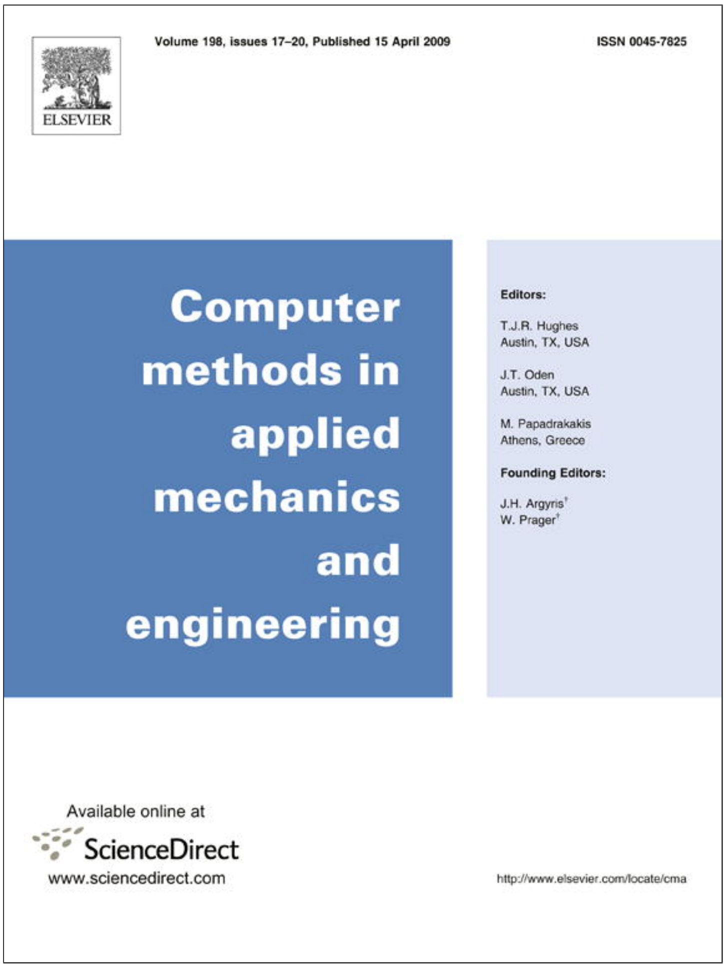

Volume 198, issues 17–20, Published 15 April 2009

ISSN 0045-7825

ELSEVIER

Computer methods in applied mechanics and engineering

Available online at
ScienceDirect
www.sciencedirect.com

http://www.elsevier.com/locate/cma

Comput. Methods Appl. Mech. Engrg. 198 (2009) 1513–1534

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

A discontinuous-Galerkin-based immersed boundary method with non-homogeneous boundary conditions and its application to elasticity

Ramsharan Rangarajan [a,1], Adrián Lew [a,*,2], Gustavo C. Buscaglia [b,3]

[a] *Department of Mechanical Engineering, Stanford University, Stanford, CA 94305-4040, USA*
[b] *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 13560-970 São Carlos, SP, Brasil*

ARTICLE INFO

Article history:
Received 22 August 2008
Received in revised form 21 December 2008
Accepted 8 January 2009
Available online 31 January 2009

Keywords:
Immersed boundary methods
Elasticity
Non-homogeneous boundary conditions
Discontinuous Galerkin

ABSTRACT

We propose a discontinuous-Galerkin-based immersed boundary method for elasticity problems. The resulting numerical scheme does not require boundary fitting meshes and avoids boundary locking by switching the elements intersected by the boundary to a discontinuous Galerkin approximation. Special emphasis is placed on the construction of a method that retains an optimal convergence rate in the presence of non-homogeneous essential and natural boundary conditions. The role of each one of the approximations introduced is illustrated by analyzing an analog problem in one spatial dimension. Finally, extensive two- and three-dimensional numerical experiments on linear and nonlinear elasticity problems verify that the proposed method leads to optimal convergence rates under combinations of essential and natural boundary conditions.

1. Introduction

This article is concerned with the creation of a discontinuous-Galerkin-based immersed boundary method for quasistatic elasticity problems. In doing so, we propose a strategy to impose non-homogeneous Dirichlet and Neumann boundary conditions while preserving optimal order of convergence. These ideas are not particular to elasticity, but rather to general second-order elliptic problems. We specify two such methods for piecewise linear elements in two and three spatial dimensions, and demonstrate the quadratic order of convergence of the resulting numerical scheme with numerical examples.

Immersed boundary methods refer here to numerical methods that construct an approximation to a boundary value problem on a background grid that need not fit the domain of the problem. By choosing as background grid that of a simple-to-mesh open set that contains the domain of the problem, the well-known difficulties in automatically meshing complex domains are sidestepped. On the other hand, the difficulties are now transferred to the design of strategies to impose boundary conditions.

Two key features that distinguish different IB methods are: (a) how the exact domain of the problem is approximated in the background grid and (b) how boundary conditions on the boundary of the approximate domain are imposed. Domains are nearly always approximated, and as a result, the boundary of the problem is only an approximation of the exact one. This naturally poses the question of how boundary conditions originally prescribed on the exact boundary are approximated and enforced on the approximate one.

In [32] a discontinuous-Galerkin-based immersed boundary method (DG-IBM) was proposed, dealing with the simpler case of homogeneous Dirichlet boundary conditions on the entire boundary, and formulated for the Poisson and the scalar reaction–diffusion equations. The approximate domain was constructed as the interior of the zero sublevel-set of the interpolant of the signed distance function to the boundary of the exact domain. The approximation of boundary conditions in this case was trivial: homogeneous Dirichlet boundary conditions were imposed on the approximate boundary. A two-dimensional implementation of the method with piecewise linear triangles was described. The method was numerically shown to attain second-order convergence rate in the L^2 norm.

Approximations of the domain and non-homogeneous boundary conditions of both Dirichlet and Neumann type are by no means problems specific to IB methods. In fact, these were issues addressed early on in the creation of classical finite element methods, see, e.g. [63–65,16,46,47,8–10,31,52]. For instance, when adopting a polygonal approximation of a curved domain, the rate of convergence is at best quadratic in the L^2 norm irrespective of the polynomial order of the basis functions, see [52,57,12]. Both

[*] Corresponding author.
E-mail addresses: rram@stanford.edu (R. Rangarajan), lewa@stanford.edu (A. Lew), gustavo.buscaglia@icmc.usp.br (G.C. Buscaglia).

[1] Supported by the Stanford Graduate Fellowship and Department of the Army Research Grant W911NF-07-2-0027.

[2] Supported by Department of the Army Research Grant W911NF-07-2-0027, the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54GM072970, an NSF-Career Award, NSF Grant No. CMMI-0747089, and an ONR Young Investigator Award, ONR Grant No. N000140810852.

[3] Supported by FAPESP (Brasil) and CNPq (Brasil).

0045-7825/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2009.01.018

approximations should be appropriately tailored to guarantee optimal order of convergence of the method. Isoparametric elements are perhaps the optimal result of these considerations, albeit they are not the only possibility. Following [31] in the construction of isoparametric elements, the approximate boundary interpolates the exact one at some nodes. The interpolant of the prescribed exact boundary values at these same nodes acts as the approximate boundary condition on the approximate boundary. In this way the error stemming from approximating the domain and that from approximating the boundary values are of the same order. This is why subparametric elements are hardly a convenient alternative.

Recently, there have been efforts to use functions inspired by CAD as basis functions to approximate solutions of partial differential equations. B-splines and NURBS have been used very successfully in CAD for accurate representations of complex geometries. This success is attributed not only to the special properties of these functions, but also to efficient evaluation algorithms credited to de Casteljau [19] and De Boor [18], among others. Apart from guaranteeing accurate domain representation, they have the added convenience of allowing for the construction of spaces with higher order continuity, for instance C^1 or C^2, instead of the typical C^0. Early methods that adopted these ideas are the R-functions method (RFM) [60,45] and the Weighted extended B-splines (Web-splines) method [25,26]. Both are closely related meshfree methods that use B-splines as basis functions. However, they do not exploit the properties of B-splines for domain representation; they use them solely for interpolation purposes. The information about the domain is implicitly conveyed via the boundary condition. Homogeneous essential boundary conditions are enforced on the exact boundary by multiplying the basis functions by a smooth enough function that vanishes only at the boundary, for instance an R-function [60,45]. Non-homogeneous essential boundary conditions require either the solution of a system of equations involving all unknowns along the boundary to determine the approximate boundary conditions [25], or they are simply approximated in a suboptimal way, resulting in a diminished convergence rate as observed in the presence of boundary locking [45].

The ideas in *isogeometric analysis* [27] constitute a new effort in a similar direction. In this context, NURBS are adopted both to accurately represent the boundary and to construct approximations of the solution. As with the other two spline-based methods, homogeneous essential boundary conditions are easily imposed, while additional care is needed for non-homogeneous ones. In [49], NURBS-based discretizations are adopted only near the boundary, to harness the accurate representation of the domain, while retaining traditional finite element approximations in the interior.

Applications of some immersed boundary methods to elasticity can be found in [11,7,48], among others. See also [4] for some related ideas. A qualitative discussion, by no means comprehensive, of different approaches to approximate the domain and impose homogeneous boundary conditions in immersed boundary methods was provided in [32]. A recent review can be found in [40]. Herein, we focus part of the discussion on the imposition of non-homogeneous boundary conditions. A broad class of IB methods rely on constructing an artificial set of forces on the exact Dirichlet boundary to impose the prescribed boundary conditions, e.g. [24,43,30,62]. Instead of defining an approximate boundary, the computed force is spread over several grid points by some regularization of the surface Dirac delta distribution commensurate with the mesh size. Non-homogeneous Neumann boundary conditions can be dealt with similarly; in this case the force is simply prescribed. As in penalty methods, a stiff system of equations may be obtained depending on how artificial forces are constructed.

A second class of IB methods imposes Dirichlet boundary conditions directly. If not carefully crafted, these approaches often lead to suboptimal convergence rates. Boundary locking may occur [32], which is a loss in the approximation properties near the Dirichlet boundary because of the constraints imposed by the strong boundary conditions. Locking can be avoided by imposing boundary conditions weakly through Lagrange multipliers (e.g. [5,41]), such as in the fictitious domain method [22], or by carefully selecting collocation points at which boundary values are enforced. Alternatively, Nitsche's method or any of its variants [17] can be adopted. They utilize a modified weak form of the problem to impose boundary conditions; we refer to [4] for a discussion. In [36] essential boundary conditions are imposed strongly and locking avoided by essentially regarding triangles or tetrahedra along the boundary as degenerated quadrilaterals or hexahedra. DG-IBM circumvents locking by introducing a discontinuous Galerkin discretization on those elements intersected by the boundary. Additionally, the lack of continuity across elements enables the choice of a convenient basis within each element intersected by the boundary. In [32] one such basis was proposed having two of the nodes located along the approximate boundary, instead of at the vertices of the triangle. Prescribed boundary values can be directly set therein, sidestepping the need to impose additional constraints, like, for instance, a ghost-cell method would do [59]. Approaches that utilize local remeshing to construct a finite element mesh that conforms to the boundary will of course not display any locking (e.g. [20]), but this task could become quite cumbersome in three-dimensions. Both local remeshing and the immersed boundary approaches may lead to poorly shaped elements or parts of it, with perhaps the latter being somewhat more advantageous.

Similar difficulties in imposing essential and natural boundary conditions have been faced by meshless methods, e.g. [4,6,21], or the natural element method [53]. The boundary is always immersed in these cases. Furthermore, it is often difficult to construct interpolants with the associated sets of basis functions. Boundary values have therefore been imposed as constraints. Lagrange multipliers [2,5,22] and penalty methods [3] are commonly used for this purpose, though they come with their own share of troubles. As for immersed boundary methods, selecting the right space of Lagrange multipliers is a delicate task; a very large space may lead to a degenerate system of equations, while the other extreme will result in poorly satisfied boundary conditions. When a penalty approach is adopted, it is necessary to deal with the well-know difficulty of finding a value of the penalty parameter that will impose the constraint without overly stiffening the system of equations. The use of Nitsche's method in this context is reported in [28,4,51]. Hybrid approaches in which standard finite elements are adopted near the boundary have also been proposed [29]. Finally, the characteristic function method described in [4, p. 111] is a very general idea that enables the construction of high-order approximations in immersed boundary and meshless methods, at the cost of modifying the approximation space in large parts of the domain. No standard and simple solution seems to be available to impose non-homogeneous boundary conditions in this case.

The contributions of this manuscript are:

(1) The formulation of DG-IBM for linear and nonlinear elasticity problems, demonstrating its performance with numerical examples.
(2) The formulation of a strategy to impose Dirichlet and Neumann boundary conditions in immersed boundary methods, and the specification of two alternative ways of doing it when piecewise linear discretizations over simplices are adopted.

The outline of the paper is as follows. Section 2 introduces a set of general conditions that, if satisfied by an immersed boundary method for a linear elliptic boundary value problem, will guarantee at least a linear convergence rate in the energy norm. These conditions are somewhat standard, albeit not yet stated in the context of immersed boundary methods. The first condition requires that the underlying finite element space be able to approximate the exact solution. However, it must do so by attaining some prescribed values at the Dirichlet boundary. This is where the construction of the approximate non-homogeneous essential boundary conditions discussed in this paper becomes important. The second condition is simply an asymptotic consistency one, standard for some discontinuous Galerkin methods and related to the analysis of the so-called variational crimes, see, e.g. [52]. Section 3 describes the elasticity problem, while the method is formulated in Section 4. In particular, Section 4.4 describes two alternative methods to impose non-homogeneous Neumann and Dirichlet boundary conditions, based on the strategy proposed in Section 2. The first method is somewhat simpler, but of a narrower scope, since it cannot be fully automatized. The second method, instead, is generally applicable. In Sections 4.4 and 4.5 we analyze the satisfaction of the two conditions for convergence of Section 2 in the case of a one-dimensional problem. Despite the fact that there are a number of important differences with higher-dimensional cases, we provide this analysis to illustrate how the two conditions in Section 2 play a role in the optimal convergence of the method. Being one-dimensional, the proofs are short and concise, and devoid of a number of technical details needed in higher-dimensional cases. Section 5 demonstrates the optimal convergence rate of the methods proposed here through two- and three-dimensional examples in linear elasticity, for which analytical expressions for their exact solutions are known. We pay special attention to different combinations indexed of types of boundary conditions, including problems in which the closure of the Dirichlet and Neumann boundaries may intersect. We conclude with two numerical examples in nonlinear elasticity aimed at demonstrating the ease of the method to handle complicated geometries. These are the compression of a femur, and the deformation of an image-based geometry.

2. Designing the method

Before we describe the method proposed in this article, we outline some conditions that will help in constructing it. These largely follow arguments made in [12] for the formulation of finite element methods with non-homogeneous boundary conditions in curved domains. As shown there, the detailed proof of convergence of a specific method is a rather delicate task. The goal herein, however, is to describe key conditions that should be kept in mind in the design of these methods. We therefore do this in a very general framework, which while valid for this task, does not fully expose the subtleties involved in understanding the precise way these methods converge.

We would like to numerically approximate the exact solution of a second-order elliptic boundary value problem on a sequence of easy-to-discretize domains, none of which might be the exact one, with optimal convergence rate. As we shall see next, there exists a wide spectrum of possible constructions to accomplish it, as long as some simple conditions are satisfied.

Let $(V, \|\cdot\|_V)$ be a Banach space of real-valued functions over an open bounded domain Ω. Consider the problem of finding a function $u \in V^{\partial} + \bar{u}$ such that

$$a(u, v) = F(v) \quad \forall v \in V^{\partial}, \tag{1}$$

where $V^{\partial} = \{v \in V : v|_{\Gamma^d} = 0\}$, $a : V \times V \mapsto \mathbb{R}$ is a bilinear form and $F : V \mapsto \mathbb{R}$ is a linear form. Here $\emptyset \neq \Gamma^d \subseteq \partial\Omega$, and $\bar{u} \in V$. We assume that a, F, Ω, Γ^d and $\bar{u}$ are such that this problem has a unique solution.

We construct next a numerical approximation to the solution of this problem. We consider first a sequence of domains $\{\Omega_h\}$, indexed by the mesh size h, such that as $h \searrow 0$, Ω_h is a better and better representation of Ω, in a sense to be discussed later. Additionally, we let $\mathscr{B}$ be a simple open and bounded domain, like a box or a ball, such that $\overline{\Omega \cup (\cup_h \Omega_h)} \subset \mathscr{B}$.

Since the exact solution and the approximations are defined on different domains, we shall assume that it is possible to extend the definition of u (smoothly enough) to $\mathscr{B}$. We shall denote its extension with u as well.

Let V_h be a finite dimensional space of functions with Ω_h as their domain. We assume that we can define a norm $|||\cdot|||_h$, possibly dependent on h, on

$$\mathrm{span}\{V_h, u\} = \left\{ v : \Omega_h \mapsto \mathbb{R} \mid v = v_h + \lambda u|_{\Omega_h},\ v_h \in V_h,\ \lambda \in \mathbb{R} \right\}.$$

Additionally, we assume that it is possible to define an *approximation* operator $\Pi_h : V^{\partial} + \bar{u} \mapsto V_h$ which satisfies that

(i) There exists $C > 0$ independent of h and that may depend on u such that

$$|||u - \Pi_h u|||_h < C(u)h, \tag{2}$$

where u is the exact solution of problem (1).

(ii) For all $v \in V^{\partial} + \bar{u}$ we have that

$$\Pi_h v|_{\Gamma_h^d} = \Pi_h \bar{u}|_{\Gamma_h^d}, \tag{3}$$

where $\Gamma_h^d \subseteq \partial\Omega_h$.

The discrete problem consists in finding $u_h \in V_h^{\partial} + \Pi_h \bar{u}$ such that

$$a_h(u_h, v_h) = F_h(v_h) \quad \forall v_h \in V_h^{\partial}, \tag{4}$$

where $V_h^{\partial} = \{v_h \in V_h : v_h|_{\Gamma_h^d} = 0\}$ is the space of admissible test functions. Here $F_h : V_h \mapsto \mathbb{R}$ is a linear functional, and $a_h : [\mathrm{span}\{V_h, u\}]^2 \mapsto \mathbb{R}$ is a coercive and continuous bilinear form, i.e., there exists $M, m > 0$ independent of h such that

$$|a_h(v, w)| \leqslant M|||v|||_h|||w|||_h \tag{5}$$

for all $v, w \in \mathrm{span}\{V_h, u\}$ and

$$m|||v_h|||_h^2 \leqslant a_h(v_h, v_h) \tag{6}$$

for all $v_h \in V_h^{\partial}$. Additionally, a_h and F_h should satisfy an *asymptotic consistency* condition

$$|a_h(u, v_h) - F_h(v_h)| \leqslant C(u)h|||v_h|||_h \quad \forall v_h \in V_h^{\partial} \tag{7}$$

for some $C(u) > 0$ independent of h that may depend on u, the exact solution of (1). We can now state the following lemma.

