A Discontinuous-Galerkin based Immersed Boundary Method with Nonhomogeneous Boundary Conditions and its Application to Elasticity

Ramsharan Rangarajan, Adrián J. Lew, Gustavo C. Buscaglia [a,b,c]

[a] *Department of Mechanical Engineering, Stanford University, CA 94305-4040, USA*
rram@stanford.edu

[b] *Department of Mechanical Engineering, Stanford University, CA 94305-4040, USA*
lewa@stanford.edu

[c] *Centro Atómico Bariloche and Instituto Balseiro 8400 Bariloche, Argentina*
gustavo@cab.cnea.gov.ar

Abstract

This article proposes a discontinuous Galerkin based immersed boundary method(DG-IBM) with an extension method for Dirichlet and Neumann boundary conditions(bcs) to an approximate boundary. The resulting numerical scheme does not require boundary fitting meshes, and elegantly avoids boundary locking by switching the elements cut by the boundary to a discontinuous Galerkin approximation. The extension methods for bcs are quite general- they can be used for problems in two and three dimensions, with non-homogeneous bcs and allows for strict enforcement of Dirichlet boundary conditions. Extensive numerical experiments with two and three dimensional elasticity problems shows that the proposed methods leads to optimal convergence rates in the L_2 norm.

Key words: Dirichlet boundary conditions, Neumann boundary conditions, Extension methods, Discontinuous Galerkin method, Cartesian grids, Elasticity, Immersed boundary.

1 Introduction

The motivation for this paper is to create a numerical method that can be used to solve boundary value problems on complex domains using a sequence of approximate domains. The method is tailored to satisfy what we call an *asymptotic consistency condition*, which ensures that the numerical solution computed converges to the exact solution with the optimal rate. The ingredients of DG-IBM, namely approximating the exact domain of the problem, building the solution space and defining the boundary conditions on the resulting approximate boundary- are constructed keeping in mind the conditions necessary to satisfy such a condition. While methods to solve boundary value problems on curved domains has been extensively studied in the literature, the approach we have adopted here is, to the best of our knowledge, new, elegant and suitable for academic as well as practical applications.

DG-IBM is an immersed boundary method that approximates the domain of the problem using a mesh over an encompassing domain that is easy to discretize. As highlighted in [27], a key ingredient of DG-IBM is the use of a discontinuous Galerkin approximation on elements that are intersected by the boundary in order to overcome a phenomenon called "boundary locking". This permits enforcing Dirichlet boundary conditions strongly. Like with other immersed boundary methods, the question arises about what to prescribe as boundary conditions on the resulting approximate boundary. We address this question using a numerical extension method which can be used for non-homogeneous Dirichlet and Neumann boundary conditions in two and three dimensional problems. A nice feature of the numerical extension method is that it does not rely on the boundary conditions being specified by analytical expressions; for instance, this would be the case when boundary conditions arise from companion numerical simulations.

Handling domains with curved boundaries using the finite element method, though extensively studied, continues to be a significant challenge. We briefly review the use of isoparametric elements, immersed boundary methods and meshless methods to handle such problems focusing more on the ideas and challenges in these methods rather than on details. This will serve as a good point of departure before describing the DG-IBM formulation. While the review is by no means exhaustive, the ideas and references mentioned is expected to guide the reader to the relevant literature.

A simple and widely adopted approach for finite element analysis on curved domains is using an inscribed polygon(or polyhedron) to approximate the problem domain. Imposing boundary conditions is also easy- prescribed values are imposed at the nodes that lie on the exact boundary. However, such an approximation is known to have severe limitations, especially with regard

to the order of convergence of the numerical solution. The solution seems to lie in better approximating the domain and the boundary. References [47–49,15,36,37,9–11,26] provide illuminating details in this direction. The idea has been to map conventional straight edged-elements to curved/pie-shaped ones, for instance using isoparametric elements, that conform to the exact boundary yielding an exact description of the domain. A short review of using curved elements and interpolated boundary conditions is presented in the next section. Some of these ideas will serve as a benchmark against which to compare the methods proposed in this paper. A more recently introduced method called the *Isogeometric analysis* of Hughes et al. in [23] is a new effort in a similar direction. The emphasis is once again, on accurately representing the boundary, which in this case is achieved using basis functions generated from NURBS.

Immersed boundary methods, in contrast to the standard finite element method, approximate the complex domain using a sequence of simpler meshes instead of attempting to represent it exactly. First proposed by Peskin in [33], it has been particularly useful in the context of fluid-structure interaction problems in which using Cartesian grids simplifies computations significantly. Most IB methods are distinguished by how they address two main issues. The first one is determining the domain of the problem in a possibly nonconforming grid. Almost always, the domain is only determined approximately. As a result, the boundary of the problem is now only an approximation of the exact one. The second issue then deals with prescribing boundary conditions on the approximate boundary. The boundary conditions imposed often leads to suboptimal convergence of the solution because of a phenomenon called boundary locking. Intuitively, this results from the solution space being over-constrained due to imposing Dirichlet boundary conditions strongly. There are two simple ways to resolve this issue- either impose boundary conditions weakly or enrich the solution space, at least near the boundary. Methods which use a forcing approach, either continuous [21,33,25,22] or discrete, and the ghost-fluid methods [16,18,28] belong to the former category. Numerical accuracy when using a forcing method relies on a good choice of distribution functions to regularize the Dirac delta functions on the boundary, which arise naturally in the formulation. This idea can be interpreted as using a penalty method to prescribe values on the boundary, and expectedly, it can result in a stiff system of equations. The ghost-cell or ghost-fluid method on the other hand uses interpolations in cells outside the domain to implicitly incorporate boundary conditions. Methods that use a local mesh refinement near the approximate boundary as well as DG-IBM fall into the latter category, though for different reasons. Local remeshing [17] increases the number of nodes on the boundary, possibly making the approximate boundary conform with the exact one, and hence avoids over constraining the system of equations when strictly imposing Dirichlet boundary conditions. DG-IBM on the other hand, avoids adding/moving nodes on the boundary, which is tedious and difficult

to code, especially in 3D problems. Instead, the solution space is enriched by permitting discontinuities in the solution where it is needed most- near the boundary.

The previous two numerical methods relied on representing complex domains using meshes. A significant chunk of effort in these numerical methods is directed towards using a mesh that is optimal for a given geometry- balancing accuracy of the domain representation and the computational resources available. Meshless methods[5,7] bypass this arduous task by relying on the use of points to represent the domain, which is far easier than creating a boundary fitting mesh for the domain (as in conventional finite element methods) or approximating it in a non-conforming mesh (as done in immersed boundary methods). Meshless methods are thus an attractive alternative for problems on complex domains, but as we highlight here, there are other issues to worry about. While the domain is easily represented, the choice of basis functions to create the solution space gives rise to computational difficulties. Basis functions no longer have compact support, unlike the "delta" property of the basis functions used in conventional finite element methods. This results in a dense stiffness matrix which makes handling large problems computationally expensive. The use of basis functions with non compact support poses difficulties in imposing boundary conditions as well- which are imposed as constraints rather than directly. Lagrange multipliers[3,6,20] and penalty methods[4] are commonly used for this purpose, though they come with their own share of troubles. It is a tricky business to select the right space of Lagrange multipliers[31]- while a degenerate system of equations may result from using too large a space, the other extreme may not impose boundary conditions well. On the other hand, choosing the value of the penalty parameter is an obvious problem while imposing boundary conditions using penalty. Often, numerous trial simulations are necessary to even determine an optimal range of the penalty parameter required; and the price of using penalty us further inflated by the increasing stiffness of the system of equations. Nitsche's method[24,5,39], which uses a modified weak form of the problem to impose boundary conditions, is used both in meshless as well as standard finite element methods. Since much of the problems in imposing boundary conditions arises because of the non compact support of the basis functions (especially near the boundary), it makes sense to locally modify the interpolation scheme near the boundary as done in [46]. The result is a meshless method that has finite elements on the boundary (which have the "delta" property), making it far simpler to impose boundary conditions. Reference [19] provides a more detailed survey of methods to impose boundary conditions in meshless methods. The interested reader is also directed to reference [32] for an account of the natural element method that shares many features with meshless methods.

In the following section, a more detailed review of using isoparametric elements for curved domains is presented. As mentioned before, DG-IBM is

tailored to satisfy an asymptotic consistency condition, which guarantees the convergence properties of the method. The details of this condition and a simple proof showing the convergence of the numerical solution computed using a consistent method is presented in section 3. The most important part of the paper is section 4, which describes the DG-IBM formulation- it describes how the domain is approximated in a non-conforming mesh, the construction of the solution space and the numerical extension method for extending the given boundary conditions to the approximate boundary. The formulation is explicitly written down for the linear and nonlinear elasticity problem. While the rigorous proofs to show that the method constructed here is asymptotically consistent will be presented in a companion paper, in this article, we focus on learning from examples. In section 5, using two and three dimensional examples in elasticity, we investigate the convergence properties of the proposed formulation for various combinations of nonhomogeneous Dirichlet and Neumann boundary conditions. Also presented are two novel applications in nonlinear elasticity- simulation of the compression of a femur bone and the deformation of an image based geometry.

