E, Hughes TJR. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of nurbs, immersed boundary methods, and t-spline CAD surfaces. *Comput Methods Appl Mech Eng*. 2012;249–252:116–50.
15. Höllig K, Reif U. Nonuniform web-splines. *Comput Aided Geom Des*. 2003;20(5):277–94.
16. Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Appl Numer Math*. 2012;62(4):328–41.
17. 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.
18. Rüberg T, Aznar JMG. Numerical simulation of solid deformation driven by creeping flow using an immersed finite element method. *Adv Model Simul Eng Sci*. 2016;3(1):1–31.
19. Bandara K, Rüberg T, Cirak F. Shape optimisation with multiresolution subdivision surfaces and immersed finite elements. *Comput Methods Appl Mech Eng*. 2016;300:510–39.
20. Hansbo A, Hansbo P. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Comput Methods Appl Mech Eng*. 2004;193(33):3523–40.
21. Mergheim J, Steinmann P. A geometrically nonlinear FE approach for the simulation of strong and weak discontinuities. *Comput Methods Appl Mech Eng*. 2006;195(37):5037–52.
22. Holzapfel GA. *Nonlinear solid mechanics: a continuum approach for engineering*. New York: Wiley; 2000.
23. Wriggers P. *Nonlinear finite element methods*. Berlin: Springer; 2008.
24. Ortiz M, Pandolfi A. Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *Int J Numer Methods Eng*. 1999;44(9):1267–82.
25. Höllig K, Reif U, Wipperfurth J. B-spline approximation of Neumann problems. Technical report. Germany: University of Stuttgart; 2001.
26. Jones M, Baerentzen JA, Šrámek M. 3D distance fields: a survey of techniques and applications. *IEEE Trans Vis Comput Graph*. 2006;12(4):581–99.
27. Piegl L, Tiller W. *The NURBS book*. 1997. Monographs in visual communication. Berlin: Springer; 1997.
28. Jörg P, Reif U. *Subdivision Surfaces. Geometry and Computing*, vol. 3. Berlin: Springer; 2008. doi:10.1007/978-3-540-76406-9.

29. Cirak F, Ortiz M, Schröder P. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Int J Numer Methods Eng.* 2000;47(12):2039–72.
30. Pabst HF, Springer JP, Schollmeyer A, Lenhardt R, Lessig C, Froehlich R. Ray casting of trimmed NURBS surfaces on the GPU. In: Knoll A, Wald I, Parker S, Hansen C, editors. *IEEE Symposium on interactive ray tracing.* Piscataway: IEEE; 2006. p. 151–60.
31. Shapiro V. Real functions for representation of rigid solids. *Comput Aided Geom Des.* 1994;11(2):153–75.
32. Ericson C. *Real-time collision detection.* Boca Raton: CRC Press; 2004.
33. Mauch SP. *Efficient algorithms for solving static Hamilton-Jacobi equations.* PhD thesis. Pasadena: California Institute of Technology; 2003.
34. Rank E, Ruess M, Kollmannsberger S, Schillinger D, Düster A. Geometric modeling, isogeometric analysis and the finite cell method. *Comput Methods Appl Mech Eng.* 2012;249–252:104–15.
35. Pasko A, Adzhiev V, Sourin A, Savchenko V. Function representation in geometric modeling: concepts, implementation and applications. *Vis Comput.* 1995;11(8):429–46.
36. Duff T. Interval arithmetic recursive subdivision for implicit functions and constructive solid geometry. In: *ACM SIGGRAPH computer graphics*, vol. 26. New York: ACM; 1992. p. 131–38.
37. Sukumar N, Chopp DL, Moës N, Belytschko T. Modeling holes and inclusions by level sets in the extended finite-element method. *Comput Methods Appl Mech Eng.* 2001;190(46):6183–200.
38. Kobbelt LP, Botsch M, Schwanecke U, Seidel HP. Feature sensitive surface extraction from volume data. In: Pockock L, editor. *Proceedings of the 28th annual conference on computer graphics and interactive techniques.* New York: ACM; 2001. p. 57–66.
39. Song X, Jüttler B. Modeling and 3D object reconstruction by implicitly defined surfaces with sharp features. *Comput Graph.* 2009;33(3):321–30.
40. Müller B, Kummer F, Oberlack M. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *Int J Numer Methods Eng.* 2013;96(8):512–28.
41. Sudhakar Y, de Almeida JM, Wall WA. An accurate, robust, and easy-to-implement method for integration over arbitrary polyhedra: application to embedded interface methods. *J Comput Phys.* 2014;273:393–415.
42. Düster A, Parvizian J, Yang Z, Rank E. The finite cell method for three-dimensional problems of solid mechanics. *Comput Methods Appl Mech Eng.* 2008;197(45):3768–82.
43. Annavarapu C, Hautefeuille M, Dolbow JE. A robust Nitsche's formulation for interface problems. *Comput Methods Appl Mech Eng.* 2012;225:44–54.
44. Rangarajan R, Lew AJ. Universal meshes: a method for triangulating planar curved domains immersed in nonconforming meshes. *Int J Numer Methods Eng.* 2014;98(4):236–64.
45. White FM, Corfield I. *Viscous fluid flow*, vol. 3. New York: McGraw-Hill; 2006.
46. McLean WC. *Strongly elliptic systems and boundary integral equations.* Cambridge: Cambridge University Press; 2000.
47. Bornemann P, Cirak F. A subdivision-based implementation of the hierarchical b-spline finite element method. *Comput Methods Appl Mech Eng.* 2013;253:584–98.
48. Bazilevs Y, Hsu M-C, Scott M. Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Comput Methods Appl Mech Eng.* 2012;249:28–41.

**Submit your manuscript to a SpringerOpen<sup>®</sup> 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](http://springeropen.com)

---