Lemma 2.1. *If conditions (2), (3) and (7) are satisfied, then*

$$|||u - u_h|||_h < C(u)h, \tag{8}$$

for some $C(u) > 0$ independent of h that may depend on u.

The proof is rather straightforward, and is shown next. However, we first discuss the hypotheses made.

The asymptotic consistency condition (7) expresses the relationship between the numerical method proposed and the problem to be solved. A similar asymptotic consistency condition is stated and proved in [12]. It also shares some features with those considered in [52] when analyzing errors arising from variational crimes, or those adopted in [35,47], among others. The asymptotic consistency condition in (7), however, is often encountered in the convergence analysis of the class of discontinuous Galerkin methods adopted herein, see, e.g. [33]. In fact, the proof of the above lemma follows essentially the same steps as that shown in [33]. Mesh-dependent norms as $|||\cdot|||_h$ are often encountered in these contexts.

Several key features of an immersed boundary method are embedded in (7). Its satisfaction often imposes restrictions on how Ω_h approximates Ω and Γ_h^d approximates Γ^d in terms of the mesh size h. Since natural boundary conditions are part of F_h, it also imposes restrictions on the way in which these are approximated. Notice that essential boundary condition on Γ_h^d are strictly imposed by $\Pi_h \bar{u}|_{\Gamma_h^d}$.

Condition (2) states that $\Pi_h u$ should approximate u at the appropriate rate. The key difference with a typical interpolation operator is that it should be able to do so by perhaps not interpolating u near the boundary, as condition (3) requests. This is necessary because the values of u on Γ_h^d are not generally known. If such an approximation operator can be constructed, it provides a natural way to impose essential boundary conditions. For example, if $\bar{u} = 0$, then it should be possible to approximate the solution of problem (1) with functions that are zero on Γ_h^d, as done in [32], even though u may not be zero there.

Not surprisingly, there is considerable freedom in the way the domain and essential and natural boundary conditions are approximated. For example, a method given by particular choices of a_h and F_h can be modified by constructing $\tilde{F}_h = F_h + L_h$, with any $L_h \in V_h^*$ such that $\|L_h\|_{V_h^*} < Ch$, for some $C > 0$ independent of h, and $\tilde{a}_h = a_h + b_h$, with any $b_h : [\text{span}\{V_h, u\}]^2 \mapsto \mathbb{R}$ such that $|b(u, v_h)| \leqslant C(u)h|||v_h|||_h$ for all $v_h \in V_h^\partial$, with $C(u)$ independent of h. The variety of possible approximation of Neumann boundary conditions are reflected in the multiplicity of choices for F_h. Similarly, the different options for the construction of approximation operators Π_h, or for the choice of $\bar{u}$, give rise to different schemes for the imposition of essential boundary conditions, all of them with the same order of convergence.

It is interesting to note that conditions (2) and (3) are precisely the ones that are not satisfied when boundary locking appears in the example discussed in [32]. In this case, V_h is the standard conforming finite element space of piecewise linear triangles, $V = H^1(\Omega)$, $u \in H^2(\Omega) \cap H_0^1(\Omega)$, $\bar{u} = 0$, Ω_h is such that the intersection of $\partial\Omega_h$ with the interior of any element is either empty or a straight segment, Π_h is the standard interpolation operator over piecewise linear triangles that do not intersect the approximate boundary that satisfies $\Pi_h u|_{\partial\Omega_h} = 0$, and $|||\cdot|||_h = \|\cdot\|_{H_0^1(\Omega_h)}$.

We conclude with two remarks. First, as we shall see next in the proof, we need only define the bilinear form of the numerical method on $\text{span}\{V_h, u\}$, although it is often possible to do it in a larger space. Second, lemma 2.1 compares functions by changing domains for each value of h; it says nothing about the convergence of the numerical approximations in Ω. This is apparently not an easy task, see, e.g. [12].

Proof of Lemma 2.1. We first bound the difference between $\Pi_h u$ and the numerical solution u_h. Note that because of (3), $(\Pi_h u - u_h) \in V_h^\partial$. We then have

$$\begin{aligned} m|||\Pi_h u - u_h|||_h^2 &\leqslant a_h(\Pi_h u - u_h, \Pi_h u - u_h) \\ &= a_h(\Pi_h u - u, \Pi_h u - u_h) \\ &\quad + a_h(u - u_h, \Pi_h u - u_h) \\ &\leqslant M|||\Pi_h u - u|||_h|||\Pi_h u - u_h|||_h \quad \text{by } (5) \\ &\quad + |a_h(u, \Pi_h u - u_h) - F_h(\Pi_h u - u_h)| \quad \text{by } (4) \\ &\leqslant C(u)h|||\Pi_h u - u_h|||_h \quad \text{by } (2),\ (7). \end{aligned}$$

It follows that:

$$|||\Pi_h u - u_h|||_h \leqslant C(u)h \tag{9}$$

and from here and (2) that

$$|||u - u_h|||_h \leqslant |||u - \Pi_h u|||_h + |||\Pi_h u - u_h|||_h \leqslant C(u)h, \tag{10}$$

which concludes the proof. □

The results in this section will next be used as design guidelines for the strategy to impose non-homogeneous essential and natural boundary conditions in DG-IBM. As mentioned, the task of proving that the proposed method satisfies all conditions in this section when restricted to a linear problem, and its convergence in Ω, will not be addressed in this article, but rather in the near future. Herein we shall only prove these conditions in the one-dimensional case, which due to its simplicity, illustrates how these ideas materialize in the proposed method. The optimal convergence rate of the method in two- and three-dimensional cases will be typified with extensive numerical examples.

Before formulating the method, we specify next the elasticity problem to which the method will be applied.

3. The nonlinear elasticity problem

We consider a simple elastic body with reference configuration $\Omega \subset \mathbb{R}^n$, an open and connected set. The nonlinear elasticity problem consists in finding the deformation mapping $\boldsymbol{\varphi} : \Omega \mapsto \mathbb{R}^n$ which is a stationary point of the potential energy functional

$$I[\boldsymbol{\varphi}] = \int_\Omega [W(\nabla\boldsymbol{\varphi}) - \mathbf{B}\cdot\boldsymbol{\varphi}]dV - \int_{\Gamma^\tau} \overline{\mathbf{T}}\cdot\boldsymbol{\varphi}\,dS \tag{11}$$

among all smooth enough deformation mappings that are equal to $\bar{\boldsymbol{\varphi}}$ on Γ^d. Here $W : \mathbb{R}^{n\times n} \mapsto \mathbb{R}$ is the strain energy density, Γ^τ and Γ^d are open subsets of $\Gamma = \partial\Omega$ such that $\Gamma^d \cap \Gamma^\tau = \emptyset$, $\mathbf{B} : \Omega \mapsto \mathbb{R}^n$ indicates the force per unit volume applied on the body, and $\bar{T} : \Gamma^\tau \mapsto \mathbb{R}^n$ specifies the external tractions imposed on a subset of its boundary.

The Euler–Lagrange equations corresponding to this variational principle are

$$\begin{aligned} &\text{DIV}[\mathbf{P}(\nabla\boldsymbol{\varphi}(\mathbf{X}))] + \mathbf{B} = 0 \quad \text{in } \Omega, \\ &\mathbf{P}(\nabla\boldsymbol{\varphi}(\mathbf{X}))\cdot\mathbf{N} = \overline{\mathbf{T}} \quad \text{on } \Gamma^\tau, \end{aligned}$$

where $\mathbf{N}$ is the unit outward normal to Γ^τ, and

$$\mathbf{P}(\mathbf{F}) = \frac{\partial W}{\partial \mathbf{F}}(\mathbf{F})$$

denotes the first Piola–Kirchhoff stress tensor.

The existence and regularity of solutions to this problem depend crucially on the properties of W, among others. We refer the reader to [37,15] for a more comprehensive discussion. For the discussions herein we shall assume that solutions are smooth enough so that the approximation conditions (2) and (3) can be satisfied.

We will consider examples with two different strain energy densities. The first one is that of a neo-Hookean material extended to the compressible range, namely,

$$W(\mathbf{F}) = \frac{\lambda}{2}\log(\det(\mathbf{F}))^2 - \mu\log(\det(\mathbf{F})) + \frac{\mu}{2}\text{tr}(\mathbf{F}^\mathrm{T}\mathbf{F} - \mathbf{I}) \tag{12}$$

if $\det(\mathbf{F}) > 0$, and $W(\mathbf{F}) = +\infty$ otherwise. Here $\lambda > 0$ and $\mu > 0$ are material parameters, and $\mathbf{I}$ is the identity $n \times n$ tensor. The stress tensor follows as:

$$\mathbf{P}(\mathbf{F}) = \lambda\log(\det(\mathbf{F}))\mathbf{F}^{-\mathrm{T}} + \mu(\mathbf{F} - \mathbf{F}^{-\mathrm{T}}). \tag{13}$$

The second strain energy density is that of a linear elastic material, namely,

$$W(\mathbf{F}) = \frac{1}{2}(\mathbf{F} - \mathbf{I}) : \mathbb{A} : (\mathbf{F} - \mathbf{I}), \tag{14}$$

where $\mathbb{A}$ is a fourth order tensor of elastic moduli. In particular, for an isotropic material we have, in indicial notation,

$$\mathbb{A}_{ijkl} = \lambda\,\delta_{ij}\delta_{kl} + \mu\big(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\big). \tag{15}$$

The stress tensor follows as:

$$\mathbf{P}(\mathbf{F}) = \mathbb{A} : (\mathbf{F} - \mathbf{I}). \tag{16}$$

4. Formulation of the method

In this section we describe the formulation of the method when piecewise linear elements in two and three dimensions are adopted. Most ideas described here, however, are directly applicable to other types of discretizations, such as quads or hexahedra. We review first the basic features of DG-IBM already introduced in [32], and then describe two methods to impose non-homogeneous boundary conditions.

4.1. Meshes

Let Ω be an open domain in $\mathbb{R}^n$, $n = 2, 3$ with, for simplicity, a smooth boundary $\Gamma = \partial\Omega$ (e.g. C^2). We assume that $\mathscr{B} \supseteq \Omega$, is an open domain, conveniently chosen such that it is trivial to construct a family of quasiuniform (see [13]) conforming meshes $\mathscr{T}_h$ of n-simplices over it. The parameter h is the mesh size. An element E in $\mathscr{T}_h$ is a closed n-simplex with an orientable boundary Γ^E and unit outward normal $\mathbf{N}^E$.

4.2. Approximation of the domain

Let $\phi : \mathscr{B} \mapsto \mathbb{R}$ be a continuous function, piecewise smooth, such that $\phi < 0$ in Ω, $\phi = 0$ on Γ and $\phi > 0$ in $\mathscr{B} \setminus \overline{\Omega}$. The boundary of Ω is thus the zero level-set of ϕ, and so ϕ is termed the level-set function. Next, let

$$\Phi_h = \{v_h \in H^1(\mathscr{B}) : v_h|_E \in P_1(E) \forall E \in \mathscr{T}_h\}, \tag{17}$$

where $P_1(E)$ denotes the space of polynomials of degree less or equal than 1. Let $\phi_h \in \Phi_h$ be the function that interpolates ϕ at every node in $\mathscr{T}_h$. The approximate domain Ω_h is defined as

$$\Omega_h = \{x \in \mathscr{B} : \phi_h(x) < 0\} \tag{18}$$

and let $\Gamma_h = \partial\Omega_h$ denote its boundary.

By adopting piecewise linear functions to interpolate ϕ, the zero level-set of ϕ_h, and hence the resulting approximate boundary, is simple to compute. It is composed of straight segments in two dimensions and planar polygons in the three-dimensional case. Thus the distance between Γ_h and Γ is $\mathscr{O}(h^2)$ for smooth curved boundaries, while for polyhedral ones, Γ_h and Γ may coincide nearly everywhere for some choices of ϕ. Additionally, efficient quadrature rules over the resulting sections are easy to construct. Finally, the construction of the approximate boundary in three-dimensions is greatly simplified by this choice, since the boundary may intersect four edges of a tetrahedron at points that do not necessarily lie on the same plane.

For convenience, and following a somewhat standard practice in level-set methods, we choose ϕ to be the signed distance function to Γ, defined as

$$\phi(x) = \begin{cases} -d(x,\Gamma) & \text{if } x \in \Omega, \\ d(x,\Gamma) & \text{if } x \notin \Omega, \end{cases} \tag{19}$$

where

$$d(x,\Gamma) = \inf_{y\in\Gamma} \|x - y\|, \tag{20}$$

where $\|\cdot\|$ is the Euclidean distance in $\mathbb{R}^n$.

Fast techniques to compute distance functions, particularly on Cartesian grids are discussed in [38,39,58,42]. An interesting alternative method to describe the geometry is to construct ϕ with R-functions ([50,44,45,60]) which define Ω implicitly using boolean operations over easy to define primitives. Alternatively, *surface fitting* methods can also be adopted [61,14,7,23].

Notice that the construction of the interpolant ϕ_h only requires the evaluation of ϕ at nodal locations. To prevent the appearance of sections of elements cut by the boundary that are either very small or have a bad aspect ratio, we choose the nodal values for ϕ_h as follows:

$$\phi_h(x_a) = \begin{cases} \phi(x_a) & \text{if } |\phi(x_a)| > C_{\mathrm{TOL}}h, \\ 0 & \text{otherwise}, \end{cases} \tag{21}$$

where C_{TOL} is a small constant. In our examples $C_{\mathrm{TOL}} = 10^{-6}$. By performing this correction in the construction of the interpolant ϕ_h, the location of the approximate boundary is guaranteed to either cross through the nodes or be away from them by a distance that does not deteriorate the aspect ratio of $\Omega_h \cap E$ for any element E as h goes to zero.

Correction (21) makes it possible to construct a sequence of meshes for which the distance between Γ and Γ_h is $\mathscr{O}(h)$ instead of $\mathscr{O}(h^2)$. In practice, however, it is rather difficult to encounter such a sequence of meshes. These pathological cases can be corrected by simply moving the nodes away from Γ slightly so that the correction is not needed. This is always possible for h small enough. With this in mind, we can assume that the distance between Γ and Γ_h is always $\mathscr{O}(h^2)$. A similar remark can be made for other pathological situations, such as when ϕ is identically zero at all nodes of an element. These inconveniences cannot happen for h small enough.

Finally, with the definition of Ω_h we define two submeshes or collections of elements

$$\mathscr{R}_h = \left\{E \text{ element} \in \mathscr{T}_h : \overset{o}{E} \subset \Omega_h\right\}, \tag{22}$$

$$\mathscr{Q}_h = \left\{E \text{ element} \in \mathscr{T}_h : \overset{o}{E} \cap \Gamma_h \neq \emptyset\right\}, \tag{23}$$

which are the sets of elements whose interior are completely contained in Ω_h and the set of elements whose interior are intersected by the approximate boundary, as sketched in Fig. 1. Notice that for some elements the intersection with the approximate boundary could be an entire face; these belong to $\mathscr{R}_h$.

4.3. Approximation of functions

The space of solutions on $\mathscr{Q}_h \cup \mathscr{R}_h$, is defined as the space V_h,

$$V_h = V_R \times V_Q = \left\{v_h \in L^2(\mathscr{Q}_h \cup \mathscr{R}_h) : v_h|_{\mathscr{Q}_h} \in V_Q, v_h|_{\mathscr{R}_h} \in V_R\right\}, \tag{24}$$

where

$$V_R = \{v_h \in H^1(\mathscr{R}_h) : v_h|_E \in P_1(E)\ \forall E \in \mathscr{R}_h\}, \tag{25}$$

$$V_Q = \{v_h \in L^2(\mathscr{Q}_h) : v_h|_E \in P_1(E)\ \forall E \in \mathscr{Q}_h\}. \tag{26}$$

As a result, functions in V_h are continuous across faces shared between any two elements in $\mathscr{R}_h$, but are allowed to be discontinuous across the boundary of any element in $\mathscr{Q}_h$, including those faces shared with elements in $\mathscr{R}_h$.

Discontinuous Galerkin approximations have the enticing feature that the basis functions within each element can be chosen independent of its neighbors. For elements in $\mathscr{R}_h$ this choice is immaterial to the resulting approximation. However, for those elements in $\mathscr{Q}_h$, a smart choice of the set of basis functions simplifies the imposition of essential boundary conditions. We provide one

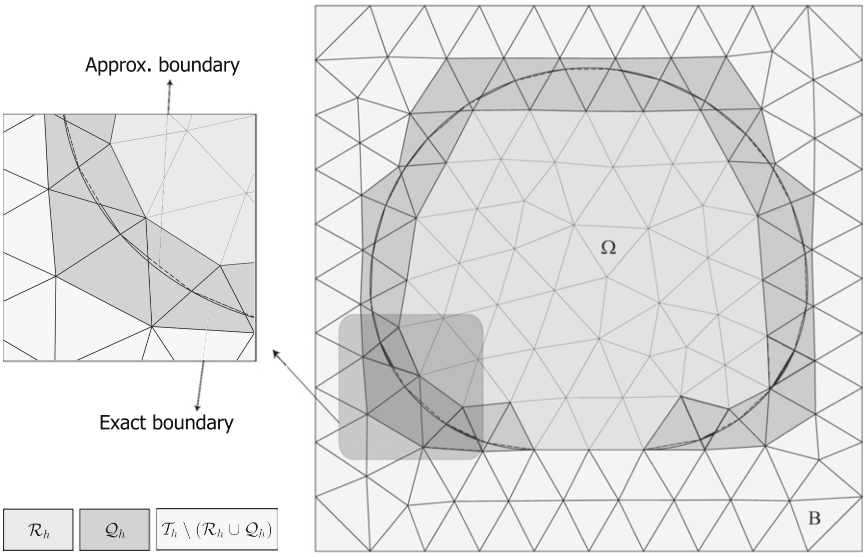

Fig. 1. The problem domain Ω is immersed in an encompassing and easy to discretize domain $\mathscr{B}$. The triangulation on the square domain $\mathscr{B}$ is denoted by $\mathscr{T}_h$. The elements of $\mathscr{T}_h$ cut by the boundary Γ are denoted by $\mathscr{Q}_h$ and the elements that lie entirely in Ω are denoted by $\mathscr{R}_h$.

such choice for elements in three spatial dimensions along with suggested quadrature rules in Appendix A.

4.3.1. Discontinuous Galerkin approximation

Since functions in V_h are permitted to contain discontinuities, we define the derivatives for these functions using a discontinuous Galerkin approximation. Let $\mathscr{M}_h$ be the submesh of $\mathscr{T}_h$ containing all the elements across whose boundaries $v_h \in V_h$ may have discontinuities. Consequently, $\mathscr{M}_h$ contains all elements of $\mathscr{Q}_h$ and those elements of $\mathscr{R}_h$ that share at least one face with an element in $\mathscr{Q}_h$.