2 The finite element method for domains with curved boundaries

A simple and well studied second order boundary value problem is the Dirichlet problem on a two dimensional domain Ω – the Poisson equation with homogeneous Dirichlet boundary conditions on the smooth boundary Γ of Ω. The simplest approach with the finite element method is to use a polygonal approximation(Ω_h) of the domain with vertices lying on Γ. A triangulation of Ω_h, serves as the mesh for the standard finite element method using, for example, linear triangle elements (see figure 1). Here, h is a mesh parameter such that as $h \to 0$, Ω_h and Γ_h approach Ω and Γ respectively. The numerical solution is determined by imposing boundary conditions at the nodes that lie on Γ and solving a set of linear equations. In the skin region $\Omega \setminus \Omega_h$, the solution is extended by zero. The numerical solution is known to converge to the exact solution with $\mathcal{O}(h^2)$ in the L_2 norm on Ω. While the solution thus computed satisfies the boundary condition on Γ (in addition to being zero on Γ_h), it does not satisfy the Poisson equation everywhere in Ω- particularly in the skin region. This was studied by Strang in his article[40] aptly titled *"Change in solution due to change in domain"* and by Thomée in [42]. Their analysis shows that the $\mathcal{O}(h^2)$ convergence is to be expected using linear elements. But more crucially, they showed that the rate of convergence does not improve with using higher order elements. That Γ_h was always at a distance that is $\mathcal{O}(h^2)$ away from Γ is shown to be the limiting factor.

Use of Curved Elements: The solution then lies in in better approximating the boundary. This can be achieved by using meshes that conform to the exact

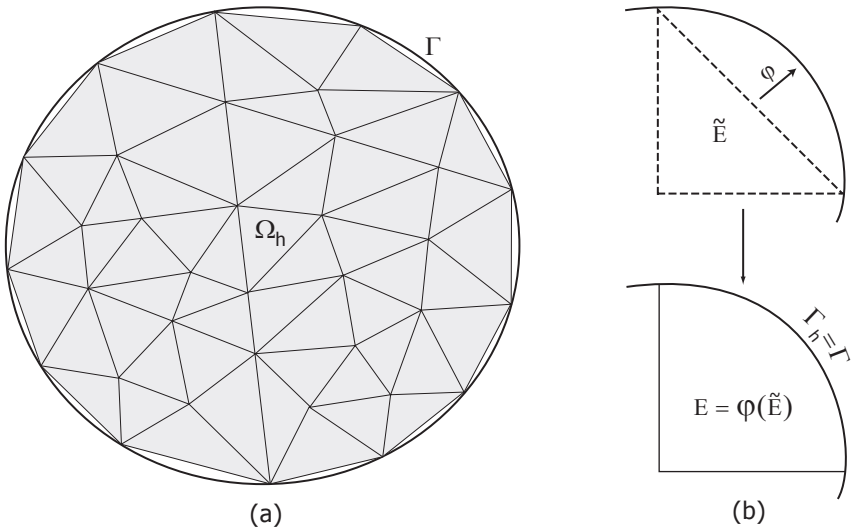


Fig. 1. Shown on the left is a polygonal approximation (Ω_h) of a circular domain. The boundary Γ is approximated by straight segments. The distance between Γ_h and Γ is $\mathcal{O}(h^2)$. On the right, the idea of using *curved* elements is depicted. The element $\tilde{E}$ with straight edges is transformed into a pie-shaped element E whose curved edge now conforms with the exact boundary.

boundary. This is depicted in figure 1 where triangles along the boundary are pie-shaped with curved edges; the idea is the same for three dimensional domains. A systematic way to do this and still retain the simplicity of using polygonal meshes is as follows. As described in the previous paragraph, a triangulation $\tilde{\mathcal{T}}_h$ is constructed over a polygonal/polyhedral approximation $\tilde{\Omega}_h$ of Ω. A mesh $\mathcal{T}_h$ that conforms to the boundary Γ is obtained by transforming elements of $\tilde{\mathcal{T}}_h$ that have edges(faces) whose vertices lie on Γ, into curved elements. The remaining edges/faces and elements remain unaltered (straight). As described in [26], the mapping from straight to curved elements can be specified using the charts for the boundary Γ, assuming that there are a finite number of them. By using such a map, the problem is thus mapped to a polygonal domain whose discretization is the mesh $\tilde{\mathcal{T}}_h$.

When the map from straight to curved elements has the same order as the element (i.e. the degree of the polynomial basis functions), the elements are called isoparametric. Since the convergence rate of the numerical solution is limited by the smaller of the order of the element and the order of approximation of the boundary, isoparametric elements are a common choice for general curved domains.

Interpolated Boundary Conditions: Both homogeneous and non-homogeneous Dirichlet boundary conditions are easily *imposed* because the nodal basis functions still satisfy the delta property- prescribed values are simply imposed at nodes that lie on Γ. However, the computed solution may not equal the prescribed values everywhere on Γ, and in general, agreement may be only at a

finite number of points. This translates to using interpolated boundary conditions; i.e., imposing as boundary condition, a function in the finite element space which equals the prescribed function at nodes that lie on Γ. The importance of such an approximation and the consequent error estimates is discussed in [37].

An alternative method for imposing non-homogeneous Dirichlet boundary conditions is given by Bramble and King in [12]. The boundary conditions on Γ_h are defined as a projection of the boundary condition defined in Γ. In conjunction with a multigrid approach, optimal convergence rates are achieved. In Reference [42], a fourth order accurate multigrid scheme was devised using a polygonal approximation with linear elements, in contrast to the second order accuracy mentioned previously. Though such multigrid approaches can be used to improve accuracy of numerical schemes, we will not adopt it in this paper.

3 Designing the method- Asymptotic Consistency

Before we outline our method, we pause for a brief discussion that will help constructing it. Our objective is to numerically approximate the exact solution of a boundary value problem on a sequence of easy to discretize domains, none of which might be the exact one. However, by choosing the approximate domains and function spaces wisely, we will attempt to get closer to the exact solution, and at the optimal rate. While the actual approximation schemes that we use are described in later sections, we discuss some ideas that will aid in constructing them.

Consider the weak form of a boundary value problem to find a function $u : \Omega \mapsto \mathbb{R}$ in a Banach space $(V, \|\cdot\|)$, such that

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

along with Dirichlet and Neumann boundary conditions

$$u = \bar{u} \quad \text{on } \Gamma^d \text{ and} \tag{2}$$
$$\nabla u \cdot \mathbf{n} = g \quad \text{on } \Gamma^\tau, \tag{3}$$

where $V_\partial = \{v \in V : v|_{\Gamma^d} = 0\}$, a is a bilinear form and F is a linear operator. We note that the Neumann boundary condition is embedded in the definition of F. We assume that $\Gamma^d \cup \Gamma^\tau = \partial\Omega$, $\Gamma^d \cap \Gamma^\tau = \emptyset$ and that this problem has a unique solution.

While the conventional finite element methods (like described in the previous section) attempt to solve this problem by representing the domain as accu-

rately as possible, we tread a slightly different path.
We consider a sequence of domains, Ω_h, parameterized by h (for example the mesh size), to approximate Ω such that as $h \downarrow 0$, Ω_h is a better and better representation of Ω. Indeed, for a given small value of h, we would like to quantify how well Ω_h approximates Ω. Such an approximation can be quite handy for complex geometries, for not only can an exact representation be too expensive, it may also be unncessary at times. For example, the image based geometries and a femur bone model presented later in the paper, are already approximate representations of the exact domain and attempting to represent it accurately may not be a fruitful exercise.

Before describing the discrete problem, we construct the necessary function spaces. Let V_h be a finite dimensional space of functions on $(\Omega \cup \Omega_h)$ and $V_{h,\Omega} = \{\chi_\Omega v_h : v_h \in V_h\}$, χ_Ω being the characteristic function of the set Ω. We define a norm $\|\!| \cdot |\!\|_h$ (possibly depending on h) for $(V + V_{h,\Omega})$ such that $\forall v \in V$, $\|\!| v |\!\|_h = \|v\|$. We assume that we can define a norm $\|\!| \cdot |\!\|_{V_h}$ on V_h such that $\forall v_h \in V_h$, $\|\!| v_h |\!\|_{V_h} = \|\!| \chi_\Omega v_h |\!\|_h$. This naturally constructs a linear restriction operator $(\cdot)^r : V_h \mapsto V_{h,\Omega}$ serving as a bijection between V_h and $V_{h,\Omega}$.

Additionally, we assume that we can define two continuous operators- an *approximation* operator $\Pi_h : V \mapsto V_h$ and an *extension* operator $(\cdot)^e$ on V (c.f. [1]) that extends the definition of functions in V to all of $\mathbb{R}^n$. By requiring that for $v \in V$, $v^e|_\Omega = v$, the extension operator enables us to extend the definition of functions on Ω to ones on $\Omega \cup \Omega_h$. For any $v \in V$, we require that the approximation operator satisfy

$$\|\!| v - (\Pi_h v)^r |\!\|_h < Ch|v|_V, \tag{4}$$

where $|\cdot|_V$ is an appropriate seminorm on V.

The discrete problem is to find $u_h \in (V_h^\partial + \Pi_h \bar{u})$ such that

$$a_h(u_h, v_h) = F_h(v_h) \quad \forall v_h \in V_h^\partial, \tag{5}$$

where $V_h^\partial = \{v_h \in V_h : v_h|_{\Gamma_h^d} = 0\}$ is the space of admissible test functions, and $a_h : [V^e|_{\Omega_h} + V_h]^2 \mapsto \mathbb{R}$, $F_h : V_h \mapsto \mathbb{R}$ are continuous and satisfy an *asymptotic consistency* condition

$$|a_h(u^e, v_h) - F_h(v_h)| \leq Ch|u|_V \|\!| v_h |\!\|_{V_h} \quad \forall\ v_h \in V_h^\partial. \tag{6}$$

Further, we assume that a_h is coercive in V_h, with coercivity constant γ independent of h. The importance of the consistency condition cannot be over emphasized. As shown below, it guarantees the approximation properties of the numerical solution. Embedded in this condition are details of how well Ω_h approximates Ω, the choice of the subsets of $\partial(\Omega \cup \Omega_h)$ on which to impose Dirichlet and Neumann boundary conditions as well as what to impose.