Next, we consider the finite dimensional space of functions

$$\mathbf{M}_h^n = \{\mathbf{m}_h \in L^2(\mathscr{R}_h \cup \mathscr{Q}_h)^n; \mathbf{m}_h|_E \in (P_1(E))^n \ \forall E \in \mathscr{R}_h \cup \mathscr{Q}_h\}. \tag{27}$$

Functions in $\mathbf{M}_h^n$ are vector valued with n components. Following the framework in [1], we adopt the Bassi–Rebay numerical fluxes to construct approximations in $\mathbf{M}_h^n$ to derivatives of functions in V_h.

To this end, we recall the standard definitions of the jump $[\![\cdot]\!]$ and average $\{\cdot\}$ operators of a function across any element face e, namely

$$[\![y]\!] = y^- - y^+, \qquad \{y\} = \frac{1}{2}(y^- + y^+). \tag{28}$$

Here y is any function in $\bigcup_{E\in\mathscr{T}_h} H^1(\mathring{E})$, and the superscripts $+$ and $-$ correspond to evaluating the trace of y on either side of e. For vector-valued functions, these definitions are applied componentwise.

The approximation of derivatives of functions in V_h is obtained through the linear operator $\mathbf{D}_{DG} : V_h \mapsto \mathbf{M}_h^n$, the discontinuous Galerkin derivative, defined in every element $E \in \mathscr{T}_h$ as

$$\mathbf{D}_{DG} v_h|_E = \begin{cases} \nabla v_h|_E & E \notin \mathscr{M}_h, \\ \nabla v_h|_E + \mathbf{R}([\![v_h]\!])|_E & E \in \mathscr{M}_h. \end{cases} \tag{29}$$

Let $\Gamma_Q = \Omega_h \cap (\bigcup_{E\in\mathscr{Q}_h} \partial E)$. Notice that $\Gamma_Q \cap \Gamma_h = \emptyset$. The lifting operator $\mathbf{R} : L^2(\Gamma_Q) \mapsto \mathbf{M}_h^n$ is the unique element of $\mathbf{M}_h^n$ such that

$$\int_{\Omega_h} \mathbf{R}(v) \cdot \mathbf{m}_h \, dV = -\int_{\Gamma_Q} v\{\mathbf{m}_h\} \cdot \mathbf{n} \, dS \tag{30}$$

for all $\mathbf{m}_h \in \mathbf{M}_h^n$. The vector field $\mathbf{n}$ here is chosen as the outward normal to the "+"-side of a face. Derivatives of functions in V_h are in this way modified by the lifting operator to account for possible discontinuities across element boundaries. Since $\mathbf{R}$ is a linear operator, its action on V_h can be computed *a priori* in the form of additional basis functions in each element, with the nodal values of functions in V_h as degrees of freedom. In this way we avoid assembling and solving the system of equations (30) any time a discontinuous Galerkin derivative of a function in V_h is needed. More detailed discussions of this type of DG approximations and implementation can be found in [56,32].

4.4. Boundary conditions

In the following, we denote with Γ^d and Γ^τ two open subsets of Γ such that $\Gamma^d \cap \Gamma^\tau = \emptyset$ and $\overline{\Gamma^d} \cup \overline{\Gamma^\tau} = \Gamma$. The prescribed Dirichlet boundary conditions are assumed to be given by a function $u^d : \Gamma^d \mapsto \mathbb{R}$. Boundary conditions of the Neumann type are imposed on Γ^τ.

We address next how Dirichlet boundary conditions are imposed in the numerical method. The treatment of Neumann boundary conditions will be discussed later as part of the formulation of the method. To this end, we address two issues. First, we define the subsets Γ_h^d and Γ_h^τ of Γ_h on which to impose Dirichlet and Neumann boundary conditions respectively, in the numerical approximation. Second, we define the function u_h^d to be prescribed on Γ_h^d.

4.4.1. Approximation of the Dirichlet and Neumann parts of the boundary

We define Γ_h^d as

$$\Gamma_h^d = \bigcup_{E\in\mathscr{T}_h} \{\Gamma_h \cap E : \emptyset \neq E \cap \Gamma = E \cap \Gamma^d\} \tag{31}$$

and from here

$$\Gamma_h^\tau = \Gamma_h \setminus \Gamma_h^d. \tag{32}$$

Consequently, we have that $\Gamma_h^d \cup \Gamma_h^\tau = \Gamma_h$, and that $\Gamma_h^d = \Gamma_h$ if $\Gamma^d = \Gamma$.

With this definition for Γ_h^d there exists the possibility of having $\Gamma_h^d = \emptyset$ when $\Gamma^d \neq \emptyset$. This may happen only if the mesh $\mathscr{T}_h$ is not fine enough to have the interior of at least one element intersected by Γ^d only. A different construction can be made, in which Γ_h^d is defined as the intersection of Γ_h with any element intersected by Γ^d, and Γ_h^τ as its complement. We tested both, and they rendered qualitatively similar results.

4.4.2. Approximation of Dirichlet boundary conditions

We can now define Dirichlet boundary conditions on Γ_h^d. The problem to overcome here is that the function u^d has Γ^d as its domain, and since Γ_h does not necessarily interpolate Γ, the simple interpolation done for isoparametric elements is not possible here.

An alternative approach was described in Section 2. The idea is to first extend the definition of u^d into Ω, and then construct an approximation operator used to impose Dirichlet boundary conditions on Γ_h^d. More precisely, first a function $\bar{u} : \Omega \mapsto \mathbb{R}$, $\bar{u} \in V$, such that $\bar{u}|_{\Gamma^d} = u^d$ is found. An approximation operator $\Pi_h : V^\partial + \bar{u} \mapsto V_h$ that satisfies conditions (2) and (3) is then constructed, so that the boundary conditions are imposed by setting $u_h^d = \Pi_h \bar{u}|_{\Gamma_h^d}$.

We carry over this program in two different ways. The first one is of rather narrow applicability, since it requires the explicit construction of $\bar{u}$ away from the boundary. In this case we are able to choose Π_h as an interpolation operator, and hence we shall refer to it as the *interpolatory* method. The second one is widely applicable, since we only need the values of $\bar{u}$ on Γ^d. In this case, the approximation operator near the boundary does not necessarily interpolate any of the possible extensions $\bar{u}$ *for all h*. We shall refer to this later approach as the *non-interpolatory* method.

From a practical perspective, for the numerical method in (4) we only need to define what the values of $\Pi_h \bar{u}$ are on Γ_h^d. These are the only features of $\Pi_h \bar{u}$ that change the space $\Pi_h \bar{u} + V_h^\partial$ in which the numerical solution is sought. Herein, we will only indicate the method to construct $\Pi_h \bar{u}|_{\Gamma_h}$, leaving the definition of $\Pi_h \bar{u}$ in the interior of Ω_h unspecified. We do so for a scalar-valued function $\bar{u}$. Its extension to vector-valued functions is done componentwise, i.e., $(\Pi_h \bar{\mathbf{u}})_i = \Pi_h(\bar{u}_i)$, where v_i denotes the i-th component of vector-valued function $\mathbf{v}$.

For the convergence result, however, it is necessary to demonstrate that at least one function in V_h equal to $\Pi_h \bar{u}$ on Γ_h^d approximates u at the correct rate with respect to h, to verify conditions (2) and (3). To illustrate this, we construct one such approximation in a one-dimensional case. The numerical results are consistent with these conclusions for the two- and three-dimensional cases.

4.4.2.1. The interpolatory method. The first method relies on the availability of $\bar{u} \in V$ that extends the values of u^d into Ω. Without loss of generality, we will assume that $\bar{u}$ is smoothly defined in $\mathscr{B}$. The existence of such extension $\bar{u}$ can often be ascertained from standard trace theorems in nice enough domains. Notwithstanding the fact that $\bar{u}$ is generally costly or difficult to construct, there are situations in which such a function is available and hence can be used for computations. For example, when the value of u^d is explicitly given as a restriction to Γ^d of a function with an open subset of $\mathscr{B}$ as its domain. We provide a few examples of this later in Section 5.

Assuming that $\bar{u}$ is smooth enough, we define $\Pi_h \bar{u}|_{\Gamma_h}$ through standard interpolation operations. On each element E having a non-empty intersection with Γ_h, we let $\Pi_h \bar{u}|_{\Gamma_h \cap E}$ be the affine function on $\Gamma_h \cap E$ that interpolates $\bar{u}$ at points $\{x_1, \ldots, x_n\}$, with $\{x_1, \ldots, x_n\} \subset \Gamma_h \cap E$. We specify the interpolation points next.

In the case of a triangular element, $n = 2$, we interpolate $\bar{u}$ at the vertices of the segment $\Gamma_h \cap E$. This is depicted in Fig. 2, in which the interpolation points for $\bar{u}$ are drawn with dark circles. For tetrahedral elements, $n = 3$, there is an additional detail to take care of. As shown in Fig. A.1, the intersection of Γ_h with E may be a triangle or a quadrilateral. If $\Gamma_h \cap E$ is a triangle, then its three vertices are chosen as interpolation points x_1, x_2, x_3. In contrast, if $\Gamma_h \cap E$ is a quadrilateral, we choose the three vertices of the quadrilateral that form (one of) the triangle(s) with the largest area, see Fig. A.2. Trials with numerical experiments showed that this should be preferred over an arbitrary choice of three of the vertices of the quadrilateral, because the aspect ratio of some of these triangles can be very large. Notice that these interpolation points coincide with some vertices of the n-simplex when Γ_h intersects it along a face. An equivalent construction that interpolates $\bar{u}$ along Γ_h is not possible for every mesh or every boundary in the conforming case, and may lead to boundary locking, because conditions (2) and (3) cannot be satisfied.

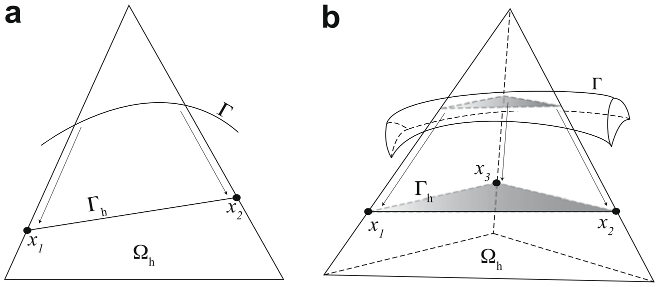

Fig. 2. Sketch of the construction of the approximate boundary values $\Pi_h \bar{u}|_{\Gamma_h}$ for elements that are intersected by the boundary Γ. When an extension $\bar{u}$ of the values prescribed on Γ is explicitly available, $\Pi_h \bar{u}|_{\Gamma_h}$ returns the affine function over the element that interpolates $\bar{u}$ at the points indicated with dark circles. In contrast, if $\bar{u}$ is not explicitly available, its values at the points of intersection between Γ and the element edges are "transported" to the corresponding interpolation points on Γ_h, as indicated in the figure.

The implementation of this method is significantly simplified in the discontinuous Galerkin context by choosing an appropriate set of basis functions. For an element $E \in \mathscr{Q}_h$, we choose basis functions $\{N_1, \ldots, N_{n+1}\}$ such that $N_i(x_j) = \delta_{ij}$ for $i, j = 1, \ldots, n$, and N_{n+1} such that $N_{n+1}|_{\Gamma_h \cap E} = 0$ and $\nabla N_{n+1} = \mathbf{n}$. Here $\mathbf{n}$ denotes the exterior unit normal to Γ_h. These are explicitly constructed in Appendix A. Elements for which one of its faces is entirely contained in Γ_h can retain their standard shape functions.

For the computation of $\Pi_h \bar{u}|_{\Gamma_h^d}$ it is enough to know $\bar{u}$ in a large enough open neighborhood of Γ^d in $\mathscr{B}$. With the chosen basis functions, $\Pi_h \bar{u}|_{\Gamma_h^d}$ is determined solely as a linear combination of $N_1, \ldots, N_n$ by interpolation of $\bar{u}$. Because of these properties, the imposition of essential boundary conditions is remarkably simplified. A similar construction when conforming approximations are adopted is unduly inconvenient.

To highlight the fact that with the proposed strategy it is possible to satisfy conditions (2) and (3) we resort to a simple one-dimensional example. This illustrates well why it is possible to approximate u with a function in V_h that is equal to $\Pi_h \bar{u}$ on Γ_h^d, with very mild restrictions on $\bar{u}$, and still retain an optimal convergence rate. The analogy of this case with higher-dimensional ones should not be taken too far, however, since in this simpler scenario optimal convergence rates can be attained even without introducing the discontinuous Galerkin enrichment along the boundary.

One possible construction for $\Pi_h u$ is illustrated in Fig. 3, for any $u, \bar{u} \in H^2(\Omega)$. For simplicity we assume that the only element intersected by the boundary, or in $\mathscr{Q}_h$, is the one with nodes x_1 and x_2. Then, for every element in $\mathscr{R}_h$, $\Pi_h u$ is set to be the piecewise linear interpolant of u at the element nodes. For the element intersected by the boundary we set

$$\Pi_h u(x) = \bar{u}(\Gamma_h^d) + \frac{u(x_2) - u(\Gamma_h^d)}{x_2 - \Gamma_h^d}(x - \Gamma_h^d). \tag{33}$$

Clearly $\Pi_h u$ satisfies condition (3), with $\Pi_h \bar{u}|_{\Gamma_h^d}$ defined as the interpolant of $\bar{u}$ at Γ_h^d. It remains to be seen that condition (2) is satisfied as well. To this end, we adopt the mesh-dependent norm

$$|||u|||_h^2 = \sum_{E \in \mathscr{T}_h} \|\nabla u\|_{0,E\cap\Omega_h}^2 + \frac{1}{h} \sum_{e \in \mathscr{T}_h} \|[\![u]\!]\|_{0,e\cap\Omega_h}^2 + \frac{1}{h}\|u\|_{0,\Gamma_h^d}^2, \tag{34}$$

where $\|\cdot\|_{0,S}$ denote the L^2-norm over the domain S, and h is a typical mesh size for a sequence of quasi-uniform meshes. This norm is

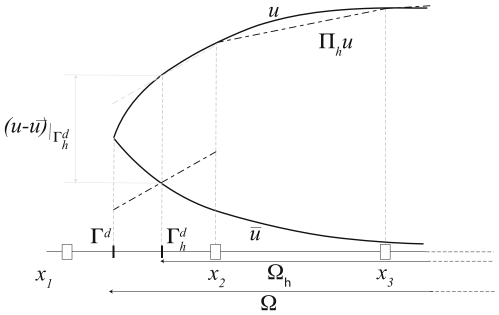

Fig. 3. One possible construction of $\Pi_h u$, shown in black dashed lines, in one-dimension with the interpolatory method. Three elements are shown, with the nodes indicated by hollow squares. The boundary Γ^d of the domain Ω, and that of its approximation Ω_h, Γ_h^d, lie within the interior of the leftmost element x_1x_2. In elements that do not intersect the boundary the exact solution u is interpolated at nodal positions. In the leftmost element $\Pi_h u$ is obtained by subtracting $(u-\bar{u})(\Gamma_h^d)$ to the linear interpolant of u at Γ_h^d and x_2, shown with a gray dashed line. In this way $\Pi_h u(\Gamma_h^d) = \bar{u}(\Gamma_h^d)$.

typically used in the analysis of discontinuous Galerkin methods. We then have that

$$|||\Pi_h u - u|||_h \leqslant |||\Pi_h u - \hat{\Pi}_h u|||_h + |||\hat{\Pi}_h u - u|||_h, \tag{35}$$

where $\hat{\Pi}_h u$ coincides with $\Pi_h u$ in $\mathscr{R}_h$ and is equal to $\Pi_h u + u(\Gamma_h^d) - \bar{u}(\Gamma_h^d)$ in $\mathscr{Q}_h$. In fact, $\hat{\Pi}_h u$ is the continuous interpolant of u in Ω_h, which we know satisfies

$$|||\hat{\Pi}_h u - u|||_h = \|\nabla \hat{\Pi}_h u - \nabla u\|_{0,\Omega_h} \leqslant Ch|u|_{2,\Omega_h}, \tag{36}$$

where $C > 0$ is a constant independent of h and $|\cdot|_{k,S}$ denotes the k-seminorm S. As in Section 2, u should be understood as the exact solution of the problem smoothly extended to $\Omega_h \setminus \Omega$, if needed.

Finally, we need to bound the first term in the right hand side of (35). Notice that $\hat{\Pi}_h u - \Pi_h u = u(\Gamma_h^d) - \bar{u}(\Gamma_h^d)$ is a constant in $\mathscr{Q}_h$, and zero elsewhere. Then,

$$\begin{aligned} |||\hat{\Pi}_h u - \Pi_h u|||_h^2 &= \frac{1}{h}\sum_e \|[\![\hat{\Pi}_h u - \Pi_h u_h]\!]\|_{0,e\cap\Omega_h}^2 \\ &= \frac{2}{h}|u(\Gamma_h^d) - \bar{u}(\Gamma_h^d)|^2. \end{aligned} \tag{37}$$

Since $u(\Gamma) = \bar{u}(\Gamma)$, we have that

$$|u(\Gamma_h^d) - \bar{u}(\Gamma_h^d)| \leqslant \left|\int_{\Gamma^d}^{\Gamma_h^d} (u' - \bar{u}')\, dx\right| \leqslant |u' - \bar{u}'|_{\infty,\Gamma^d\Gamma_h^d}|\Gamma^d - \Gamma_h^d|, \tag{38}$$

where we have taken advantage that $u' - \bar{u}'$ is continuous, since $(u-\bar{u})' \in H^1(\Omega)$. When Γ_h^d is determined as in Section 4.2 we have $|\Gamma^d - \Gamma_h^d| \leqslant Ch^2$, for some constant C independent of h. It then follows that:

$$|||\Pi_h u - u|||_h \leqslant Ch\left(|u|_{2,\Omega_h} + h^{1/2}|u' - \bar{u}'|_{\infty,\Gamma^d\Gamma_h^d}\right), \tag{39}$$

for some C independent of h and u, in agreement with condition (2). We conclude by noting that the path to obtain the bound in (39) may not be reproducible in higher dimensions, due to the use of the continuity of u', $\bar{u}'$. However, we chose it here because of its simplicity.

4.4.2.2. The non-interpolatory method. The interpolatory method has limited applicability due to the need to construct a smooth extension of u^d into the domain. Next, we propose a method that circumvents this problem. Since it only requires the prescribed boundary data, which should be provided in some form in the statement of the problem, the method can be made fully automatic. We note as well that the method does not need the solution of a system of equations with unknowns all along the boundary. This is the case, for example, when the approximate boundary conditions are constructed with some types of projections.

We assume first that the level-set function ϕ is not zero at any of the vertices of the n-simplex, i.e., none of the vertices belong to the exact boundary. The idea here is to build the value of $\Pi_h\bar{u}$ on Γ_h by "transporting" these along element edges. More precisely, the function $\Pi_h\bar{u}|_{\Gamma_h}$ is affine on $\Gamma_h \cap E$, for each element $E \in \mathscr{Q}$. For each such element, it is constructed by defining the values of $\Pi_h\bar{u}$ at the interpolation points $x_1, \ldots, x_n$ defined in Section 4.4.2.1. To this end, let e_i be the edge of the n-simplex such that $x_i = e_i \cap \Gamma_h$. There is only one such edge for each x_i, since the values of ϕ at the vertices of the simplex are different than zero. For the same reason, the set $\Gamma \cap e_i$ is non-empty. We then set

$$\Pi_h\bar{u}|_{\Gamma_h\cap e_i} = \bar{u}|_{\Gamma\cap e_i} \quad i = 1, \ldots, n. \tag{40}$$

Fig. 2 illustrates this idea, in which the "transport" of values of $\bar{u}$ from $\Gamma \cap e_i$ to $\Gamma_h \cap e_i$ is indicated with arrows. If the set $\Gamma \cap e_i$ contains more than one point, any of them is chosen to evaluate $\bar{u}$. This can only happen when the mesh is not fine enough and the boundary intersects an edge more than once.

Consider next the case in which ϕ is identically zero at one or more of the vertices. At one such vertex, $\phi_h = 0$ as well, and hence it belongs to both the exact and the approximate boundaries. In the two-dimensional case, this implies that either x_1 or x_2 lie directly on the exact boundary. A similar statement holds in the three-dimensional case for either x_1, x_2 or x_3, since as shown in Fig. A.1, in this case the intersection $\Gamma_h \cap E$ can only be a triangle. Consequently, the value of $\Pi_h\bar{u}$ at these points are obtained by directly interpolating $\bar{u}$ therein; no "transport" is needed. Notice that with this construction we recover the classical idea that, if the approximate boundary interpolates the exact one, the approximate boundary conditions are obtained by interpolating the exact ones.

Recall next that only the values of $\Pi_h\bar{u}$ on Γ_h^d are needed to impose essential boundary conditions. Because of the definition of Γ_h^d, Eq. (37), on any given element E intersected by Γ_h^d each edge e_i has a non-empty intersection with Γ_h^d and Γ^d, and $\Gamma_h \cap e_i = \Gamma_h^d \cap e_i$ and $\Gamma \cap e_i = \Gamma^d \cap e_i$. Consequently, we have that

$$\Pi_h\bar{u}|_{\Gamma_h^d\cap e_i} = \bar{u}|_{\Gamma^d\cap e_i} \quad i = 1, \ldots, n, \tag{41}$$

and hence $\Pi_h\bar{u}|_{\Gamma_h^d}$ is determined solely by $\bar{u}|_{\Gamma^d}$; no extension away from Γ^d is needed. Additionally, the intersections $\Gamma^d \cap e_i$ are simple to compute, since it only requires the solution of $\phi(x) = 0$ along the one-dimensional segment e_i.

Finally, in the one-dimensional case this idea translates into choosing $\bar{u}$ to be constant near the boundary, and the analysis in Section 4.4.2.1 is also applicable here. This illustrates why conditions (2) and (3) can be satisfied by this method. In the higher-dimensional case this idea is not equivalent to extending the boundary conditions by a constant near the boundary, since the direction along which $\bar{u}$ should be constant would change with each choice of the mesh. This is why this method is non-interpolatory.

4.4.3. On Neumann boundary conditions

Clearly, the constructions outlined in Section 4.4.2.1 or 4.4.2.2 are applicable to obtain an approximation (in some sense) on Γ_h of any function known on Γ, or parts of it, as long as the function is smooth enough. We shall take advantage of it to construct approximations of the prescribed external tractions, or Neumann boundary conditions.

In the case of the non-interpolatory method, however, it is convenient to introduce a slight modification on a few elements only. External tractions are prescribed on Γ^τ, but their approximation on Γ_h given by (40) needs to be restricted to Γ_h^τ. It is possible to have $\Gamma_h^\tau \cap e_i \neq \emptyset$, but $\Gamma^\tau \cap e_i = \emptyset$ for some of the edges e_i. This may happen in elements in which $\Gamma \cap E$ contains parts of both Γ^d and Γ^τ. Under these circumstances, it is not possible to "transport" values along such an edge e_i, since the prescribed value on $\Gamma^\tau \cap e_i$ is not defined, unless we again extend the prescribed functions away from Γ^τ. We avoid the need for this extension as follows. On any of these edges, instead of "transporting" values, we compute them as the average among the values of the approximation operator on those edges in the same element for which the "transport" is defined without the need of an extension. This operation remains local to the element, and may render discontinuous approximations of the Neumann boundary condition across elements along the boundary.

4.5. The discrete elasticity problem

The formulation of DG-IBM for elasticity consists in finding $\boldsymbol{\varphi}_h \in [V_h^\partial]^n + \Pi_h\bar{\boldsymbol{\varphi}}$ such that the functional

$$I_h[\boldsymbol{\varphi}_h] = \sum_{E\in\mathscr{T}_h} \int_{E\cap\Omega_h} [W(\mathbf{X}, \mathbf{D}_{DG}(\boldsymbol{\varphi}_h)) - \mathbf{B}_h \cdot \boldsymbol{\varphi}_h] dV - \int_{\Gamma_h^\tau} \overline{\mathbf{T}}_\mathbf{h} \cdot \boldsymbol{\varphi}_\mathbf{h}\, \mathbf{dS} \tag{42}$$

is stationary in $[V_h^\partial]^n + \Pi_h\bar{\boldsymbol{\varphi}}$, where $V_h^\partial = \{v_h \in V_h : v_h|_{\Gamma_h^d} = 0\}$, as in Section 2. The discrete force per unit volume $\mathbf{B}_h : \Omega_h \mapsto \mathbb{R}^n$ is defined as

$$\mathbf{B}_h = \begin{cases} \mathbf{B} & \text{in } \Omega \cap \Omega_h, \\ 0 & \text{in } \Omega_h \setminus \Omega. \end{cases} \tag{43}$$

The discrete external traction $\overline{\mathbf{T}}_\mathbf{h} : \Gamma_\mathbf{h}^\tau \mapsto \mathbb{R}^\mathbf{n}$ is defined as

$$\overline{\mathbf{T}}_h = \Pi_h \overline{\mathbf{T}}\big|_{\Gamma_h^\tau}. \tag{44}$$

In this definition the same approximation operator constructed to approximate Dirichlet boundary conditions is adopted. In this case the boundary data is $\overline{\mathbf{T}}$, and the value of $\Pi_h\overline{\mathbf{T}}|_{\Gamma_\mathbf{h}}$ is obtained as described in Section 4.4.2.1 or Sections 4.4.2.2 and 4.4.3, but is restricted to Γ_h^τ instead of to Γ_h^d. Of course, the boundary data $\overline{\mathbf{T}}$ needs to be smooth enough for this construction to be well-defined. We shall restrict the formulation herein to such cases.

This variational principle leads to the problem of finding $\boldsymbol{\varphi}_h \in [V_h^\partial]^n + \Pi_h\bar{\boldsymbol{\varphi}}$ such that

$$\sum_{E\in\mathscr{T}_h} \int_{E\cap\Omega_h} \left[\frac{\partial W}{\partial \mathbf{F}}(\mathbf{D}_{DG}\boldsymbol{\varphi}_h) : \mathbf{D}_{DG}\mathbf{v}_h - \mathbf{B}_h \cdot \mathbf{v}_h\right] dV - \int_{\Gamma_h^\tau} \overline{\mathbf{T}}_\mathbf{h} \cdot \mathbf{v}_\mathbf{h}\, \mathbf{dS} = \mathbf{0} \tag{45}$$

for all $\mathbf{v}_h \in [V_h^\partial]^n$. This is generally a set of nonlinear equations, except in the case in which W is that of a linear elastic material, Eq. (14).

In this case, (45) can be expressed in the form (4), with

$$a_h(\mathbf{u}_h, \mathbf{v}_h) = \sum_{E\in\mathscr{T}_h} \int_{E\cap\Omega_h} \mathbf{D}_{DG}\mathbf{u}_h : \mathbb{A} : \mathbf{D}_{DG}\mathbf{v}_h dV \tag{46}$$

as the bilinear form and

$$F_h(\mathbf{v}_h) = \sum_{E\in\mathscr{T}_h} \int_{E\cap\Omega_h} \mathbf{B}_h \cdot \mathbf{v}_h dV + \int_{\Gamma_h^\tau} \overline{\mathbf{T}}_\mathbf{h} \cdot \mathbf{v}_\mathbf{h}\, \mathbf{dS} \tag{47}$$

as the linear functional. The coercivity and continuity of the bilinear form in (46) depend on the elastic moduli $\mathbb{A}$, and an additional stabilization term may be needed, see [33,54,55].

This concludes the formulation of the method. To illustrate the asymptotic consistency of the proposed method, we analyze it in a one-dimensional example. As discussed in Section 4.4.2.1, this is a simplified case that does not illustrate well the difficulties found in higher-dimensional situations. Nevertheless, it showcases how the different approximations made play a role in the achieving the desired order of convergence.

Consider the linear elasticity problem in, without loss of generality, $\Omega = (0, 1)$, with Dirichlet boundary conditions at $\Gamma^d = 0$, and Neumann or traction boundary conditions at $\Gamma^\tau = 1$. The approximation is constructed with a sequence of meshes parameterized by h, the size of the largest element in the mesh, with $h \searrow 0$. The nodes of the each mesh are numbered, from left to right, with $x_0, \ldots, x_N$. The approximate domain is constructed on $\Omega_h = (\Gamma_h^d, \Gamma_h^\tau)$, such that $|\Gamma_h^d| = \mathscr{O}(h^2)$, $|1 - \Gamma_h^\tau| = \mathscr{O}(h^2)$, and $\Gamma_h^d, \Gamma^d \in [x_0, x_1)$ and $\Gamma_h^\tau, \Gamma^\tau \in (x_{N-1}, x_N]$. Discontinuities in the numerical solution are allowed to appear at x_1 and x_{N-1}. The elastic moduli degenerates into a scalar, denoted here with Y. We assume that B_h is smooth enough so that the solution of this problem is in $H^2(\Omega)$. We then have

$$\begin{aligned} a_h(u, v_h) - F_h(v_h) &= \sum_{E\in\mathscr{T}_h} \int_{E\cap\Omega_h} \left[Y(u' + R([\![u]\!]))(v_h' + R([\![v_h]\!])) - B_h v_h\right] dx - \overline{T}_h v_h(\Gamma_h^\tau) \\ &= \int_{\Omega_h} (-Yu'' - B_h) v_h dx + \left[Yu'(\Gamma_h^\tau) - \overline{T}_h\right] v_h(\Gamma_h^\tau) \\ &\quad + \sum_{E\in\mathscr{T}_h} \int_{E\cap\Omega_h} Yu'R([\![v_h]\!]) dx + \left[Yu'[\![v_h]\!]\right]_{x_1, x_{N-1}} \end{aligned}$$

In this identity we have used the facts that u and u' have no discontinuities, and that v_h may have discontinuities only at x_1 and x_{N-1}. We next bound each one of the terms separately.

The last two terms of the right hand side appear in other discontinuous Galerkin methods, see, e.g. [33], so here we just state that they satisfy that

$$\left|\sum_{E\in\mathscr{T}_h} \int_{E\cap\Omega_h} Yu'R([\![v_h]\!]) dx + \left[Yu'[\![v_h]\!]\right]_{x_1, x_{N-1}}\right| \leqslant Ch|u|_{2,\Omega_h} |||v_h|||_h. \tag{48}$$

The aspects of the asymptotic consistency particular to this immersed boundary method are embedded in the first two terms of the right hand side. We will use the facts that for any $v_h \in V_h^\partial$

$$|v_h|_{\infty,\Omega_h} \leqslant C|||v_h|||_h \tag{49}$$

for some $C > 0$, independent of h. The proof is straightforward, and is given at the end of this section.

By recalling that $Yu\prime(\Gamma^\tau) = \overline{T}$, we have for the second term that

$$\begin{aligned} \left|\left[Yu'(\Gamma_h^\tau) - \overline{T}_h\right] v_h(\Gamma_h^\tau)\right| &= Y\left|v_h(\Gamma_h^\tau)\right|\left(\left|u'(\Gamma_h^\tau) - u'(\Gamma^\tau)\right| + \left|\overline{T} - \overline{T}_h\right|\right) \\ &\leqslant C|||v_h|||_h Y\left(|u|_{2,\Omega_h} |\Gamma^\tau - \Gamma_h^\tau|^{1/2} + C_{\overline{T}}|\Gamma^\tau - \Gamma_h^\tau|\right) \\ &\leqslant Ch|||v|||_h\left(|u|_{2,\Omega_h} + C_{\overline{T}}h\right). \end{aligned}$$

Here $C_{\overline{T}}$ is a non-negative constant that arises from the type of approximation operator we adopt. For the non-interpolatory method, $\overline{T}_h = \overline{T}$, so $C_{\overline{T}} = 0$. For the interpolatory one, $\overline{T}$ is extended to the interior of Ω_h with any smooth function, so $C_{\overline{T}}$ is its Lipschitz constant. We have also utilized the following immediate consequence of Cauchy–Schwartz inequality

$$\left|u'(\Gamma_h^\tau) - u'(\Gamma^\tau)\right| = \left|\int_{\Gamma_h^\tau}^{\Gamma^\tau} u'' dx\right| \leqslant |u|_{2,\Omega_h} |\Gamma_h^\tau - \Gamma^\tau|^{1/2} \tag{50}$$

and the fact that $|\Gamma_h^\tau - \Gamma^\tau| = \mathscr{O}(h^2)$.

Finally, for the first term we use the fact that u is the solution of the exact problem in Ω.

$$\begin{aligned} \left|\int_{\Omega_h} (-Yu'' - B_h) v_h dx\right| = \left|\int_{\Omega_h\setminus\Omega} Yu'' v_h dx\right| &\leqslant Y|u|_{2,\Omega_h} \|v_h\|_{0,\Omega_h\setminus\Omega} \\ &\leqslant Y|u|_{2,\Omega_h} |v_h|_{\infty,\Omega_h} |\Omega_h \setminus \Omega|^{1/2} \\ &\leqslant Ch|u|_{2,\Omega_h} |||v_h|||_h, \end{aligned} \tag{51}$$

where we have used the fact that $|\Omega_h \setminus \Omega| = \mathscr{O}(h^2)$.

To conclude, we prove (49), which follows after expressing $v_h(x)$ as an integral of its derivative plus the discontinuities, and the fact that $v_h(\Gamma_h^d) = 0$, to wit

$$\begin{aligned} |v_h|^2_{\infty,\Omega_h} &\leqslant \left[\int_I |v_h'|dx + |[\![v_h(x_1)]\!]| + |[\![v_h(x_{N-1})]\!]|\right]^2 \\ &\leqslant 3\left[|v_h|^2_{1,I} + |[\![v_h(x_1)]\!]|^2 + |[\![v_h(x_{N-1})]\!]|^2\right] \\ &\leqslant 3\max(h,1)|||v_h|||^2_h. \end{aligned}$$

Here $I = (\Gamma_h^d, x_1) \cup (x_1, x_{N-1}) \cup (x_{N-1}, \Gamma_h^\tau)$.

5. Numerical examples

We now present numerical examples to examine the formulation of DG-IBM for elasticity problems with non-homogeneous boundary conditions. The first example consists in determining the deformation of a two dimensional annular ring. It is followed by a three-dimensional example, in which the torsion of a thick spherical shell is computed. In both examples we impose displacements and traction boundary conditions with the interpolatory and the non-interpolatory methods. In each case, the rate of convergence of the numerical solution to the exact one is examined and indeed shown to be optimal. These examples are followed by two nice applications – we simulate the nonlinear deformation of a femur bone under compression and of an image-based geometry. These examples serve to illustrate that the method is indeed as easy for a complex geometry as it is for a simple one.

No stabilization terms were needed in any of the examples below.

5.1. Deformation of an annulus

In this example, we are interested in describing the deformation of a long and thick walled linear elastic hollow cylinder. Its inner surface is rigidly fixed while non-zero displacements are prescribed on the outer surface. Under plane strain conditions, the domain of the problem is an annular ring $\Omega = \{\mathbf{X} \in \mathbb{R}^2 : r_0 < \|\mathbf{X}\| < r_1\}$ where $r_0 = 2$ and $r_1 = \sqrt{24.5}$ units, and $\|\cdot\|$ denotes the Euclidean norm. The boundary conditions imposed on the deformation mapping are

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X})|_{\Gamma_{r_0}} &= \mathbf{X}, \\ \overline{\boldsymbol{\varphi}}(\mathbf{X})|_{\Gamma_{r_1}} &= 6\mathbf{X}/5, \end{aligned} \tag{52}$$

where $\Gamma_{r_i} = \{\mathbf{X} \in \mathbb{R}^2 : \|\mathbf{X}\| = r_i\}$, $i = 0, 1$. Notice that we have $\Gamma^d = \Gamma$. We choose the enclosing domain $\mathscr{B}$ as a square of dimension 10 units centered at the origin and cover it by an unstructured mesh of triangles which does not necessarily conform to the annulus. The material is linear elastic, isotropic and homogeneous, with material constants $\lambda = 1$ and $\mu = 1$. The level-set function that defines Ω was chosen as the signed distance function

$$\phi(\mathbf{X}) = \max\{r_0 - \|\mathbf{X}\|, \|\mathbf{X}\| - r_1\} \tag{53}$$

and the approximate domain is given by the interior of the zero sublevel-set of its nodal interpolant, as discussed in Section 4.2. Naturally, the approximate and exact boundaries may not coincide as shown in the Fig. 4. This problem has an exact solution of the form

$$\boldsymbol{\varphi}_{ex}(\mathbf{X}) = \left(\frac{A_1}{\|\mathbf{X}\|^2} + A_2 + 1\right)\mathbf{X}, \tag{54}$$

where

$$A_1 = -0.956098 \quad \text{and} \quad A_2 = 0.239024.$$

The resulting traction on Γ_{r_1} is given by

$$\overline{\mathbf{T}}(\mathbf{X}) = 2\left((\lambda + \mu)A_2 - \frac{\mu A_1}{24.5}\right)\mathbf{e}_r. \tag{55}$$

For later use, we denote the approximation of the inner circle Γ_{r_0} by Γ_{h,r_0}, and that of the outer circle Γ_{r_1} by Γ_{h,r_1}.