The Dirichlet boundary condition on Γ_h^d is strictly imposed as $\Pi_h \bar{u}$ while the Neumann boundary condition is implicitly defined in the choice of F_h. Notice that there is considerable freedom in choosing the Dirichlet and Neumann boundary conditions- the only requirement to be satisfied is the consistency condition. For example, while deciding the Neumann boundary conditions, we observe that for any $L_h \in V_h^*$ with $\|L_h\|_{V_h^*} < Ch$, $\tilde{F}_h = F_h + L_h$ would still satisfy the consistency condition.

With the consistency condition as above, we claim that $\|\!|u - u_h^r|\!\|_h < Ch(|u|_V + |u - \bar{u}|_V)$. Let $w = u - \bar{u}$ and $w_h = u_h - \Pi_h \bar{u}$.

$$\begin{aligned}
\|\!|u - u_h^r|\!\|_h &= \|\!|(u - \bar{u}) - (u_h^r - (\Pi_h \bar{u})^r) + (\bar{u} - (\Pi_h \bar{u})^r)|\!\|_h, && (7)\\
&\leq \|\!|w - w_h^r|\!\|_h + \|\!|\bar{u} - (\Pi_h \bar{u})^r|\!\|_h, && (8)\\
&\leq \|\!|w - (\Pi_h w)^r|\!\|_h + \|\!|(\Pi_h w)^r - w_h^r|\!\|_h + \|\!|\bar{u} - (\Pi_h \bar{u})^r|\!\|_h. && (9)
\end{aligned}$$

We investigate each term in the above inequality. In what follows, C will be a positive constant independent of h and can change from one line to the next. By definition of Π_h,

$$\begin{aligned}
\|\!|w - (\Pi_h w)^r|\!\|_h &< Ch|w|_V \quad \text{and} && (10)\\
\|\!|\bar{u} - (\Pi_h \bar{u})^r|\!\|_h &< Ch|\bar{u}|_V. && (11)
\end{aligned}$$

We also have

$$\begin{aligned}
\|\!|(\Pi_h w)^r - w_h^r|\!\|_h &= \|\!|(\Pi_h w - w_h)^r|\!\|_h, && (12)\\
&< C\|\!|\Pi_h w - w_h|\!\|_{V_h} \quad (\text{since } (\cdot)^r \text{ bounded}). && (13)
\end{aligned}$$

We can bound $\|\!|\Pi_h w - w_h|\!\|_{V_h}$ using the asymptotic consistency condition as follows:

$$\begin{aligned}
|a_h(u^e, v_h) - F_h(v_h)| &< Ch|u|_V \|\!|v_h|\!\|_{V_h}, && (14)\\
\text{i.e., } |a_h(w^e + \bar{u}^e, v_h) - a_h(w_h + \Pi_h \bar{u}, v_h)| &< Ch|u|_V \|\!|v_h|\!\|_{V_h}. && (15)
\end{aligned}$$

Hence,

$$\begin{aligned}
|a_h(w^e - w_h, v_h)| &< Ch|u|_V \|\!|v_h|\!\|_{V_h} + |a_h(\bar{u}^e - \Pi_h \bar{u}, v_h)|,\\
&\leq Ch|u|_V \|\!|v_h|\!\|_{V_h} + M\|\!|\bar{u}^e - \Pi_h \bar{u}|\!\|_{V_h} \|\!|v_h|\!\|_{V_h}, \quad (\text{continuity of } a_h)\\
&\leq Ch|u|_V \|\!|v_h|\!\|_{V_h} + MCh|u|_V \|\!|v_h|\!\|_{V_h},\\
&\leq Ch|u|_V \|\!|v_h|\!\|_{V_h}, && (16)
\end{aligned}$$

where the constant C can change from one line to the next. From the coercivity

of a_h on V_h, we have

$$\begin{aligned}\gamma\|\Pi_h w - w_h\|_{V_h}^2 &\leq a_h(\Pi_h w - w_h, \Pi_h w - w_h) \\ &= a_h(\Pi_h w - w^e, \Pi_h w - w_h) + a_h(w^e - w_h, \Pi_h w - w_h) \\ &\leq M\|\Pi_h w - w^e\|_{V_h}\|\Pi_h w - w_h\|_{V_h} + Ch|w|_V\|\Pi_h w - w_h\|_{V_h}.\end{aligned} \tag{17}$$

Thus, we have

$$\begin{aligned}&\gamma\|\Pi_h w - w_h\|_{V_h} \leq M\|\Pi_h w - w^e\|_{V_h} + Ch|w|_V \\ \Rightarrow \quad &\|\Pi_h w - w_h\|_{V_h} \leq Ch|w|_V \\ \Rightarrow \quad &\|(\Pi_h w)^r - (w_h)^r\|_h \leq Ch|w|_V.\end{aligned} \tag{18}$$

And hence,

$$\begin{aligned}\|u - u_h^r\|_h &\leq \|w - (\Pi_h w)^r\|_h + \|(\Pi_h w)^r - w_h^r\|_h + \|\bar{u} - (\Pi_h \bar{u})^r\|_h \\ &\leq Ch|w|_V + Ch|w|_V + Ch|\bar{u}|_V \qquad \text{from eq.(10,11,18)} \\ &\leq 2Ch(|u|_V + |u - \bar{u}|_V)\end{aligned} \tag{19}$$

To summarize in a nutshell, given a problem of finding u on Ω, we can carefully define an extended problem on $\Omega \cup \Omega_h$ and solve for u_h. The restriction of u_h on Ω, namely u_h^r is a good approximation of u.

4 DG-IBM Formulation with extended Boundary Conditions

In this section, we outline the DG-IBM formulation with linear elements to solve for an unknown scalar field, keeping in line with the consistency condition given in the previous section and with the ideas introduced in [27]. However, the framework presented next is a little different from what was described above. For example, V_h will be a space of functions defined on Ω_h instead of $\Omega_h \cup \Omega_h$. Consequently, the approximate solution is determined only on the domain Ω_h. In a later article, we will rigorously show when and how the formulation presented here is equivalent to the one described above. This formulation is only a specific example from the class of asymptotically consistent methods- but as will be evident from the numerical examples, it is a particularly simple and useful one.

4.1 Meshes

Let Ω be an open domain in $\mathbb{R}^n$ with, for simplicity, a smooth boundary $\Gamma = \partial\Omega$ (e.g., C^0 and piecewise C^1). We assume that $\mathcal{B} \supseteq \Omega$, is an open domain,

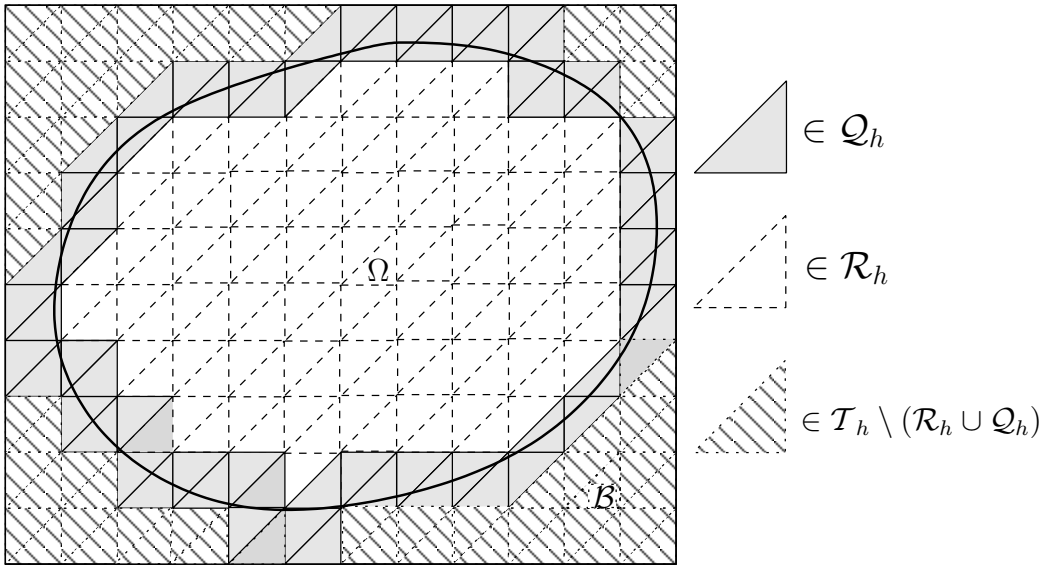


Fig. 2. The problem domain Ω is immersed in an encompassing and easy to discretize domain $\mathcal{B}$. The triangulation on $\mathcal{B}$, which is a rectangle here, is denoted by $\mathcal{T}_h$. The elements of $\mathcal{T}_h$ cut by the boundary Γ are denoted by $\mathcal{Q}_h$ and the elements that lie entirely in Ω are denoted by $\mathcal{R}_h$.

conveniently chosen such that it is trivial to construct a family of quasiuniform (see [13]) conforming meshes $\mathcal{T}_h$ over it. The parameter h is the mesh size. For definiteness, we shall assume that $\mathcal{T}_h$ is a triangulation of n-simplices. An element E in $\mathcal{T}_h$ is a closed $n-$simplex with an orientable boundary Γ^E and unit outward normal $\mathbf{N}^E$. Most ideas described here are directly applicable to other types of discretizations, such as quads or hexahedra.

As depicted in figure 2, we define two submeshes or collections of elements

$$\mathcal{R}_h = \left\{ E \text{ element } \in \mathcal{T}_h \colon \overset{o}{E} \subset \Omega \right\} \tag{20}$$

$$\mathcal{Q}_h = \left\{ E \text{ element } \in \mathcal{T}_h \colon \overset{o}{E} \cap \Gamma \neq \emptyset \right\}, \tag{21}$$

which are the sets of elements that are completely contained in Ω and the set of elements that are intersected by the boundary.

4.2 Approximation of the domain

Let $\phi \colon \mathcal{B} \mapsto \mathbb{R}$ be a $H^1(\mathcal{B})$ function such that $\phi < 0$ in Ω, $\phi = 0$ on Γ and $\phi > 0$ in $\mathcal{B} \setminus \overline{\Omega}$. The boundary of Ω is thus the zero level-set of ϕ, and so ϕ is termed the level-set function. Next, let

$$\Phi_h = \{ v_h \in H^1(\mathcal{B}) : v_h \mid_E \in P_1(E) \quad \forall E \in \mathcal{T}_h \}, \tag{22}$$

where, more generally, $P_k(E)$ denotes the space of polynomials of degree less or equal than $k \in \mathbb{N}$. Let $\phi_h \in \Phi_h$ be the function that interpolates ϕ at every node in $\mathcal{T}_h$. The approximate domain Ω_h is defined as

$$\Omega_h = \{x \in \mathcal{B} \colon \phi_h(x) < 0\}, \tag{23}$$

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 is contained within elements in $\mathcal{Q}_h$. The resulting approximate boundary is then simple to compute– it is composed of straight segments (piecewise linears) in two dimensions and planar polygons in the three dimensional case. Thus Γ_h provides nearly exact approximations of polyhedral boundaries and $\mathcal{O}(h^2)$ approximation of smooth curved boundaries. Additionally, efficient quadrature rules over the resulting sections are easy to construct.

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

$$\phi(x) = \begin{cases} -d(x,\Gamma) & \text{if } \ x \in \Omega, \\ d(x,\Gamma) & \text{if } \ x \notin \Omega. \end{cases} \tag{24}$$

where

$$d(x,\Gamma) = \inf_{y \in \Gamma} \|x - y\|, \tag{25}$$

$\|\cdot\|$ being the Euclidean distance in $\mathbb{R}^n$. As defined, the function ϕ has the property that it is positive in the interior of Ω, identically zero on Γ and negative in $\mathcal{B} \setminus \bar{\Omega}$.

Fast techniques to compute distance functions, particularly on Cartesian grids are discussed in [29,30,43]. Alternative methods such as R- functions ([38,34,35,44]) which define Ω implicitly using boolean operations over easy to define primitives or *surface fitting* methods [45,14,8] which construct a smooth interpolant of the boundary, are equally suitable.

Note 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

$$\phi_h(x_a) = \begin{cases} \phi(x_a) & \text{if } |\phi(x_a)| > \frac{C_{\mathrm{TOL}}}{L} h^2 \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

where C_{TOL} is a small constant and L is a characteristic dimension of Ω. In our examples $C_{\mathrm{TOL}} = 10^{-6}$ and $L = 1$. 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 scales with h^2, so as to preserve the quadratic convergence of the method.

4.3 Approximation of functions

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

$$V_h = V_R \times V_Q = \left\{ v_h \in L^2(\mathcal{Q}_h \cup \mathcal{R}_h) \colon v_h|_{\mathcal{Q}_h} \in V_Q,\ v_h|_{\mathcal{R}_h} \in V_R \right\}, \tag{27}$$

where

$$V_R = \{ v_h \in H^1(\mathcal{R}_h) :\ v_h|_E \in P_k(E)\ \ \forall E \in \mathcal{R}_h \} \tag{28}$$
$$V_Q = \{ v_h \in L^2(\mathcal{Q}_h) :\ v_h|_E \in P_k(E)\ \ \forall E \in \mathcal{Q}_h \}. \tag{29}$$

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

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 $\mathcal{M}_h$ be the submesh of $\mathcal{T}_h$ containing all the elements across whose boundaries $v_h \in V_h$ may have discontinuities. Consequently, $\mathcal{M}_h$ contains all elements of $\mathcal{Q}_h$ and those elements of $\mathcal{R}_h$ that share at least one face with an element in $\mathcal{Q}_h$. Following the framework introduced by Arnold et al. in [2], the discontinuous Galerkin method studied here utilizes the Bassi-Rebay numerical fluxes to approximate the space of derivatives $\mathbf{W}_h^d$, of functions that may be discontinuous across element boundaries.

$$\mathbf{W}_h^d = \{ \mathbf{w}_h \in L^2(\mathcal{M}_h)^n ; \mathbf{w}_h|_E \in (P_d(E))^n\ \forall E \in \mathcal{M}_h \}, \tag{30}$$

where $d \in \mathbb{N}$. Functions in $\mathbf{W}_h^d$ are vector valued with n components, n being the number of partial derivatives of functions in V_h.

For elements $E \in \mathcal{M}_h$, define the jump $[\![\cdot]\!]$ and average $\{\cdot\}$ across a face $e \notin \Gamma_h$ for functions in V_h and $\mathbf{W}_h^d$ as

$$[\![v_h]\!] = v_h^- - v_h^+, \qquad [\![\mathbf{w}_h]\!] = \mathbf{w}_h^- - \mathbf{w}_h^+, \tag{31}$$
$$\{v_h\} = \frac{1}{2}\left(v_h^- + v_h^+\right), \qquad \{\mathbf{w}_h\} = \frac{1}{2}\left(\mathbf{w}_h^- + \mathbf{w}_h^+\right), \tag{32}$$

where the superscripts $+$ and $-$ correspond to evaluating functions on either side of e, $+$ being on the side of the outward normal to e. For faces $e \in \Gamma_h$,

we only need to define $[\![v_h]\!] = v_h$ and $\{\mathbf{w}_h\} = \mathbf{w}_h$. The space $\mathbf{W}_h^d$ may be constructed with polynomials of degree d different from k, that was used to build V_h. However, if $k \leq d+1$, it is possible to use static condensation to define a linear operator we call $\mathbf{D}_{DG}\colon V_h \mapsto \mathbf{W}_h^d$, which is defined on every element $E \in \mathcal{T}_h$ as

$$\mathbf{D}_{DG} v_h|_E = \begin{cases} \nabla v_h|_E & E \notin \mathcal{M}_h \\ \nabla v_h|_E + \mathbf{R}^E([\![v_h]\!]) & E \in \mathcal{M}_h. \end{cases} \tag{33}$$

The operator $\mathbf{D}_{DG}$ provides the discontinuous Galerkin derivative while the lifting operator $\mathbf{R}\colon L^2(\Gamma_{Q_h}) \mapsto \mathbf{W}_h^d$, which restricts to $\mathbf{R}^E$ in element E, assigns to any function $v \in L^2(\Gamma_{Q_h})$, a unique element of $\mathbf{W}_h^d$ as

$$\int_{w_M \cap \Omega_h} \mathbf{R}(v) \cdot \mathbf{w}_h d\Omega = -\int_{\Gamma_Q} v\{\mathbf{w}_h\} \cdot \mathbf{n}\, d\Gamma \quad \forall\ \mathbf{w}_h \in \mathbf{W}_h^d. \tag{34}$$

More succinctly, if N_a and L_b are the bases chosen for functions in V_h and $\mathbf{W}_h^d$ restricted to E, then for any $v \in V_h$,

$$\mathbf{R}^E(v_a N_a) = v_a \mathbf{R}^E(N_a) \tag{35}$$

$$= v_a \mathbf{e}_i L_b R^E_{abi} \tag{36}$$

From an implementation perspective, the matrix for the lifting operators (R_{abi}) for each element can be precomputed and used to computed the DG derivative instead of looping over each face every time a derivative needs to be computed. A more detailed discussion of this DG approximation and implementation can be found in [41].

4.4 Extension of Boundary Conditions

Let Dirichlet and Neumann boundary condition be specified as $f : \Gamma^d \mapsto \mathbb{R}$ and $\tau : \Gamma^\tau \mapsto \mathbb{R}$ respectively, where Γ^d and Γ^τ are subsets of Γ such that $\Gamma^d \cap \Gamma^\tau = \emptyset$ and $\Gamma^d \cup \Gamma^\tau = \Gamma$. To define boundary conditions on Γ_h, 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. Second, we define functions $\tilde{f}$ and $\tilde{\tau}$ to impose as Dirichlet and Neumann boundary conditions on these two subsets.

The set Γ_h^d is defined as the union of pieces of the approximate boundary belonging to elements in $\mathcal{Q}_h$ which are not intersected by Γ^τ, i.e.,

$$\Gamma_h^d = \bigcup_{E \in \mathcal{Q}_h : \Gamma^\tau \cap E = \emptyset} \Gamma_h^E, \tag{37}$$

where Γ_h^E is the abbreviation for $\Gamma_h \cap E$.

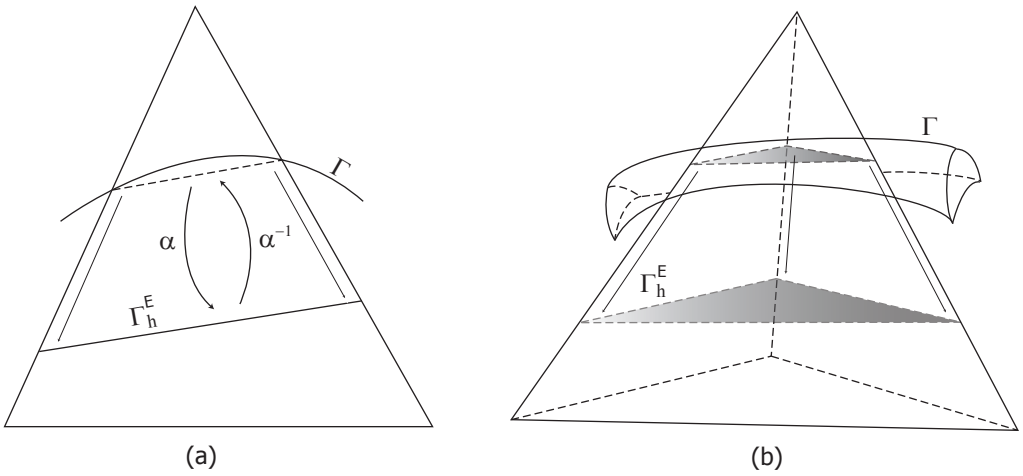


Fig. 3. Illustration of the idea of a numerical extension for a function for triangular and tetrahedral meshes. The restriction of the approximate boundary to each element, Γ_h^E is assumed to be a straight segment in 2D (on the left) or a planar section in 3D (on the right). The values of the function f at the intersection points of Γ and the edges of the element are transported to corresponding points on Γ_h^E.