Fig. 5 shows the Euclidean norm of the displacements of the annulus computed using the interpolatory method for the boundary conditions, with the extension given by

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} \quad \text{near } \Gamma_{h,r_0}, \\ \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= 6\mathbf{X}/5 \quad \text{near } \Gamma_{h,r_1}, \end{aligned} \tag{56}$$

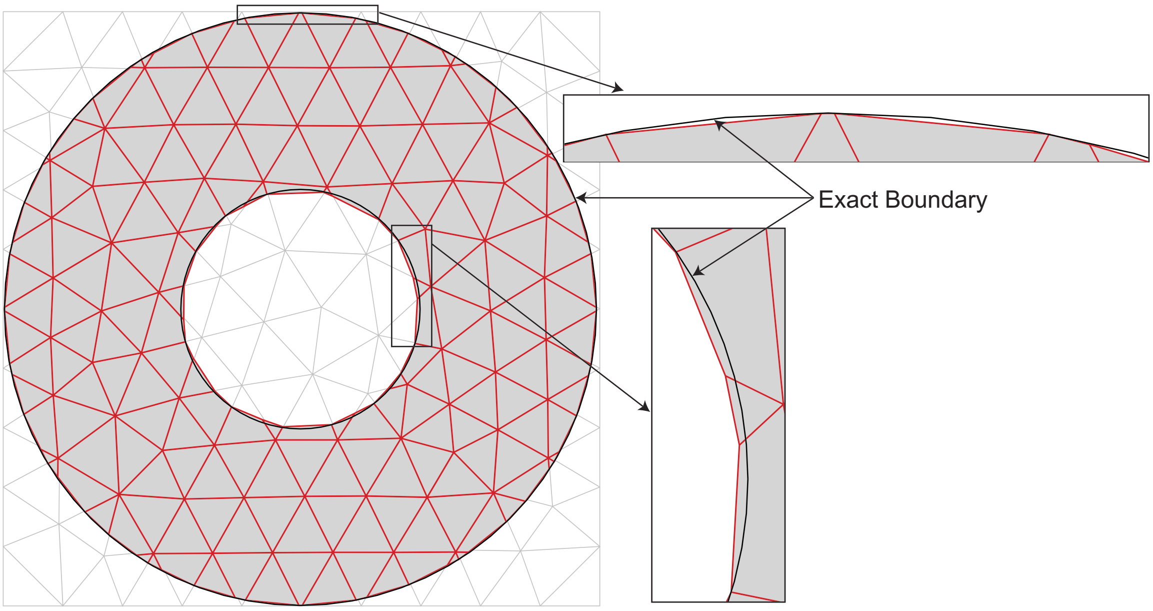

Fig. 4. Details of the approximation of the domain for the example of the annular ring. The initial background mesh $\mathscr{T}_h$ in which the boundary is immersed is shown in light gray. Notice that the approximate boundary may lie inside and/or outside the exact domain and that the exact boundary does not always intersect the element faces where the approximate one does.

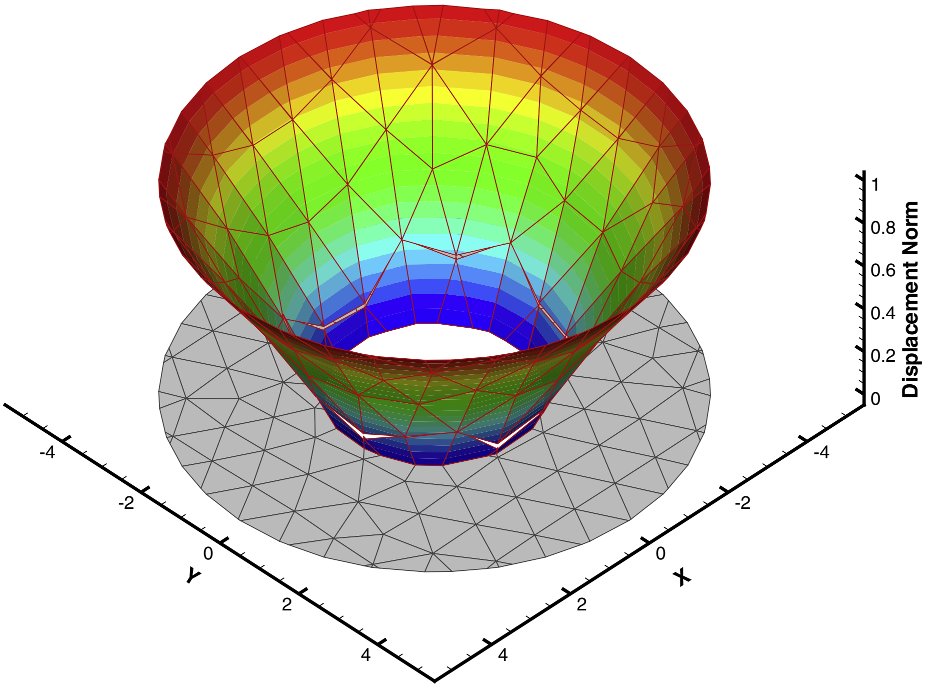

Fig. 5. Euclidean norm of the displacements at each point of the domain. The mesh is shown in the *X–Y* plane, while the norm of the displacement is displayed on the axis orthogonal to it. Notice the clearly evident discontinuities in the solution, only along the faces of elements that are intersected by the approximate boundary. These discontinuities are in fact used to recover the quadratic order of convergence of the solution.

where by "near" a set S we mean an open neighborhood of S. Since displacements are prescribed on the entire boundary, in this example $\Gamma_h^d = \Gamma_h$. Notice that this extension is equivalent to imposing zero displacements on the approximate boundary Γ_{h,r_0}. The deformation reveals the discontinuities in the displacement field near the boundary that have been utilized to yield a good approximation of the exact solution.

5.2. Torsion of a spherical shell

Next, we are interested in simulating the torsion of a spherical shell, again made of a linear elastic material with $\lambda = 1$ and $\mu = 1$. The domain of the problem is $\Omega = \{\mathbf{X} \in \mathbb{R}^3 : r_0 < \|\mathbf{X}\| < r_1\}$ with $r_0 = 0.75$ and $r_1 = 2.25$ units respectively. The encompassing domain $\mathscr{B}$ is a cube of side 5 units centered at the origin that is meshed using tetrahedral elements. The level-set function ϕ_h is the linear interpolant of the signed distance function in Eq. (53). Fig. 6 shows the tetrahedral mesh of the domain $\mathscr{B}$ and the approximate domain Ω_h.

Non-homogeneous boundary conditions are imposed everywhere on the boundary as

$$\begin{aligned}\overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_o \sin 2\xi \; \mathbf{e}_\theta \quad \text{on } \Gamma_{r_0} = \{\mathbf{X} \in \mathbb{R}^3 : \|\mathbf{X}\| = r_0\},\\ \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_1 \sin 2\xi \; \mathbf{e}_\theta \quad \text{on } \Gamma_{r_1} = \{\mathbf{X} \in \mathbb{R}^3 : \|\mathbf{X}\| = r_1\},\end{aligned} \tag{57}$$

where ξ and θ are the standard zenith and azimuthal angles in a spherical coordinate set centered at $\mathbf{X} = 0$ and

$$K_0 = 0.0984375,$$
$$K_1 = 0.8859375.$$

This problem has an exact solution given by

$$\boldsymbol{\varphi}(\mathbf{X}) = \mathbf{X} + K\|\mathbf{X}\|^2 \sin 2\xi \mathbf{e}_\theta \tag{58}$$

with $K = 0.175$. The traction acting on the surface Γ_{r_1} is computed as

$$\overline{\mathbf{T}}(\mathbf{X}) = K\mu r_1 \sin 2\xi \; \mathbf{e}_\theta. \tag{59}$$

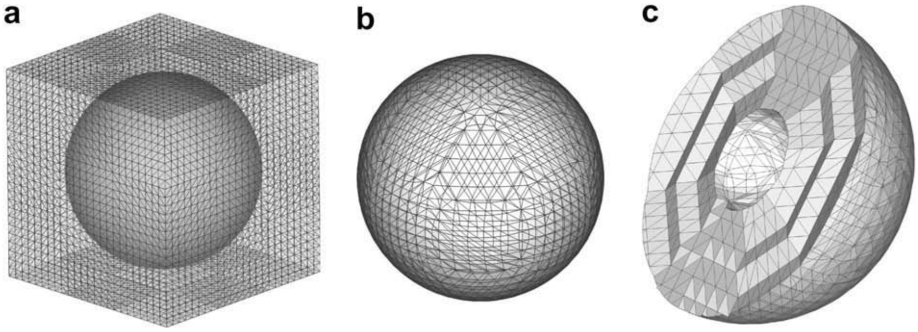

Fig. 6. Approximation of the spherical shell in a tetrahedral mesh of a cube. Shown on the left is the exact domain immersed in the mesh, followed by the approximate domain defined by the interior of the zero sublevel-set of the interpolant of the function ϕ given by Eq. (53). A section of the approximate solid shown on the right reveals more details of the approximation.

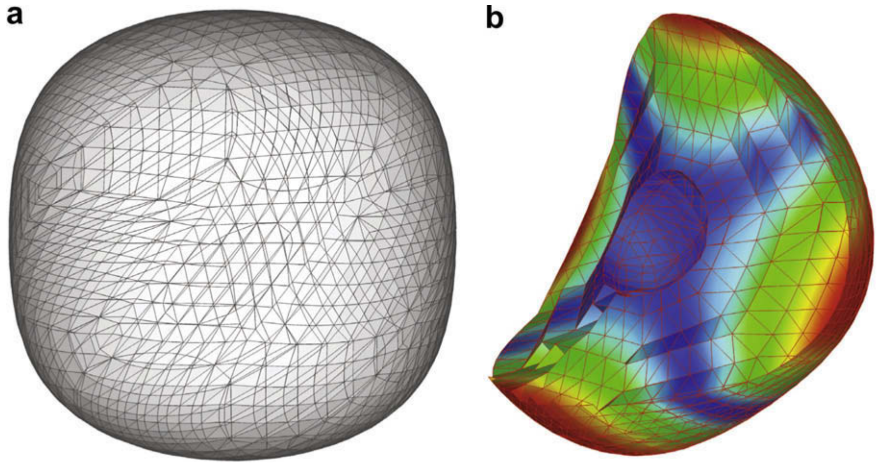

Fig. 7. Deformed configuration of the spherical shell computed using the extension given in (56). Shown on the right is a section of the deformed configuration corresponding to the section shown in Fig. 6c. The color contours correspond to the Euclidean norm of the displacement field.

The deformation of the spherical shell is shown in Fig. 7, computed by adopting the interpolatory method for the boundary conditions, with the extensions

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_o \sin 2\xi \; \mathbf{e}_\theta \quad \text{near } \Gamma_{h,r_0}, \\ \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_1 \sin 2\xi \mathbf{e}_\theta \quad \text{near } \Gamma_{h,r_1}, \end{aligned} \tag{60}$$

where analogous to the previous example, we denote with Γ_{h,r_0} and Γ_{h,r_1} the approximations of the inner and outer boundaries of the spherical shell, respectively.

5.3. Convergence curves

Next, we compute the convergence curves for the two examples in Sections 5.1 and 5.2. The error in the numerical solution of the two examples is examined and the order of convergence determined. We do so for each of the two methods to impose boundary conditions and for both the Dirichlet and Neumann type.

Part of the charm of an immersed boundary method is that the mesh can be easily refined, and such a refinement simultaneously improves the approximation of the domain, the approximation space and the approximation of the boundary conditions. In the examples that follow, triangular meshes are refined by dividing each triangle into four by joining the midpoints of each side. Similarly tetrahedral meshes are refined by dividing each tetrahedron into eight by joining midpoints of edges. It is possible to do so while keeping the aspect ratio bounded, see, e.g. [34]. In either case, the mesh parameter h is halved at successive refinements. The L^2-norm of the error in the numerical solution is computed over the domain $\Omega \cap \Omega_h$ as

$$\|\boldsymbol{\varphi} - \boldsymbol{\varphi}_h\|_0 = \left(\int_{\Omega \cap \Omega_h} (\boldsymbol{\varphi} - \boldsymbol{\varphi}_h) \cdot (\boldsymbol{\varphi} - \boldsymbol{\varphi}_h) dV \right)^{1/2}. \tag{61}$$

We will show convergence curves in this norm for most of the examples. The L^2-convergence is generally obtained from the convergence in $|||\cdot|||_h$ through a standard duality argument. In this case, the optimal convergence rate is quadratic in h. We will also show the convergence curve in $|||\cdot|||_h$ in one example, which measures the convergence rate of the strains and the discontinuities. In this norm, the optimal convergence rate is linear with h. In the presence of boundary locking, the convergence rate in both norms is adversely affected. We computed the errors over $\Omega \cap \Omega_h$, to avoid introducing the effects of the arbitrary extension of the exact solution into $\Omega_h \setminus \Omega$. We show however the computation of the error over Ω_h in one case (Fig. 9).

5.3.1. Convergence with Dirichlet boundary conditions only

In both examples Dirichlet boundary conditions were imposed on the entire boundary and consequently we had $\Gamma_h^d = \Gamma_h$. In these two cases the boundary conditions are prescribed as a restriction to Γ of some smooth function defined in an open neighborhood around it. Consequently, these functions themselves constitute natural extensions of the boundary data as needed for the interpolatory method; boundary conditions on the inner and outer boundaries are constructed by interpolating them. However, these extensions are not the only possibilities. We now consider other extensions, and numerically verify that the order of convergence does not depend on the particular extension chosen; the magnitude of the error does, however.

For the example of the annulus, another possible extension is

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} \quad \text{near } \Gamma_{h,r_0}, \\ \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + 0.2\|\mathbf{X}\| \; \mathbf{e}_r + \sin \pi (r - \sqrt{24.5}) \; \mathbf{e}_\theta \quad \text{near } \Gamma_{h,r_1}, \end{aligned} \tag{62}$$

or, alternatively, a third possible extension is

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} \quad \text{near } \Gamma_{h,r_0}, \\ \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= (r_1 + 0.2) \; \mathbf{e}_r \quad \text{near } \Gamma_{h,r_1}. \end{aligned} \tag{63}$$

The last extension, Eq. (63), corresponds to adopting as the value for the Dirichlet boundary condition at an interpolation point in Γ_h the value prescribed at its closest point in Γ.

Fig. 8a shows the error as a function of the mesh parameter h. Adopting any of these extensions for the two-dimensional example, the numerical solution converges to the exact one quadratically in h. Finally, Fig. 9 shows the convergence curve in the norm $|||\cdot|||_h$ defined in (34) when extension (56) is adopted. The value of the error in $\Omega_h \setminus \Omega$ was computed by simply evaluating the expression of the exact solution (54) there. As expected, the error for the derivatives and the magnitude of the discontinuities decrease linearly with h, an optimal convergence rate.

Similarly, for the example of the spherical shell, the following functions constitute a smooth extension

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_0[\sin 2\xi \mathbf{e}_\theta + \sin(\pi(r - r_0))] \; \mathbf{e}_\xi \quad \text{near } \Gamma_{h,r_0}, \\ \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_1[\sin 2\xi \mathbf{e}_\theta + \sin(\pi(r - r_1))] \; \mathbf{e}_\xi \quad \text{near } \Gamma_{h,r_1}. \end{aligned} \tag{64}$$

Once again, optimal quadratic convergence in the L^2 norm is confirmed by Fig. 8(b).

Finally, we utilize the interpolatory method to impose essential boundary conditions on Γ_{h,r_0}, with analytical extensions (56) and (60) for the annulus and the spherical shell, respectively. On Γ_{h,r_1}, however, essential boundary conditions are imposed through

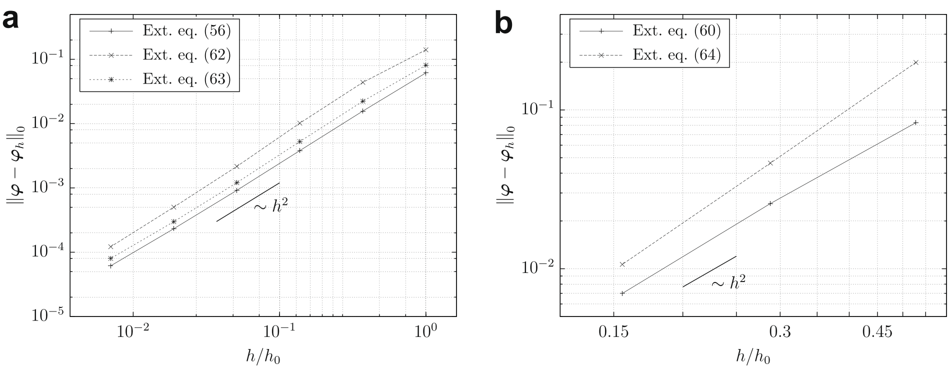

Fig. 8. Rate of convergence of the numerical solution computed using different extensions to impose essential boundary conditions with the interpolatory method. Results for both the annulus and the spherical shell are shown. Optimal convergence rates are achieved in both cases.

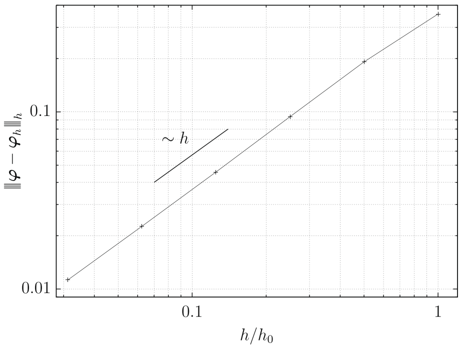

Fig. 9. Convergence curve in the norm $|||\cdot|||_h$, which measures the error in the approximation of the derivative inside each element and the magnitude of the discontinuities. The linear convergence rate with h is optimal, and reflects the absence of boundary locking.

the non-interpolatory method. Fig. 10 shows that once again, the solution converges with the optimal order for both the two- and three-dimensional examples.

5.3.2. Convergence with Neumann boundary conditions

We investigate next the convergence curves in the two examples when traction boundary conditions are imposed on parts of the boundary. To this end, we impose traction boundary conditions on the outer boundary of the domain, Γ_{r_1}, and retain the same essential boundary conditions in the inner one, Γ_{r_0}. The tractions on the outer boundary are those obtained from evaluating the exact solution there. More precisely, for the annulus

$$\overline{\boldsymbol{\varphi}}(\mathbf{X}) = \mathbf{X} \quad \text{on } \Gamma_{r_0}, \tag{65}$$

$$\overline{\mathbf{T}}(\mathbf{X}) = 2((\lambda+\mu)A_2 - \mu A_1/24.5)\mathbf{e}_r \quad \text{on } \Gamma_{r_1} \tag{66}$$

and for the spherical shell

$$\overline{\boldsymbol{\varphi}}(\mathbf{X}) = \mathbf{X} + K_0 \sin 2\xi \mathbf{e}_\theta \quad \text{on } \Gamma_{r_0}, \tag{67}$$

$$\overline{\mathbf{T}}(\mathbf{X}) = K\mu r_1 \sin 2\xi \ \mathbf{e}_\theta \quad \text{on } \Gamma_{r_1}. \tag{68}$$

With these boundary conditions, the exact solutions of the two problems remain unaltered, i.e., still given by (54) and (58). That the boundary of each of the two domains can be decomposed into two disconnected components is very convenient, because we can independently choose the type of boundary conditions on each one of them.

Now, we inspect the performance of the interpolatory and non-interpolatory methods to impose traction boundary conditions. Essential boundary conditions on $\Gamma_h^d = \Gamma_{h,r_0}$ are imposed via the interpolatory method, with extensions (56) and (60) for the two- and three-dimensional examples, respectively. For the two-dimensional example, we consider the following alternative extensions for the traction on Γ_{h,r_1}

$$\overline{\mathbf{T}}(\mathbf{X}) = 2[(\lambda+\mu)A_2 - \mu A_1/24.5]\mathbf{e}_r, \quad \text{or} \tag{69}$$

$$\begin{aligned}\overline{\mathbf{T}}(\mathbf{X}) = {} & 2[(\lambda+\mu)A_2 - \mu A_1/\|\mathbf{X}\|^2]\ \mathbf{e}_r \\ & + (r_1 - \|\mathbf{X}\|)[(\lambda+\mu)A_2 + \mu A_1/\|\mathbf{X}\|^2]\ \mathbf{e}_\theta,\end{aligned} \tag{70}$$

while for the three-dimensional one we chose

$$\overline{\mathbf{T}}(\mathbf{X}) = K\mu r_1 \sin 2\xi \ \mathbf{e}_\theta, \tag{71}$$

$$\overline{\mathbf{T}}(\mathbf{X}) = K\mu[\|\mathbf{X}\| \sin 2\xi \mathbf{e}_\theta - (r_1 - \|\mathbf{X}\|)\sin^2\xi \ \mathbf{e}_\xi], \tag{72}$$

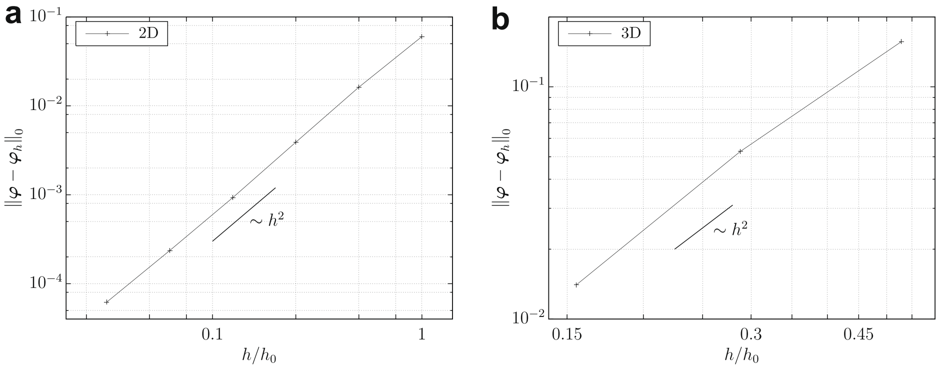

Fig. 10. Convergence curves computed using the non-interpolatory method to impose Dirichlet boundary conditions on Γ_{h,r_1} for the two- and three-dimensional examples. The convergence rate is approximately quadratic in both cases.

all of them valid near $\Gamma_h^\tau = \Gamma_{h,r_1}$. Extensions (69) and (71) simply evaluate (66) and (68) on Γ_{h,r_1}. However, the other two are quite arbitrary, included simply to illustrate the independence of the order of convergence on the choice of the (smooth) extension. Fig. 11 shows the optimal convergence of the solutions obtained with each one of these extensions when the interpolatory method is utilized for the approximation of the normal tractions. Similarly, Fig. 12 shows the convergence of the computed solution for the non-interpolatory one.

5.3.3. Combination of Dirichlet and Neumann boundary conditions

We have not yet dealt with a case in which Dirichlet and Neumann boundary conditions are prescribed on the same component of Γ. We consider this possibility now. To avoid the often–encountered singularities near $\overline{\Gamma^\tau} \cap \overline{\Gamma^d}$, we once again set the boundary conditions on Γ^d and Γ^τ by evaluating those of the smooth exact solutions there.

In both examples, essential boundary conditions given by (65) and (67) are imposed on Γ_{r_0}. For the example of the annulus, we consider the following boundary conditions on Γ_{r_1}

$$\overline{\boldsymbol{\varphi}}(\mathbf{X}) = 6\mathbf{X}/5 \quad \text{if } \mathbf{X}\cdot\mathbf{e}_v \leqslant \sqrt{5}, \tag{73a}$$

$$\overline{\mathbf{T}}(\mathbf{X}) = 2\left((\lambda+\mu)A_2 - \frac{\mu A_1}{24.5}\right)\mathbf{e}_r \quad \text{if } \mathbf{X}\cdot\mathbf{e}_v > \sqrt{5}. \tag{73b}$$

Here $\mathbf{e}_v$ is a any unit vector. Similarly, for the example of the spherical shell we impose the following boundary conditions on Γ_{r_1}

$$\overline{\boldsymbol{\varphi}}(\mathbf{X}) = \mathbf{X} + K_1 \sin 2\xi \mathbf{e}_\theta \qquad \text{if } \mathbf{X}\cdot\mathbf{e}_v \leqslant \sqrt{1.25}, \tag{74a}$$

$$\overline{\mathbf{T}}(\mathbf{X}) = K\mu r_1 \sin 2\xi\ \mathbf{e}_\theta \qquad \text{if } \mathbf{X}\cdot\mathbf{e_v} > \sqrt{1.25}. \tag{74b}$$

The Neumann boundary Γ^τ for each example is the subset of Γ_{r_1} defined in (73b) or (74b), while Γ^d is the complement of its closure in Γ. Essential boundary conditions on $\Gamma_h^d = \Gamma_{h,r_0}$ are imposed via the interpolatory method, with extensions (56) and (60) for the corresponding problems. In contrast, near Γ_{h,r_1}^d and Γ_{h,r_1}^τ the extensions are constructed by evaluating, for each example, (73) and (74) near the boundary. Fig. 13 shows the optimal order of convergence when both the interpolatory and non-interpolatory methods are adopted on Γ_{h,r_1}.

5.4. Stabilization

Discontinuous Galerkin methods for elliptic problems may need stabilization terms. These often take the form of a penalization of the magnitude of the discontinuities in the numerical solution.

We did not add a stabilization term in any of the simulations presented so far. However, we performed numerical experiments for linear elastic problems analogous to those carried out in [32] for the Poisson's equation. The gradual onset of the boundary locking phenomenon as discontinuities near the boundary are more severely penalized is also observed in this case. This is what is expected from the conforming finite element space when the constraints imposed by the prescribed boundary conditions are strictly enforced. Indeed, this is shown in Fig. 14a for the case of the annulus, which reveals a suboptimal $\mathcal{O}(h)$ convergence as the parameter scaling the stabilization term, β, is very large. A similar situation is reflected in the 3D example. Fig. 14b shows the $L^2(\Omega\cap\Omega_h)$-error for different values of β for two successive refinements of the mesh. The $\mathcal{O}(h)$ convergence for large values of β is in contrast with the $\mathcal{O}(h^2)$ convergence for values of β that are small or zero.

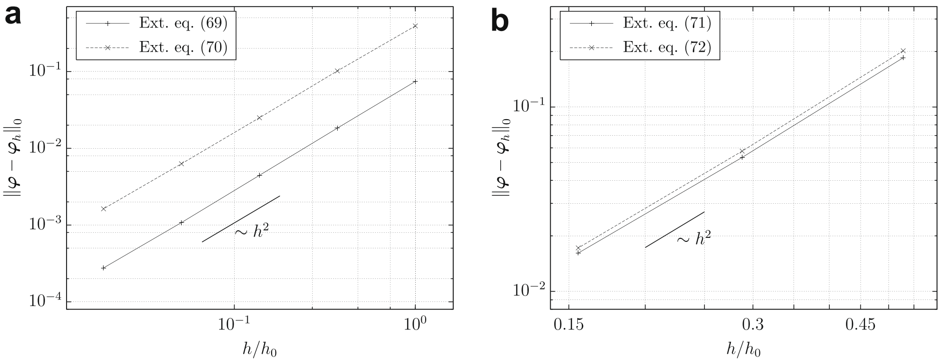

Fig. 11. Quadratic convergence of numerical solutions when adopting the interpolatory method to construct approximate traction boundary conditions on Γ_{h,r_1} in the annulus and spherical shell examples.

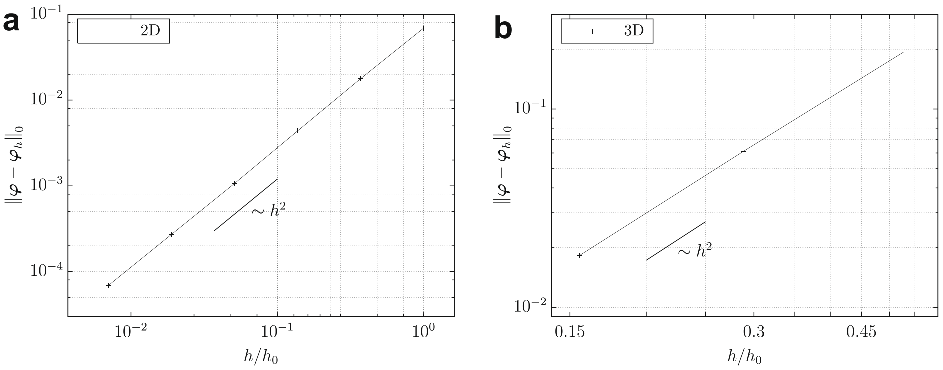

Fig. 12. Use of the non-interpolatory method for the approximation of traction boundary conditions on Γ_{h,r_1}. Optimal convergence rates are achieved.

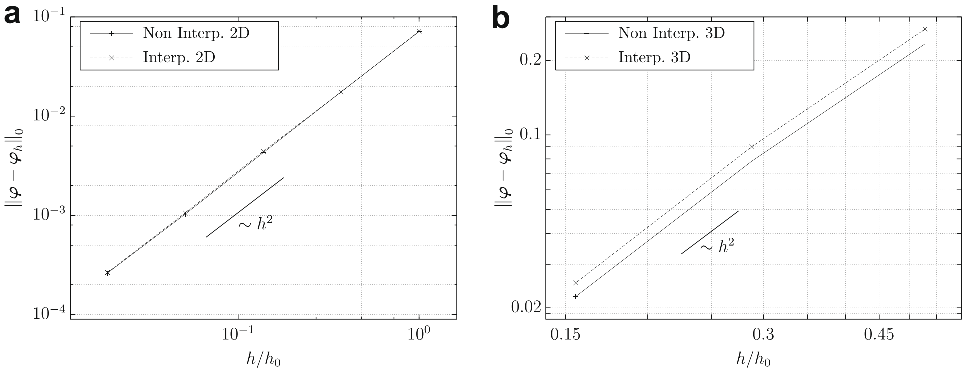

Fig. 13. Combination of Dirichlet and Neumann boundary conditions on Γ_{r_1}. Both the interpolatory and the non-interpolatory method display optimal convergence rate.

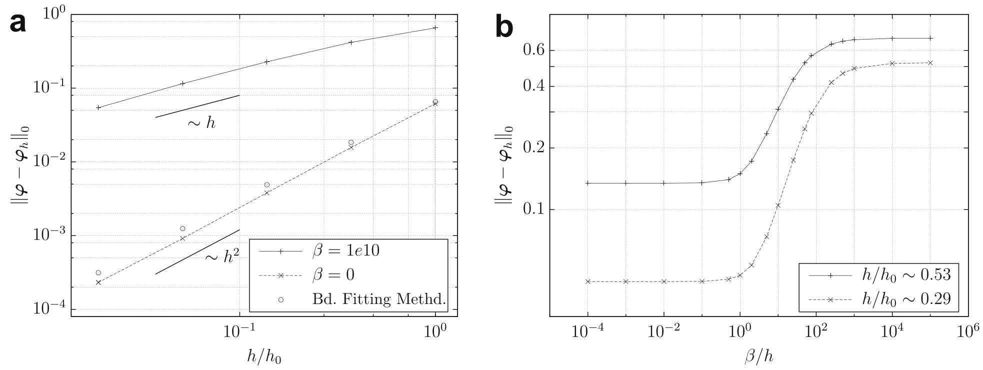

Fig. 14. Smooth onset of the locking phenomenon when transitioning from a DG to a conforming discretization. We do so by increasing the relative importance of a stabilization term that consists of a quadratic penalization of the magnitude of the discontinuities, see, e.g. [32]. When the stabilization parameter, β, is large, an essentially conforming discretization is recovered. Shown on the left are the convergence curves for the example of the annulus with no stabilization, and a large stabilization parameter. Penalization of the discontinuities results in a loss of a full order of convergence. Also shown is the convergence curve for the same example but when adopting the boundary–fitting method described in Section 5.6. Shown on the right is the error in the numerical solution for the example of the spherical shell for different values of β. The poor approximation of the solution for large β is reflected in a larger error and lower reduction in the error upon subsequent refinement.

5.5. On the error in the interior of the domain

The appearance of boundary locking can be roughly described as a consequence of the impossibility to approximate functions with non-zero gradients parallel to the boundary, see [32]. It is then fair to wonder whether the loss of accuracy is restricted to a region at a distance of order h near the boundary, or it actually pollutes the accuracy of the solution well in the interior of the domain. The numerical example that follows shows that the latter can actually happen; the loss of accuracy near the boundary may affect the rate of convergence of the solution in the interior of the domain as well.

The example consists in setting $\beta = 10^{10}$ so as to force a nearly conforming approximation space, and then find the solution of the two-dimensional annulus problem with Dirichlet boundary conditions imposed on the entire boundary. These were imposed via the interpolatory method with the extensions in (56). The L^2-error of the solution over the domain $\{\mathbf{X} \in \Omega : 3 < \|\mathbf{X}\| < \sqrt{20}\}$, whose closure is strictly included in Ω, is plotted in Fig. 15. The convergence rate is found to be suboptimal – the error decreases linearly with h.

5.6. Comparison with a boundary-fitting method

We compare next how the accuracy of DG-IBM compares with a conventional finite element calculation in which boundary-fitting meshes are used. For simplicity, we chose a polygonal domain included in the interior of and very close to the annulus to perform a comparison. Dirichlet boundary conditions were imposed everywhere on the boundary, by interpolating the *exact* solution to the annulus example there. Meshes of triangles exactly fitting the boundary of the polygon were constructed, and standard P_1 conforming elements were adopted. The numerical solutions obtained in this way are approximations to the exact solution in the interior of the polygon.

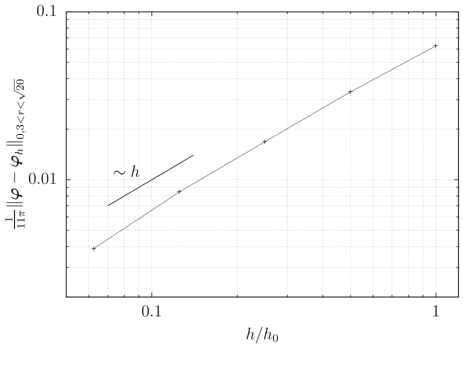

Fig. 15. Convergence curve for the error in the interior of the annulus of the two-dimensional example, computed with a nearly conforming discretization ($\beta = 10^{10}$). The error in the figure was computed over the region $\{3 < \|\mathbf{X}\| \sqrt{20}\}$. This example show that the suboptimal convergence rate due to boundary locking is not necessarily restricted to a region near the boundary, but it may pollute the solution everywhere in the domain.

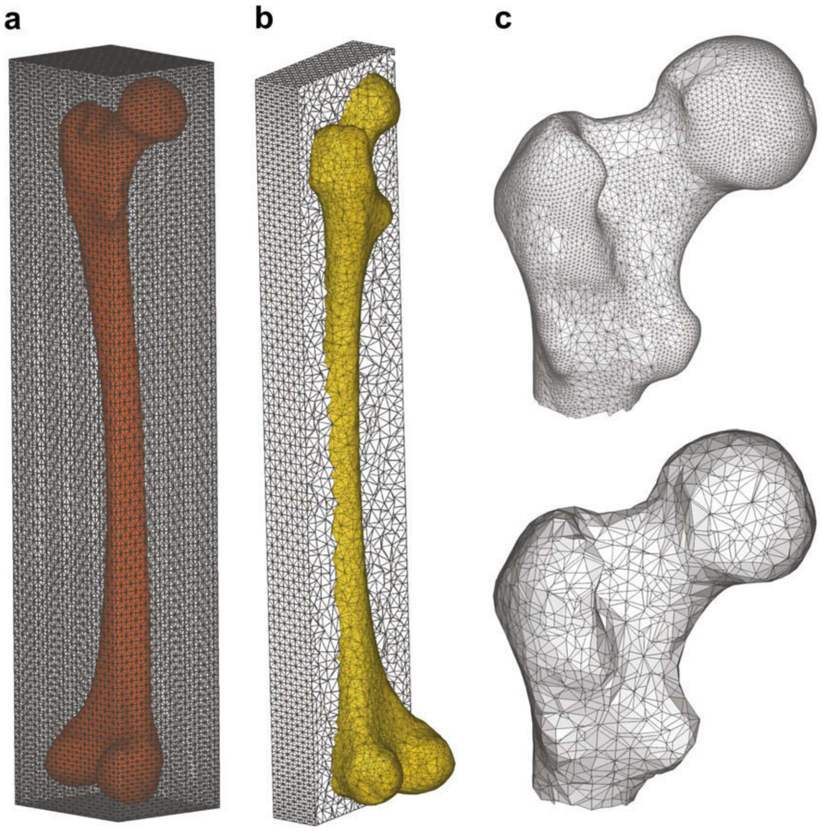

Fig. 16. Approximation of the geometry of a femur by immersing the boundary in a mesh. Shown on the far left is the exact surface immersed in a parallelepiped. The parallelepiped is meshed with tetrahedra. The signed distance function is then constructed, and its zero level-set is evaluated to obtain the approximate geometry, shown in the middle. On the right, the exact and approximated geometry for the head of the bone are displayed.