Consequently, we have that $\Gamma_h^d = \Gamma_h$ if $\Gamma^d = \Gamma$. Neumann boundary conditions are imposed on the remaining part of Γ_h, i.e.,

$$\Gamma_h^\tau = \Gamma_h \setminus \Gamma_h^d. \tag{38}$$

Hence, $\Gamma_h^d \cup \Gamma_h^\tau = \Gamma_h$ analogous to having $\Gamma^d \cup \Gamma^\tau = \Gamma$.

We can now define boundary conditions on Γ_h by extending the functions f and τ to the sets Γ_h^d and Γ_h^τ respectively. One such extension method, which we term as the *Numerical Extension* is described here.

4.4.1 Numerical Extension of Dirichlet Boundary Conditions

Given the function $f : \Gamma^d \mapsto \mathbb{R}$, we define a new function $\tilde{f} : \Gamma_h^d \mapsto \mathbb{R}$ which will serve as the Dirichlet boundary condition for the problem on Ω_h. We define $\tilde{f}$ piecewise in each element by defining $\tilde{f}$ on every non-empty set $\Gamma_h^{d,E}$. Note that Γ_h^E is a linear segment for $n = 2$ and a planar section for $n = 3$. Let e_i denote the i^{th} edge of E, labelled sequentially. Define $\tilde{f}|_{\Gamma_h^{d,E}}$ to be the linear polynomial on $\Gamma_h^{d,E}$ whose value at the point $\Gamma_h^{d,E} \cap e_i$ is prescribed to be the value of f at $\Gamma_h^d \cap e_i$. Figure 3 illustrates this idea as one in which the values of f at the intersection point of Γ^d and edge e_i is transported as constant along e_i to define the value of $\tilde{f}$ at the intersection point of $\Gamma_h^{d,E}$ and e_i.

Note that because of the way Γ_h^d was defined, $\Gamma_h^{d,E}$ is non-empty only if $\Gamma^{d,E}$ is non-empty. This ensures that the above construction of $\tilde{f}$ makes sense because f is defined at $\Gamma^{d,E} \cap e_i$ whenever $\Gamma_h^{d,E} \cap e_i$ is non-empty. For the 3D case, an

additional detail is required. As shown in Figure 22, Γ_h^E, and hence $\Gamma_h^{d,E}$, is either a triangle or a quadrilateral. If $\Gamma_h^{d,E}$ is a triangle, $\tilde{f}$ is constructed as discussed so far. However, if $\Gamma_h^{d,E}$ is a quadrilateral, it may not be possible to construct a linear function $\tilde{f}|_{\Gamma_h^{d,E}}$ that equals the values of f transported along all four edges. In such a case, we determine $\tilde{f}|_{\Gamma_h^{d,E}}$ using the values of f on the three edges whose intersection points with $\Gamma_h^{d,E}$ forms the largest triangle in $\Gamma_h^{d,E}$. Trials with numerical experiments showed that this is should be preferred over an arbitrary choice of three edges, particularly because the aspect ratio of the quadrilateral $\Gamma_h^{d,E}$ can be large.

4.4.2 Numerical Extension of Neumann Boundary Conditions

The definition of $\tilde{\tau} : \Gamma_h^\tau \mapsto \mathbb{R}$ using τ is identical to the definition of $\tilde{f}$ using f barring for the following detail. The approximate Neumman boundary may intersect an edge even though the exact one does not, i.e., $\Gamma^{\tau,E} \cap e_i$ may be empty even though $\Gamma_h^{\tau,E} \cap e_i$ is not. Therefore, we need to define the value of $\tilde{\tau}$ at $\Gamma_h^{\tau,E} \cap e_i$ even if τ is not defined at $\Gamma^{\tau,E} \cap e_i$. On these edges, we use an average of the values of $\tilde{\tau}$ on the edges where it is known. Such an average may be needed only in a few elements- those which are intersected by both the Dirichlet and Neumann boundary (i.e., $\Gamma^d \& \Gamma^\tau$).

Remark 1: A simple interpretation of the numerical extension method, which may perhaps aid in generalizing it for higher order elements, is as follows. Within each element, a bijection $\alpha : \Gamma^E \mapsto \Gamma_h^E$ between the exact and approximate boundaries can be determined, reminiscent of the use of isoparametric elements for curved domains. The extended boundary condition is then an approximation of the function $(f \circ \alpha)$ using a linear polynomial. In the $2D$ case as well as in the $3D$ case with Γ_h^E being a triangle, $\tilde{f}|_{\Gamma_h^E}$ is simply the linear interpolant of the function $f \circ \alpha$. Note that for this construction of $\tilde{f}$, it was not necessary to explicitly determine α, unlike what is needed when using curved elements described in section 2.

Remark 2: Needless to say, the numerical extension method is possibly one among a whole class of methods that can be used to define boundary conditions on an approximate boundary using the prescribed values on the exact boundary. An obvious and perhaps widely used one is what we will henceforth refer to as an *analytical extension*. Given $f : \Gamma^d \mapsto \mathbb{R}$, the analytical extension of f, is a sufficiently smooth function $\tilde{f}$ that is defined on an open set in $\mathbb{R}^n$ containing $\Gamma^d \cup \Gamma_h^d$ and is such that $\tilde{f}|_{\Gamma^d} = f$. The Dirichlet boundary condition on Γ_h^d can be simply defined as $\tilde{f}|_{\Gamma_h^d}$. An identical discussion holds for the Neumann boundary condition as well. (Alternately, $\tilde{f}(x)$ for $x \in \Gamma_h^d$ can be defined to be the value of f at the point on Γ^d closest to x.) It is easy to see that the numerical extension did indeed define one such function even though

the extension was explicitly defined only on Γ_h^d.
Another point worth noting is that often, an analytical extension for f relies on f being given as an analytical expression. This may not always be the case. For example, the boundary conditions may be computed from a companion simulation and hence f may be known only at a few points lying on Γ^d. In such instances, the numerical extension is clearly the more favorable one. Nonetheless, owing to its simplicity and common usage, the analytical extension is an important one and its suitability for the proposed DG-IBM formulation needs to be evaluated. It is therefore thoroughly investigated along with the numerical extension method using numerical examples in section 5.

The discussion thus far is sufficient to completely define the bilinear operator $a_h(\cdot,\cdot)$ and the linear operator $F_h(\cdot)$ for a given boundary value problem. In this article, we are interested in using DG-IBM for elasticity problems and hence, these operators are explicitly written out for this particular case.

4.5 The Elasticity Problem

We consider $\Omega \subset \mathbb{R}^n$ to be the reference configuration of a body undergoing a quasistatic deformation upon the action of external loads. We are interested in finding the deformation mapping $\boldsymbol{\varphi}\colon \Omega \mapsto \mathbb{R}^n$ that maps the reference configuration Ω to its deformed configuration. The body is assumed to be simple and elastic, i.e., made of materials for which there exists a strain energy density function $W(\mathbf{X},\mathbf{F}) : \Omega \times \mathbb{R}^{n\times n} \mapsto \mathbb{R}$ satisfying the postulate of material frame indifference. Here, $\mathbf{F} = \nabla\boldsymbol{\varphi}(\mathbf{X})$ is the deformation gradient at point $\mathbf{X}$ and the material frame indifference postulate states that there exists a function $\hat{W}$ such that $\hat{W}(\mathbf{X},\mathbf{C}) = W(\mathbf{X},\mathbf{F})$, where $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$ is the Cauchy-Green deformation tensor. The constitutive relation for these materials follows as $\mathbf{P} = \partial W/\partial \mathbf{F}$.

We will consider a class of problems in which the deformation mapping of the elastic body is prescribed on a nonempty part of the boundary $\Gamma^d \subseteq \partial\Omega$. Therein, we shall request

$$\boldsymbol{\varphi} = \overline{\boldsymbol{\varphi}} \quad \text{on } \Gamma^d. \tag{39}$$

The rest of the boundary, $\Gamma^\tau = \Gamma \setminus \Gamma^d$, is subjected to external tractions, given by the function $\overline{\mathbf{T}} : \Gamma^\tau \mapsto \mathbb{R}^d$.