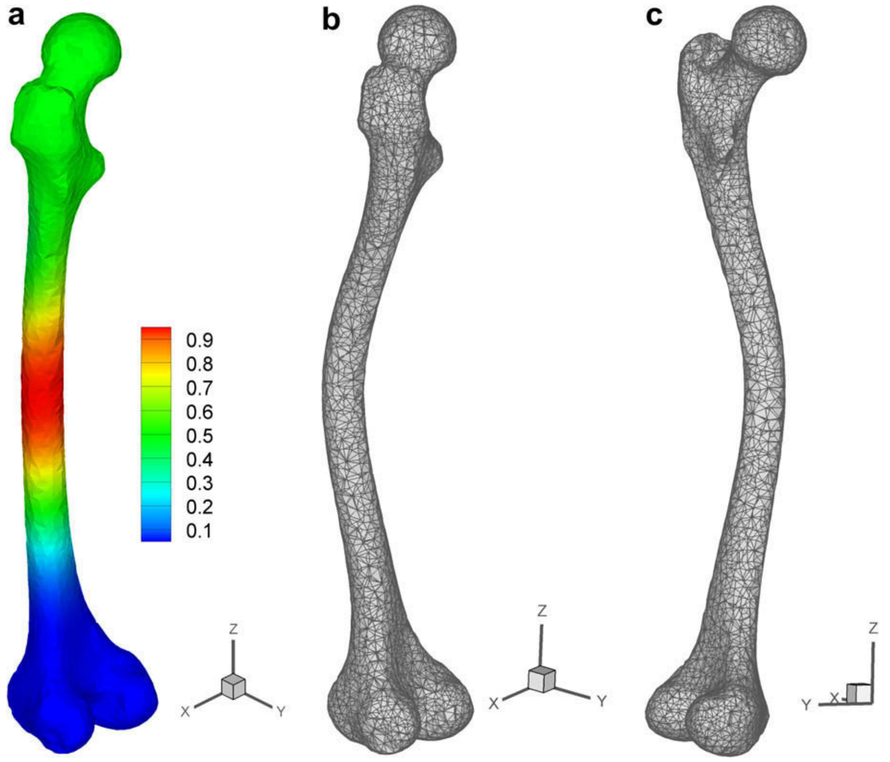

Fig. 17. Displacement contours and deformation of the femur as a result of fixing the bottom and imposing downward displacements at the top. The contours of the norm of the displacement field are shown in (a). The deformed configuration of the bone is shown in (b) and (c), from two different angles. The displacement field in (b) and (c) was scaled by a factor of 2 for visualization purposes.

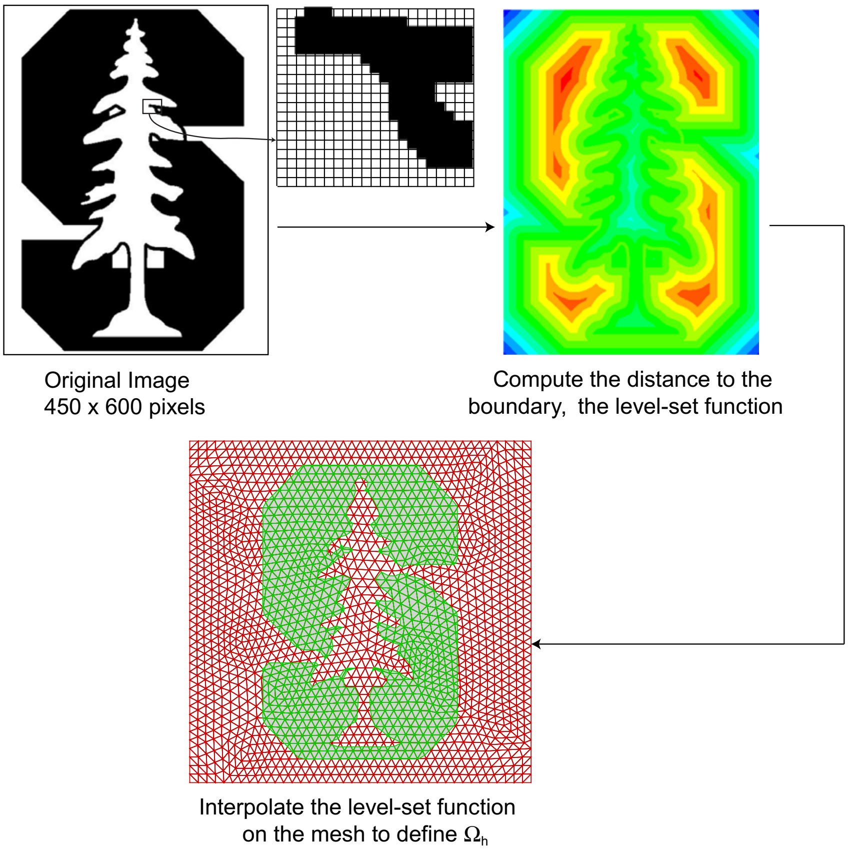
Original Image
450 x 600 pixels
Compute the distance to the
boundary, the level-set function
Interpolate the level-set function
on the mesh to define Ω_h

Fig. 18. Step towards the generation of image-based geometries. The starting point is a black and white image made of pixels, each one of them painted with one of two colors. A level-set function is constructed by computing the signed distance of each point to the closest pixel with an opposite color. This function is in turn interpolated over the background mesh. The boundary of the zero sublevel-set of this interpolant is then the approximate boundary of the domain. Notice that in the example shown here some features of the geometry are clearly lost; a finer background mesh is needed.

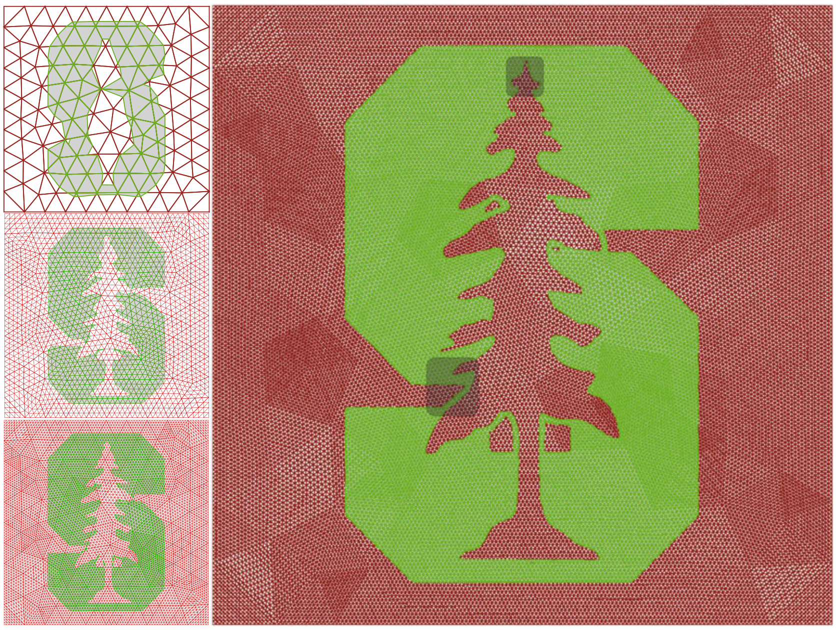

Fig. 19. Approximate domains computed with progressively finer meshes obtained by subdividing each triangle in the first mesh into four similar ones. Fig. 20 shows details of the approximate domain in the regions indicated with gray boxes in the most refined mesh.

Fig. 14a shows the result of this comparison. These are plotted together with the results obtained with DG-IBM for the same example but stated on the annular domain instead. The magnitudes of the errors in the two cases are very close. Consequently,

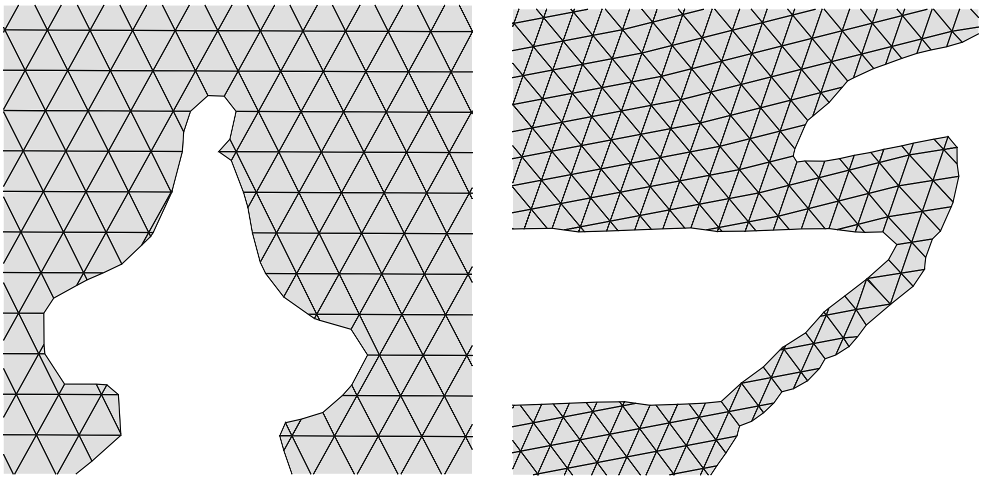

Fig. 20. Details of the approximate domain for the finest mesh used, before deformation.

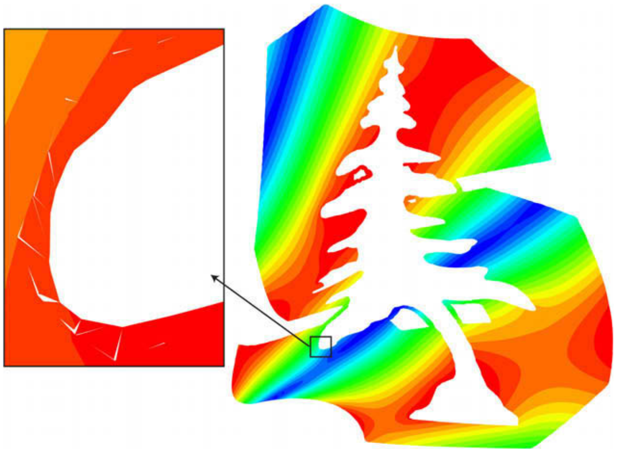

Fig. 21. Nonlinear elastic deformations of the image-based geometry. The contour plot shown is based on the Euclidean norm of the displacement field at each point. On the left, an enlarged version near the boundary, where the jumps in displacements can be easily observed.

in addition to retaining optimal convergence rate, for this example DG-IBM also returns errors whose magnitude is similar to those of a comparable boundary-fitting method. Of course, this observation may strongly depend on the problem and the mesh.

5.7. Numerical examples with nonlinear elastic materials

Next, we present two applications that showcase the use of DG-IBM for simulations of nonlinear elastic bodies of complex shapes.

5.7.1. Compression of a femur

As an illustration of the proposed method's capability to handle realistic geometries, we consider an example involving the deformation of a femur. While boundary fitting meshes are readily available for geometries far more complicated than the bone model studied here, the problem of meshing any such geometry is significantly simplified here by meshing a parallelepiped enclosing the bone and letting the DG-IBM framework take care of the rest.

A common procedure used to convert samples or physical models into a computer/CAD representation is to scan or profile the surface. Surface reconstruction algorithms may be used to correct for irregularities and represent the scanned surface as a tessellation. Mesh generation algorithms discretize the domain enclosed by the tessellation for use in a conventional finite element model. With an immersed boundary method, however, the last step is not required; the tessellation of the surface is used to (approximately) reconstruct the domain in a possibly non-boundary-fitting mesh.

The ("exact") surface of the femur adopted here consists in a fine mesh of triangles. The enclosing domain $\mathscr{B}$ is a parallelepiped, which is *easily* meshed, in this case with approximately 159,100 tetrahedral elements. The signed distance to the surface needs to be computed at the nodes of this mesh,[4] and the approximation of the geometry then follows as explained in Section 4.2. The resulting approximation of the domain is shown in Fig. 16.

As mentioned earlier, the goal of this example is to demonstrate the ability of the method to handle complex geometries. For this reason, we adopt a simple nonlinear elastic constitutive law (12) for the material, with $\lambda = \mu = 1$. For the same reason, no attempt to model the real mechanical behavior of bone is made here; bones may not undergo very large elastic deformations, and realistic constitutive relations are far more delicate than (12).

The bone was rigidly clamped at the bottom (epicondiles) and downward displacements imposed at the head. Dirichlet boundary conditions were imposed on the approximate boundary via the interpolatory method with

$$\overline{\varphi}_h(\mathbf{X}) = \begin{cases} \mathbf{X} & \text{if } \mathbf{X}\cdot\mathbf{e}_z \leqslant 8 \text{ units}, \\ \mathbf{X} - 1.0\ \mathbf{e}_z & \text{if } \mathbf{X}\cdot\mathbf{e}_z \geqslant 40 \text{ units}. \end{cases} \tag{75}$$

The length of the sample (along the Z axis) is 46.6 units while the lateral dimensions (along the X and Y axes) are 8.8 and 7.4 units; see Fig. 17 for the orientation of the axes relative to the bone. The origin of the axes is located slightly below the bone in Fig. 17. In (75), $\mathbf{e}_z$ is a unit vector along the Z-direction. The displacement imposed on the upper surface is about 2.1% of the length. Fig. 17 plots the resulting deformation.

5.7.2. Application to image-based geometries

Image-based geometries are encountered in many situations, perhaps most often in medical imaging. A possibly undesirable feature of extracting geometries from images is that the boundary is naturally jagged, or pixilated. Most often these geometries are only approximations of the exact one. In this situation the solution near the boundary is hardly relevant, and only its behavior far away from it may be meaningful. Consequently, highly refined meshes

[4] The signed distance to the surface is only needed at the nodes that are close to the surface, not on the entire mesh.

near the boundary are unnecessary, and the jagged boundary may be replaced by yet another approximation of the exact one, but smoother. The construction of the approximate domain in DG-IBM does precisely that; the approximate boundary in DG-IBM retains feature sizes of the order of the mesh size, averaging out smaller details.

We demonstrate the use of the current method to easily simulate complex geometries extracted from images. The example consists of simulating the nonlinear elastic behavior of a two-dimensional object whose geometry is given by the black pixels of a digital black-and-white image. A black-and-white image consists of a rectangular Cartesian grids in which each cell, or pixel, has one of two possible values, say 0 for white and 1 for black. The particular geometry that we consider here is shown in Fig. 18.

The signed distance to the (jagged) boundary was used as the level-set function. For this example, the distance is positive if the point is outside the domain, in a white pixel, negative when it lays in a black pixel, and zero at the boundary. The approximate domain determined by the interpolant of the level-set function is also shown in Fig. 18. It is evident that if the mesh $\mathscr{T}_h$ over which the level-set function is interpolated is much coarser that the pixel size, the domain is naturally smoothed out.

The initial background mesh is shown in gray in Fig. 19. Finer meshes are obtained by subdividing each triangle into four smaller ones. The resulting approximate domains for each of these meshes along with some of the details and features of the finest mesh are shown in Fig. 20.

Dirichlet boundary conditions are applied all along the boundary via the interpolatory method. The displacements along this boundary are interpolated from

$$\overline{\varphi}_h(\mathbf{X}) = \frac{\mathbf{X}}{\|\mathbf{X}\|}\cos\left(5\theta - \frac{\pi}{2}\right), \qquad (76)$$

where $\mathbf{X}$ is the position vector with respect to the lower left corner of the background mesh and θ is the angle formed by this position vector with the lower boundary of the same mesh. Once again, we use the same square domain $\mathscr{B}$ employed in the previous two-dimensional simulations – a square of size 10 units. The distances from the image to the lower boundary and to the left boundary of the background mesh are approximately 0.68 and 2.14 units respectively.

The material is assumed to be homogeneous throughout the domain, with $\lambda = \mu = 1$ in the constitutive relation given by (12). The resulting elastic deformation is shown in Fig. 21. The presence of jumps near the boundary are also shown therein.

6. Summary

We proposed an immersed boundary method for problems in elasticity, in which a wide class of non-homogeneous boundary conditions of both essential and natural type can be imposed while retaining an optimal convergence rate.

In designing the method, we stated two conditions that, if satisfied, guarantee an optimal convergence rate in the energy norm. There is considerable freedom for the design of methods that satisfy them. The proposed method here is simply one among many ways of accomplishing this. The key ingredients of the method are: (a) the approximation of the domain through a level-set function in the finite element space, (b) the addition of a discontinuous Galerkin discretization near the boundary to prevent boundary locking, and (c) the imposition of boundary conditions through the construction of an approximation of a smooth extension of the prescribed values on the boundary to a large enough neighborhood of the domain. We proved that the proposed method satisfies the convergence conditions in the one-dimensional case, for the purpose of illustrating how the different ingredients of the method play a role in attaining an optimal convergence rate.

In proposing the method to construct essential and natural boundary conditions, we adopted as a starting point that data on boundary conditions is provided as functions defined on the Dirichlet or Neumann boundary *only*. We then added that, if these functions are given as a restriction to the boundary of a smooth enough function in a neighborhood of the boundary, then boundary conditions can be imposed by interpolating them. Such a method was called the interpolatory method. This is, however, not always the case. Consequently, we proposed a second method, termed the non-interpolatory method, that necessitates the definition of these functions only at the boundary. This enables the method to become fully automatized, i.e., a code can receive the data of the boundary conditions on the exact boundary only and construct an optimally convergent sequence of approximations with it.

The optimal convergence rate of the method was verified with extensive numerical examples in linear elasticity that involved the two approaches to approximate boundary conditions, for both the essential and natural type. Furthermore, the versatility of the method to address nonlinear problems with complex geometries was demonstrated with two examples in nonlinear elasticity.

An interesting application of this method may be found in the context of nonlinear elasticity problems with very large deformations. It is known that under these conditions it may be necessary to resort to new discretizations of the domain as the material is deformed. The immersed boundary method could then be used to efficiently construct a new mesh on the deformed configuration of the material, and subsequently adopt the latter as a new reference configuration of the body.

In the near future we plan to address the convergence analysis for the two- and three-dimensional cases, as well as to work on higher order versions of the method.

Acknowledgement

We would like to thank Matteo Negri for helpful discussions about the convergence of these methods.

Appendix A. An implementation of DG-IBM

A simple implementation of DG-IBM using linear tetrahedral elements for three-dimensional problems is discussed here. This includes a description of types of element boundary intersections, a convenient choice of basis functions and simple quadrature rules for integration. An implementation for linear triangle elements was outlined in [32].

A.1. Element–boundary intersections

As mentioned earlier, Γ_h is composed of plane sections in three-dimensional (3D) domains. Fig. A.1 shows all possible qualitatively different plane-tetrahedron intersections. Referring to $E \cap \Omega_h$ for $E \in \mathscr{Q}_h$ as the "cut element," we see that cut elements can be tetrahedra, or wedges with six nodes or pyramids having five nodes. Also note that $\Gamma_h^E (= \Gamma_h \cap E)$ is a triangle for tetrahedra, pyramids and wedges of type (a), and a quadrilateral for type (b) wedges.