Configurations of mechanical equilibrium are stationary points of the potential energy functional

$$I[\boldsymbol{\varphi}] = \int_\Omega W(\nabla\boldsymbol{\varphi})\, d\Omega - \int_{\Gamma^\tau} \overline{\mathbf{T}} \cdot \boldsymbol{\varphi}\, d\Gamma, \tag{40}$$

from which the following Euler-Lagrange equations are obtained

$$\begin{aligned} \operatorname{div}\left[\mathbf{P}\left(\nabla\boldsymbol{\varphi}\left(\mathbf{X}\right)\right)\right] &= 0 \quad \text{in } \Omega \\ \mathbf{P}\left(\nabla\boldsymbol{\varphi}\left(\mathbf{X}\right)\right)\cdot\mathbf{N} &= \overline{\mathbf{T}} \quad \text{on } \Gamma^{\tau}, \end{aligned} \tag{41}$$

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

4.6 The Discrete Elasticity Problem

To determine a deformation mapping that satisfies the equilibrium and boundary conditions given above, we consider the corresponding weak form of the problem which is to find $\boldsymbol{\varphi} \in V^n$ (each component of $\boldsymbol{\varphi}$ belongs to V) such that

$$a(\boldsymbol{\varphi}, \mathbf{v}) = F(\mathbf{v}) \quad \forall\ \mathbf{v} \in V_{\partial}^{n} \quad \text{and} \tag{42}$$

$$\boldsymbol{\varphi}|_{\Gamma^d} = \overline{\boldsymbol{\varphi}}. \tag{43}$$

where

$$a(\boldsymbol{\varphi}, \mathbf{v}) = \int_{\Omega} \frac{\partial W}{\partial \mathbf{F}}(\mathbf{X}, \nabla\boldsymbol{\varphi}) : \nabla\mathbf{v}\ d\Omega, \tag{44}$$

$$F(\mathbf{v}) = \int_{\Gamma^{\tau}} \overline{\mathbf{T}}\cdot\mathbf{v}\ d\Gamma \quad \text{and} \tag{45}$$

$$V_{\partial} = \{v \in V :\ v|_{\Gamma^d} = 0\}. \tag{46}$$

The corresponding discrete variational principle for the DG-IBM framework is to find a stationary point $\boldsymbol{\varphi}_h \in V_h^n$ of the discrete energy functional

$$I_h[\boldsymbol{\varphi}_h] = \sum_{E\in\mathcal{T}_h} \int_{E\cap\Omega_h} W(\mathbf{X}, \mathbf{D}_{DG}(\boldsymbol{\varphi}_h))\ dV - \int_{\Gamma_h^{\tau}} \tilde{T}\cdot\boldsymbol{\varphi}_h\ dS, \tag{47}$$

where $\tilde{\mathbf{T}}$ is the extension of the function $\overline{\mathbf{T}}$. The discrete problem then is to find $\boldsymbol{\varphi}_h \in V_h^n$ that satisfies the discrete Euler-Lagrange equation

$$< \delta I_h[\boldsymbol{\varphi}_h], \mathbf{v}_h >= 0 \quad \forall\ \mathbf{v}_h \in V_{h,\partial}^{n}, \tag{48}$$

where $V_{h,\partial} = \{v_h \in V_h :\ v_h|_{\Gamma_h^d} = 0\}$. This translates to finding $\boldsymbol{\varphi}_h \in V_h^n$ such that

$$\sum_{E\in\mathcal{T}_h} \int_{E\cap\Omega_h} \frac{\partial W}{\partial \mathbf{F}}\left(\mathbf{X}, \mathbf{D}_{DG}\boldsymbol{\varphi}_h\right) : \mathbf{D}_{DG}\mathbf{v}_h\ dV - \int_{\Gamma_h^{\tau}} \tilde{\mathbf{T}}\cdot\mathbf{v}_h\ dS = 0 \quad \forall\ \mathbf{v}_h \in V_{h,\partial}^{n}, \tag{49}$$

$$\boldsymbol{\varphi}_h|_{\Gamma_h^d} = \tilde{\boldsymbol{\varphi}}_h. \tag{50}$$

Next, we explicitly write down the bilinear form $a_h(\cdot,\cdot)$ and the linear form $F_h(\cdot)$ for linear and nonlinear elasticity.

Linear Elasticity: For a linear elastic material, the strain energy density is of the form

$$W = \boldsymbol{\epsilon} : \mathbb{C} : \boldsymbol{\epsilon}, \tag{51}$$

where the infinitesimal strain field $\boldsymbol{\epsilon}$ is the symmetric gradient of the displacement field, i.e., $\boldsymbol{\epsilon} = \mathrm{sym}(\mathbf{F} - \mathbf{I})$ with $\mathrm{sym}(\mathbf{F}) = (\mathbf{F} + \mathbf{F}^{\mathrm{T}})/2$. The fourth order elasticity tensor $\mathbb{C} = \frac{\partial W}{\partial(\mathbf{F}^{\mathrm{T}}\mathbf{F})}$ has major and minor symmetries, is assumed to be constant and positive definite. The classical relationship between the Cauchy stress, denoted by $\boldsymbol{\sigma}$ and the strain field is derived as

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\epsilon}. \tag{52}$$

Further, if the material is assumed to be isotropic, the elasticity tensor reduces to

$$\mathbb{C}_{iJkL} = (\lambda + \frac{2\mu}{3})\delta_{iJ}\delta_{kL} + \mu\left(\delta_{ik}\delta_{JL} + \delta_{iL}\delta_{Jk} - \frac{2}{3}\delta_{iJ}\delta_{kL}\right), \tag{53}$$

where λ and μ are material parameters called the Lamé constants. For such a material, the stress-strain relationship reduces to

$$\boldsymbol{\sigma} = \lambda \mathrm{tr}(\boldsymbol{\epsilon})\mathbf{I} + 2\mu\boldsymbol{\epsilon}. \tag{54}$$

The discrete problem in 49 translates to finding $\boldsymbol{\varphi}_h \in V_h^n$ such that

$$a_h(\boldsymbol{\varphi}_h, \mathbf{v}_h) = F_h(\mathbf{v}_h) \quad \forall\ \mathbf{v}_h \in V_{h,\partial} \quad \text{and} \tag{55}$$

$$\boldsymbol{\varphi}_h|_{\Gamma_h^d} = \tilde{\boldsymbol{\varphi}}, \tag{56}$$

where

$$a_h(\boldsymbol{\varphi}_h, \mathbf{v}_h) = \sum_{E\in\mathcal{T}_h} \int_{E\cap\Omega_h} \mathbf{D}_{DG}(\boldsymbol{\varphi}_h) : \mathbb{C} : \mathbf{D}_{DG}(\mathbf{v}_h)\ d\Omega_h, \quad \text{and} \tag{57}$$

$$F_h(\mathbf{v}_h) = \int_{\Gamma_h^\tau} \tilde{\mathbf{T}} \cdot \mathbf{v}_h\ d\Gamma_h. \tag{58}$$

Nonlinear Elasticity: For clarity and for later use in section 5, we consider a specific example, namely a compressible Neohookean material. The strain energy density for this material is given by

$$W(\mathbf{X}, \mathbf{F}) = \frac{\lambda}{2}\ \log\left(\det(\mathbf{F})\right)^2 + \frac{\mu}{2}\ \mathrm{tr}(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I}), \tag{59}$$

if $\det(\mathbf{F}) > 0$, and $W(\mathbf{F}) = +\infty$ otherwise. Here, $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$ is the right Cauchy-Green deformation tensor. The first Piola-Kirchhoff stress is computed as

$$\mathbf{P}(\mathbf{F}) = \lambda\ \log\left(\det\mathbf{F}\right)\mathbf{F}^{-\mathrm{T}} + \mu\left(\mathbf{F} - \mathbf{F}^{-\mathrm{T}}\right), \tag{60}$$

and the symmetric second Piola-Kirchhoff stress tensor as $\mathbf{S} = 2\frac{\partial \hat{W}}{\partial \mathbf{C}} = \mathbf{F}^{-1}\mathbf{P}$. The first and second elasticity tensors given by $\mathbb{A} = \frac{\partial^2 W}{\partial \mathbf{F}^2}$ and $\mathbb{C} = \frac{\partial^2 \hat{W}}{\partial \mathbf{C}^2}$ are related as

$$\mathbb{A}_{iJkL} = 2\mathbb{C}_{IJKL}\mathbf{F}_{iI}\mathbf{F}_{kK} + \mathbf{S}_{JL}\delta_{ik}. \tag{61}$$

For this material, since 49 is a nonlinear equation in $\boldsymbol{\varphi}_h$, we adopt a Newton-Raphson iteration procedure by considering the linearization of 49 around $\boldsymbol{\varphi}_h$ as

$$< \delta^2 I_h[\boldsymbol{\varphi}_h], \mathbf{v}_h, \mathbf{u}_h > + < \delta I_h[\boldsymbol{\varphi}_h], \mathbf{v}_h >= 0 \quad \forall\ \mathbf{v}_h \in V^n_{h,\partial}, \tag{62}$$

to find $\mathbf{u}_h$ that is considered as a displacement field with respect to the configuration of the body Ω_h determined by $\boldsymbol{\varphi}_h$. This translates to solving for $\mathbf{u}_h$ around the point $\boldsymbol{\varphi}_h$ that satisfies

$$a_h(\mathbf{u}_h, \mathbf{v}_h) = F_h(\mathbf{v}_h) \quad \forall\ \mathbf{v}_h \in V_{h,\partial} \quad \text{and} \tag{63}$$

$$\boldsymbol{\varphi}_h|_{\Gamma^d_h} = \tilde{\boldsymbol{\varphi}}, \tag{64}$$

where a_h and F_h are now

$$a_h(\mathbf{u}_h, \mathbf{v}_h) = \sum_{E \in \mathcal{T}_h} \int_{E \cap \Omega_h} \mathbf{D}_{DG}\mathbf{u}_h : \mathbb{A} : \mathbf{D}_{DG}\mathbf{v}_h \ dV \quad \text{and} \tag{65}$$

$$F_h(\mathbf{v_h}) = \int_{\Gamma^\tau_h} \tilde{T} \cdot \mathbf{v}_h \ dS. \tag{66}$$

This formally completes the description of the DG-IBM formulation for elasticity problems.

5 Numerical Examples

We now present numerical examples in elasticity to examine the DG-IBM formulation with extended boundary conditions. A simple implementation procedure that is described in the appendix was used for the simulations. The first example is that of a two dimensional problem to determine the deformation of an annular ring. It is followed by a three dimensional example to be solved for the torsion of a thick spherical shell. We use the different extension methods for displacement and traction boundary conditions to solve for the deformation mapping. 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 deformation of a femur bone sample under compression and the deformation of an image based geometry. These examples serve to illustrate that the method is indeed as simple for a complex geometry as it is for a simple one.

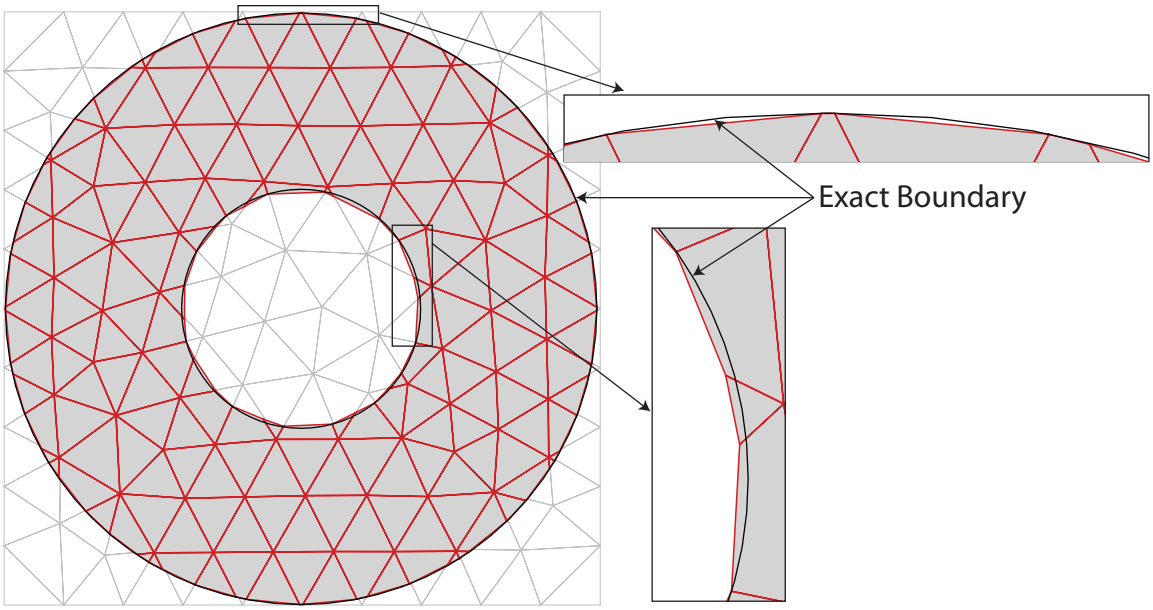


Fig. 4. Detail of approximation of the domain for the example of the annular ring. The initial background mesh $\mathcal{T}_h$ in which the boundary is immersed is shown in light gray on the back. Notice that the approximate boundary lies occasionally inside and/or outside the exact domain, as well as that the exact boundary does not always intersect the element faces where the approximate one does.

5.1 Deformation of an Annulus

In this example, we are interested in describing the deformation of a thick walled linear elastic hollow cylinder. Its inner surface is rigidly fixed while non-zero displacements are prescribed on the outer surface. Assuming plane strain, 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. The boundary conditions imposed on the deformation mapping are

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X})|_{\|\mathbf{X}\|=r_0} &= \mathbf{X} \quad \text{and} \\ \overline{\boldsymbol{\varphi}}(\mathbf{X})|_{\|\mathbf{X}\|=r_1} &= 6\mathbf{X}/5. \end{aligned} \tag{67}$$

Notice that we have $\Gamma^d = \Gamma$. We choose the enclosing domain $\mathcal{B}$ 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 elastic material is modeled as isotropic and homogeneous with the material properties in equation (54) being $\lambda = 1$ and $\mu = 1$. The function used to define the approximate domain Ω_h is chosen as the piecewise linear interpolant $\mathcal{I}\phi$ over $\mathcal{T}_h$ of the (signed distance) function

$$\phi(\mathbf{X}) = \max\{r_0 - \|X\|, \|X\| - r_1\}. \tag{68}$$

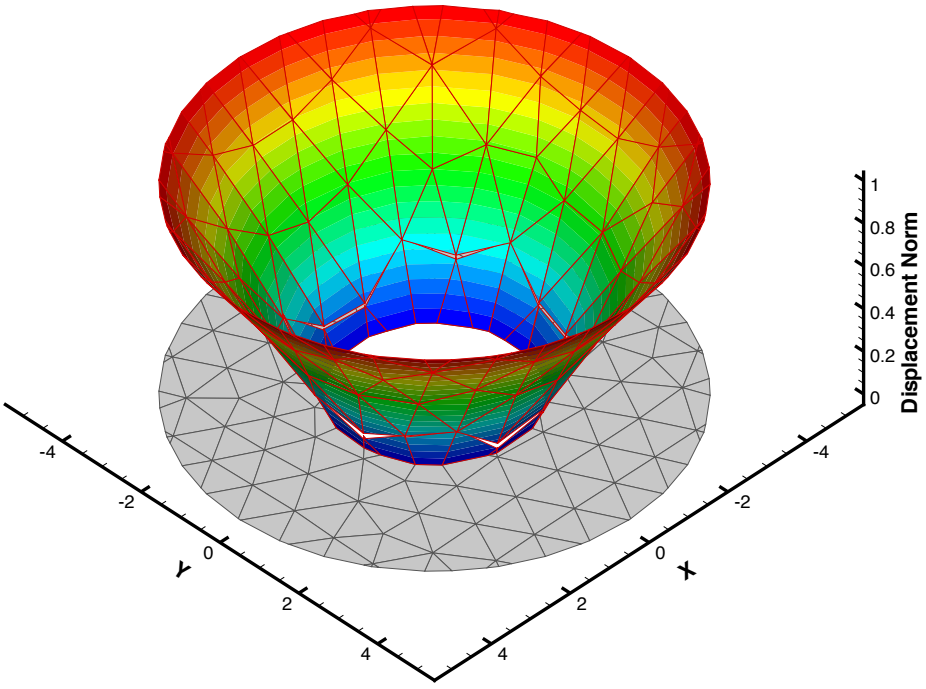


Fig. 5. Euclidean norm of the displacements at each point of the domain. The mesh is shown in the X-Y plane, while the value of the norm of the displacement is displayed on the axis orthogonal to it. Notice the clearly evident discontinuities in the solution, only along the faces of elements that are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

Recall that the zero level set of $\mathcal{I}\phi$ is the approximate boundary Γ_h. Naturally, the approximate and exact boundaries may not coincide as shown in the figure 4. This problem has an exact solution of the form

$$\boldsymbol{\varphi}_{ex}(\mathbf{X}) = \left(\frac{A_1}{\|\mathbf{X}\|^2} + A_2 + 1 \right) \mathbf{X}, \tag{69}$$

where

$$\begin{aligned} A_1 &= -0.956098 \\ A_2 &= 0.239024. \end{aligned}$$

The traction acting at $r = r_1$ is given by

$$\overline{\mathbf{T}}(\mathbf{X}) = 2 \left((\lambda + \mu) A_2 - \frac{\mu A_1}{24.5} \right) \mathbf{e}_r. \tag{70}$$

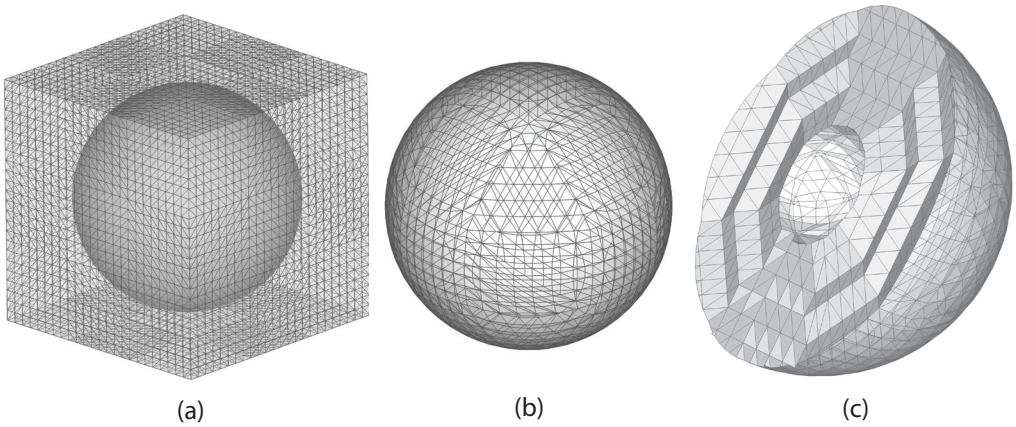


Fig. 6. Approximation of the spherical shell using a non-conforming tetrahedral mesh on a cube. Shown on the left is the exact domain immersed in the non-conforming mesh followed by the approximate domain that is the zero level set of the interpolant of the function ϕ given by equation (72). A section of the approximate solid shown on the right reveals more details of the approximation.

Figure 4 shows the approximation of the problem domain. For later use, we denote the approximation of the inner circle $r = r_0$ as Γ_h, r_0 and the outer circle $r = r_1$ to be Γ_h, r_1.