A.2. Basis functions

The standard finite element basis functions are used for elements belonging to $\mathscr{R}_h$. For elements in $\mathscr{Q}_h$ that use a discontinuous Galerkin approximation, we choose a convenient set of basis functions that permits easy imposition of Dirichlet boundary conditions. This is the case when one of the shape functions is zero on

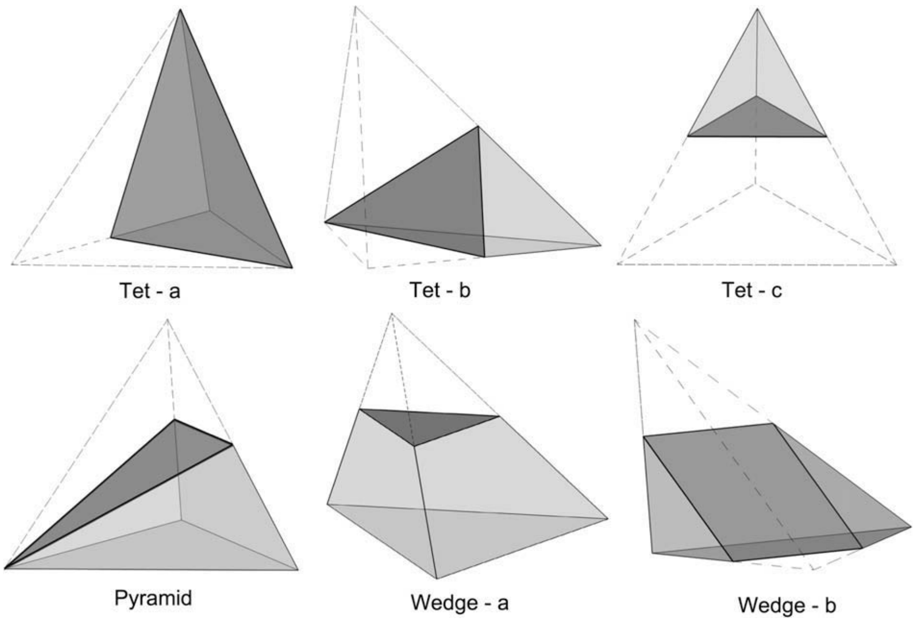

Fig. A.1. Types of plane-tetrahedron intersections; the plane represents the zero level-set of ϕ_h. The cut element can be tetrahedra, pyramids or wedges. Note that Γ_h^E is a triangle except when the cut element is a wedge of type (b).

Γ_h^E. Hence we use the first shape function (N_n) as the level-set function itself. The remaining shape functions (N_E, N_F, N_G) interpolate displacements at points of Γ_h^E. If Γ_h^E is a triangle, N_E, N_F and N_G are the standard P_1 shape functions with degrees of freedom at the vertices of the triangle. In the case of a quadrilateral, we choose as degrees of freedom the values of the function at the three vertices of the quadrilateral that form the triangle with the largest area, and the associated dual shape functions. More precisely, the four shape functions are given by

$$\begin{aligned} N_n(\mathbf{x}) &= (\mathbf{x}-\mathbf{x}_E)\cdot\mathbf{n}, \\ N_E(\mathbf{x}) &= \frac{1}{\Delta EFG}(\mathbf{x}-\mathbf{x}_F)\cdot(\mathbf{x}_F-\mathbf{x}_G)\wedge\mathbf{n}, \\ N_F(\mathbf{x}) &= \frac{1}{\Delta EFG}(\mathbf{x}-\mathbf{x}_G)\cdot(\mathbf{x}_G-\mathbf{x}_E)\wedge\mathbf{n} \quad \text{and} \\ N_G(\mathbf{x}) &= \frac{1}{\Delta EFG}(\mathbf{x}-\mathbf{x}_E)\cdot(\mathbf{x}_E-\mathbf{x}_F)\wedge\mathbf{n}, \end{aligned} \tag{A.1}$$

where E, F and G are the vertices sketched in Fig. A.2, $\Delta EFG = (\mathbf{x}_E-\mathbf{x}_F)\cdot(\mathbf{x}_F-\mathbf{x}_G)\wedge\mathbf{n}$ and $\mathbf{n}$ is the unit normal to the plane of Γ_h^E.

A.3. Quadrature rules

The key ingredient to account for in the construction of quadrature rules for DG-IBM, which makes them different from standard boundary fitting methods, is the correct computation of volumes for elements cut by the boundary.

In DG methods, elements on which the lifting operators may be different from zero need a quadrature rule that can exactly integrate quadratic polynomials – terms appearing as products of components of $\mathbf{D}_{DG}$. With this in mind, standard quadrature rules can be adopted for elements in $\mathscr{R}_h$. Some of these elements are also in $\mathscr{M}_h$, and share a face across which functions may be discontinuous. Quadrature rules for such faces need to be at least second-order accurate. Elements in $\mathscr{Q}_h$ need special quadrature rules for the sections of the element and the faces that are inside the approximate domain. We outline an integration scheme to perform all the necessary integrals exactly and efficiently.

Four Gauss points are needed to integrate quadratic polynomials over tetrahedral domains. Pyramids and wedges are subdivided into two and three tetrahedra respectively, see Fig. A.3, for the purpose of integration. The four-point rule is utilized in each tetrahedron, resulting in a total of 8 integration points for pyramids, and 12 points for wedges. Three Gauss points are used for integration over triangular faces, and four points on quadrilateral ones.

These set of rules were found to be as or more efficient than mapping cut elements to hexahedral elements to evaluate integrals. The Jacobian of such a map increases the order of the polynomials to be integrated, necessitating more quadrature points than being used here.

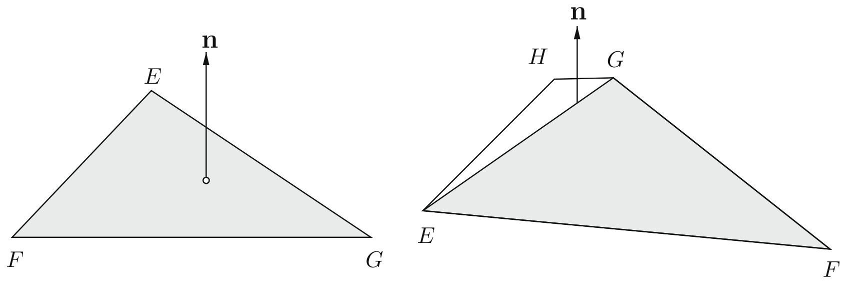

Fig. A.2. For tetrahedral elements cut by the boundary, N_E, N_F and N_G are the standard set of P_1 basis functions, which has the values of a function at E, F and G as degrees of freedom. Shape function N_n is zero on Γ_h^E.

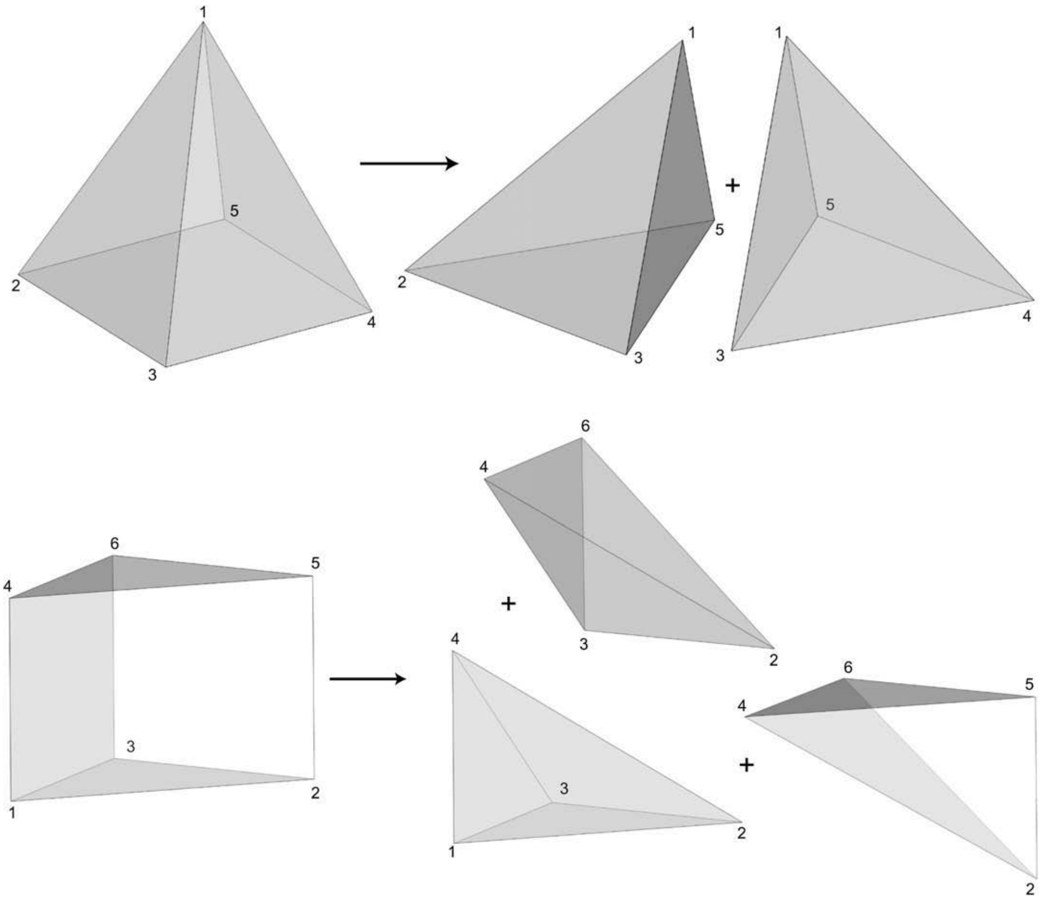

Fig. A.3. For the purpose of integration, pyramids are subdivided into two and wedges into three tetrahedra. A four-point rule is used in each tetrahedron.

References

[1] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (5) (2001) 1749–1779.

[2] I. Babuška, The finite element method with Lagrangian multipliers, Numer. Math. 20 (1973) 179–192.

[3] I. Babuška, The finite element method with penalty, Math. Comput. 27 (1973) 221–228.

[4] I. Babuška, U. Banerjee, J. Osborn, Survey of meshless and generalized finite element methods: a unified approach, Acta Numer. 12 (2003) 1–125.

[5] H. Barbosa, T.J.R. Hughes, The finite element method with Lagrange multipliers on the boundary: circumventing the Babuška-Brezzi condition, Comput. Meth. Appl. Mech. Engrg. 85 (1991) 109–128.

[6] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: an overview and recent developments, Comput. Meth. Appl. Mech. Engrg. 139 (1996) 3–47(45).

[7] T. Belytschko, C. Parimi, N. Moes, N. Sukumar, S. Usui, Structured extended finite element method for solids defined by implicit surfaces, Int. J. Numer. Meth. Engrg. 56 (2002) 609–635.

[8] A. Berger, L^2 error estimates for finite elements with interpolated boundary conditions, Numer. Math. 21 (1973) 345–349.

[9] A. Berger, R. Scott, G. Strang, Approximate boundary conditions in the finite element method, Symposia Mathematica, vol. X, Academic Press, New York, 1972, pp. 757–761.

[10] J.J. Blair, Higher order approximations to the boundary conditions for the finite element method, Math. Comput. 30 (134) (1976) 250–262.

[11] D. Boffi, L. Gastaldi, L. Heltai, C.S. Peskin, On the hyper-elastic formulation of the immersed boundary method, Comput. Meth. Appl. Mech. Engrg. (2007).

[12] J.H. Bramble, J.T. King, A robust finite element method for nonhomogeneous Dirichlet problems in domains with curved boundaries, Math. Comput. 63 (207) (1994) 1–17.

[13] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, Springer, 1994.

[14] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, T.R. Evans, Reconstruction and representation of 3D objects with radial basis functions, in: SIGGRAPH'01: Proceedings of the 28th Annual Conference on Computer graphics and Interactive Techniques, ACM, New York, NY, USA, 2001, pp. 67–76.

[15] P.G. Ciarlet, Mathematical elasticity—volume I: three-dimensional elasticity, Stud. Math. Appl. 20 (1988).

[16] P.G. Ciarlet, P.A. Raviart, Interpolation theory over curved elements, with applications to finite element methods, Comput. Meth. Appl. Mech. Engrg. 1 (1972) 217–249.

[17] R. Codina, J. Baiges, Approximate imposition of boundary conditions in immersed boundary methods, Personal Communication, 2008.

[18] C. De Boor, A Practical Guide to Splines, Springer, 2001.

[19] P. de Casteljau, Courbes et surfaces a poles, SA Andre Citroen, 1959.

[20] C. Engwer, P. Bastian, A discontinuous Galerkin method for simulations in complex domains. Preprint, <http://www/ub.uniheidelberg.de/archiv/5707/>, 2005.

[21] S. Fernández-Méndez, A. Huerta, Imposing essential boundary conditions in mesh-free methods, Comput. Meth. Appl. Mech. Engrg. 193 (2004) 1257–1275.

[22] R. Glowinski, T.W. Pan, T. Hesla, D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 25 (1999) 755–794.

[23] J.P. Gois, A. Nakano, L.G. Nonato, G.C. Buscaglia, Front-tracking with moving-least-square surfaces, J. Comput. Phys. 227 (22) (2008) 9643–9669.

[24] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external flow field, J. Comput. Phys. 105 (1993) 354–366.

[25] K. Höllig, Finite Element Methods with B-Splines, SIAM, 2003.

[26] K. Höllig, C. Apprich, A. Streit, Introduction to the Web-method and its applications, Adv. Comput. Math. 23 (1) (2005) 215–237.

[27] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Meth. Appl. Mech. Engrg. 194 (2005) 4135–4195.

[28] J. Nitsche, Uber ein variations zur losung von dirichlet-problemen bei verwendung von teilraumen die keinen randbedingungen unterworfen sind. In Abhandlungen aus dem Mathematischen Seminaren des Universität Hamburg, 1970, pp. 9–15.

[29] Y. Krongauz, T. Belytschko, Enforcement of essential boundary conditions in meshless approximations using finite elements, Comput. Meth. Appl. Mech. Engrg. 131 (1996) 133–145.

[30] M.C. Lai, C. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.

[31] M. Lenoir, Optimal isoparametric finite elements and error estimates for domains involving curved boundaries, SIAM J. Numer. Anal. 23 (3) (1986) 562–580.

[32] A. Lew, G. Buscaglia, A discontiuous-Galerkin based immersed boundary method, Int. J. Numer. Meth. Engrg. 76 (4) (2008) 427–454.

[33] A. Lew, P. Neff, D. Sulsky, M. Ortiz, Optimal BV estimates for a discontinuous Galerkin method for linear elasticity, Appl. Math. Res. Express 2004 (3) (2004) 73–106.

[34] A. Liu, B. Joe, Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision, Math. Comput. 65 (215) (1996) 1183–1200.

[35] X.D. Liu, C.T. Sideris, Convergence of the ghost fluid method for elliptic equations with interfaces, Math. Comput. 72 (244) (2003) 1731–1746.

[36] R. Löhner, J.R. Cebral, F.E. Camelli, S. Appanaboyina, J.D. Baum, E.L. Mestreau, O.A. Soto, Adaptive embedded and immersed unstructured grid techniques, Comput. Meth. Appl. Mech. Engrg. 197 (25–28) (2008) 2173–2197.

[37] J.E. Marsden, T.J.R. Hughes, Mathematical Foundations of Elasticity, Dover, Mineola, New York, 1994.

[38] S. Mauch. A fast algorithm for computing the closest point and distance transform. Preprint, <http://www.acm.caltech.edu/seanm/software/cpt/cpt.html>, 14, 2001.

[39] S. Mauch, Efficient Algorithms for Solving Static Hamilton–Jacobi Equations, Ph.D. Thesis, California Institute of Technology, 2003.

[40] R. Mittal, G. Iaccarino, Immersed boundary methods, Ann. Rev. Fluid Mech. 37 (2005) 239–261.

[41] N. Moës, E. Béchet, M. Tourbier, Imposing Dirichlet boundary conditions in the extended finite element method, Int. J. Numer. Meth. Engrg. 67 (2006) 1641–1669.

[42] F. Mut, G.C. Buscaglia, E.A. Dari, New mass-conserving algorithm for level set redistancing on unstructured meshes, J. Appl. Mech. 73 (2006) 1011.

[43] C. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.

[44] V.L. Rvachev, T.I. Sheiko, V. Shapiro, I. Tsukanov, On completeness of RFM solution structures, Comput. Mech. 25 (2000) 305–316.

[45] V.L. Rvachev, T.I. Sheiko, V. Shapiro, I. Tsukanov, Transfinite interpolation over implicitly defined sets, Comput. Mech. 18 (2001) 195–220.

[46] R. Scott, Finite Element Techniques for Curved Boundaries, Ph.D. Thesis, MIT, 1973.

[47] R. Scott, Interpolated boundary conditions in the finite element method, SIAM J. Numer. Anal 12 (1975) 404–427.

[48] J. Sethian, Structural boundary design via level set and immersed interface methods, J. Comput. Phys. (0021-9991) 163 (2) (2000) 489–528.

[49] R. Sevilla, S. Fernández-Méndez, A. Huerta, NURBS-enhanced finite element method (NEFEM), Int. J. Numer. Meth. Engrg. 76 (1) (2008) 56–83.

[50] V. Shapiro, Real functions for representations of rigid solids, Comput. Aided Geom. Design 11 (1994).

[51] R. Stenberg, On some techniques for approximating boundary conditions in the finite element method, J. Computat. Appl. Math. 63 (1995) 139–148.

[52] G. Strang, G.J. Fix, An Analysis of the Finite Element Method, second ed., Wellesley-Cambridge Press, Wellesley, MA, 2008.

[53] N. Sukumar, T. Moran, B. Belytschko, The natural element method in solid mechanics, Int. J. Numer. Meth. Engrg. 43 (1998) 839–887.

[54] A. Ten Eyck, F. Celiker, A. Lew, Adaptive stabilization of discontinuous Galerkin methods for nonlinear elasticity: analytical estimates, Comput. Meth. Appl. Mech. Engrg. 197 (2008) 2989–3000.

[55] A. Ten Eyck, F. Celiker, A. Lew, Adaptive stabilization of discontinuous Galerkin methods for nonlinear elasticity: motivation, formulation, and numerical examples, Comput. Meth. Appl. Mech. Engrg. 197 (2008) 3605–3622.

[56] A. TenEyck, A. Lew, Discontinuous Galerkin methods for nonlinear elasticity, Int. J. Numer. Meth. Engrg. 67 (2006) 1204–1243.

[57] V. Thomée, Polygonal domain approximation in Dirichlet's problem, J. Inst. Math. Appl. 11 (1973) 33–44.

[58] Y.R. Tsai, Rapid and accurate computation of the distance function using grids, jcp 178 (1) (2002) 175–195.

[59] Y.H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2) (2003) 593–623.

[60] I. Tsukanov, V. Shapiro, Implicit functions with guaranteed differential properties, in: Proceedings of the Fifth ACM Symposium on Solid Modeling and Applications, Ann Arbor, MI, June 1999, pp. 258–269.

[61] G. Turk, J.F. O'Brien, Variational implicit surfaces. Technical report, Georgia Institute of Technology, May 1999.

[62] L. Zhang, A. Gerstenberger, X. Wang, W.K. Liu, Immersed finite element method, Comput. Meth. Appl. Mech. Engrg. 193 (21–22) (2004) 2051–2067.

[63] M. Zlámal, Curved elements in the finite element method. I, SIAM J. Numer. Anal. 10 (1) (1973) 229–240.

[64] M. Zlamal, Curved elements in the finite element method. II, SIAM J. Numer. Anal. 11 (2) (1973) 347–362.

[65] M. Zlámal, The finite element method in domains with curved boundaries, Int. J. Numer. Meth. Engrg. 5 (April) (1973) 367–373.