Figure 5 shows the deformation of the annulus computed using an analytical extension for the boundary conditions as

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X})|_{\Gamma_{h,r_0}} &= \mathbf{X} \qquad \text{and} \\ \overline{\boldsymbol{\varphi}}(\mathbf{X})|_{\Gamma_{h,r_1}} &= 6\mathbf{X}/5. \end{aligned} \tag{71}$$

In this example since displacements are prescribed on the entire boundary, we have $\Gamma_h^d = \Gamma_h$. 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 repectively. The encompassing domain $\mathcal{B}$ 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

$$\phi = \max\{r_0 - \|\mathbf{X}\|, \|\mathbf{X}\| - r_1\}. \tag{72}$$

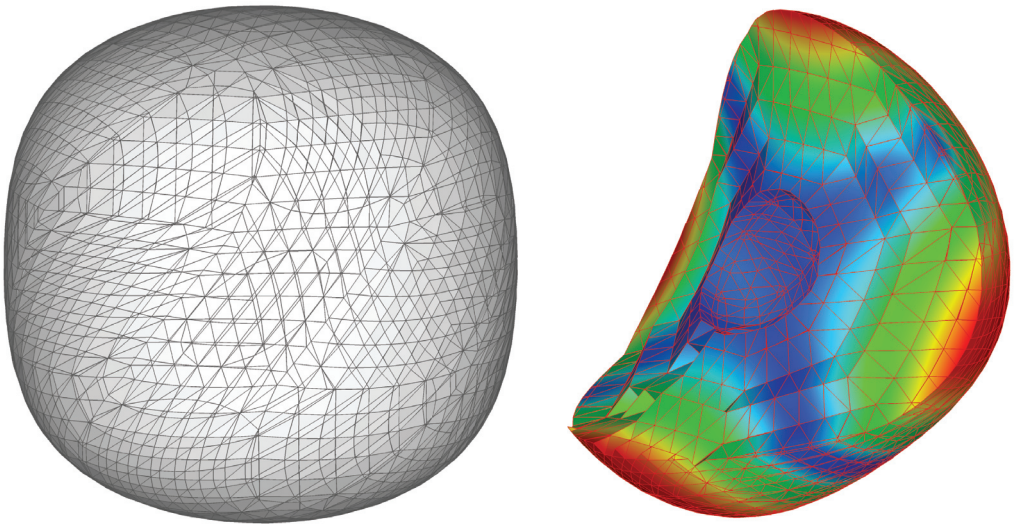


Fig. 7. Deformed configuration of the spherical shell computed using the analytical extension given in (71). Shown on the right is a section of the deformed configuration corresponding to the section shown in figure 6(c). The color contours correspond to the Euclidean norm of the displacement field.

Figure 6 shows the tetrahedral mesh of the domain $\mathcal{B}$ and the approximate domain Ω_h.

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

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_o \sin 2\xi \ \mathbf{e}_\theta \quad \text{on} \quad \Gamma_{r_0}, \\ \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_1 \sin 2\xi \ \mathbf{e}_\theta \quad \text{on} \quad \Gamma_{r_1}, \end{aligned} \tag{73}$$

where

$$\begin{aligned} K_0 &= 0.0984375 \quad \text{and} \\ K_1 &= 0.8859375, \end{aligned}$$

and ξ and θ are the zenith and azimuthal angles measured from the positive Z and X axis respectively. This problem has an exact solution given by

$$\boldsymbol{\varphi}(\mathbf{X}) = \mathbf{X} + K r^2 \sin 2\xi \ \mathbf{e}_\theta \tag{74}$$

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

$$\overline{\mathbf{T}}(\mathbf{X}) = K \mu r_1 \sin 2\xi \ \mathbf{e}_\theta. \tag{75}$$

The deformation of the spherical shell is shown in figure 7, computed using

the analytical extension for boundary conditions as

$$\begin{aligned} \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_o \sin 2\xi \ \mathbf{e}_\theta \quad \text{on} \quad \Gamma_{h,r_0}, \\ \overline{\boldsymbol{\varphi}}(\mathbf{X}) &= \mathbf{X} + K_1 \sin 2\xi \ \mathbf{e}_\theta \quad \text{on} \quad \Gamma_{h,r_1}. \end{aligned} \tag{76}$$

Though we have computed numerical solutions to the above two examples using an analytical extension for the boundary conditions, it remains to be seen if the computed solution converges to the exact one and what the rate of convergence is, if the solution does converge. This is what we do in the following section. We first describe how better approximations of the exact solution can be computed once an extension method for the boundary conditions is adopted. The two examples are modified to include traction boundary conditions without altering the exact solutions. The error in the numerical solution when computed using the analytical and numerical extension methods for Dirichlet and Neumann boundary conditions is examined and the order of convergence determined in each case.

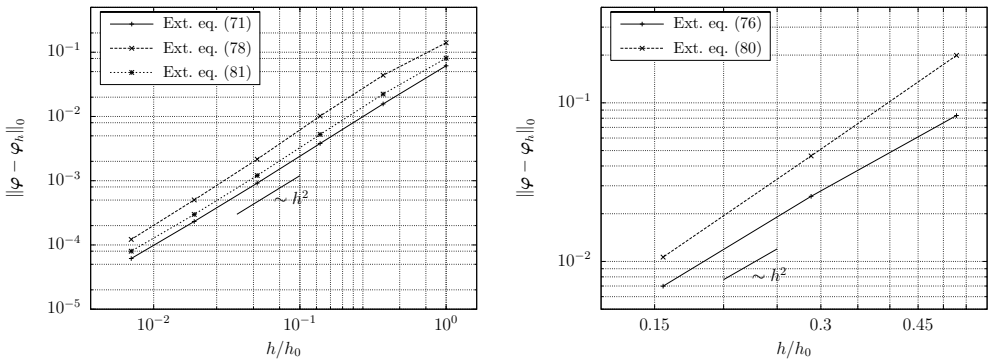
5.3 Convergence

Part of the charm of the proposed method is that more accurate solutions can be computed by refining the mesh $\mathcal{T}_h$. This is quite remarkable since such a refinement simultaneously improves the approximation of the domain, the function space and 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. 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

$$\|\boldsymbol{\varphi} - \boldsymbol{\varphi}_h\|_0 = \left(\int_{\Omega \cap \Omega_h} (\boldsymbol{\varphi} - \boldsymbol{\varphi}_h) \cdot (\boldsymbol{\varphi} - \boldsymbol{\varphi}_h) \ dV \right)^{1/2}. \tag{77}$$

The order of convergence is measured as the slope of the error versus h curve. Since we are using linear elements, we expect the convergence to be quadratic. Notice that once again, we deviate from the discussion in section 3 where we proved convergence in the triplenorm. As we will show later, convergence in the L_2 norm and the triplenorm are equivalent for this formulation.

Extension of Dirichlet Boundary Conditions: In both the examples, Dirichlet boundary conditions were imposed on the entire boundary and consequently, we had $\Gamma_h^d = \Gamma_h$. We have already used a simple and perhaps the most natural analytical extension for Dirichlet boundary conditions to compute the deformation mapping in the two examples; the analytical expressions for the



(a) (b)

Fig. 8. Rate of convergence of the numerical solution computed using different analytical extensions for the boundary conditions in the examples of the annulus and the spherical shell. Optimal convergence rates are achieved in both cases.

boundary conditions prescribed on Γ_{r_0} and Γ_{r_1} were just evaluated at their respective approximations Γ_{h,r_0} and Γ_{h,r_1}. For the example of the annulus, another possible analytical extension would be the following:

$$\overline{\boldsymbol{\varphi}}_h(\mathbf{X}) = \mathbf{X} \qquad \text{on} \quad \Gamma_{h,r_0} \quad \text{and}$$
$$\overline{\boldsymbol{\varphi}}_h(\mathbf{X}) = \mathbf{X} + 0.2r\ \mathbf{e}_r + \sin \pi(r - \sqrt{24.5})\ \mathbf{e}_\theta \quad \text{on} \quad \Gamma_{h,r_1}. \tag{78}$$
$$\tag{79}$$

Figure 8(a) shows the error as a function of the mesh parameter h. While the magnitude of the error depends on the choice extension, the rate of convergence does not. Using both these extensions for the two dimensional example, the numerical solution converges to the exact one with the optimal rate of two.

Similarly, for the example of the spherical shell, the following functions constitute legitimate analytical extensions

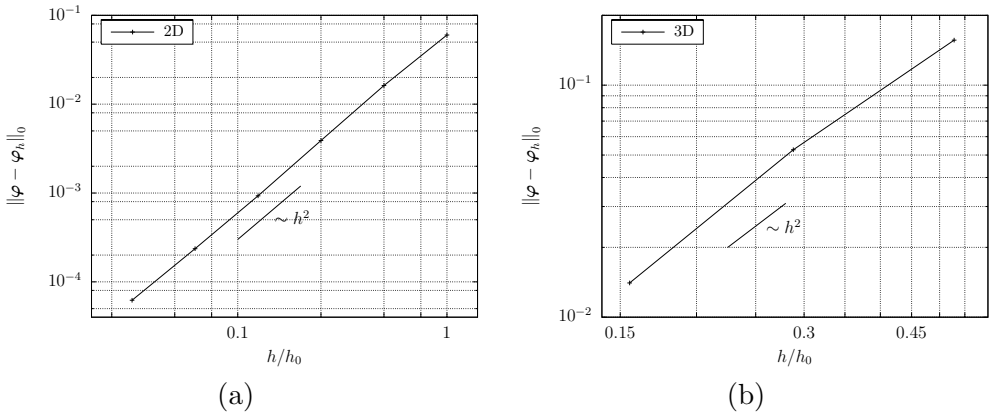
$$\overline{\boldsymbol{\varphi}}_h(\mathbf{X}) = \mathbf{X} + K_0[\sin 2\xi\ \mathbf{e}_\theta + \sin(\pi(r - r_0))]\ \mathbf{e}_\xi \quad \text{on} \quad \Gamma_{h,r_0} \quad \text{and}$$
$$\overline{\boldsymbol{\varphi}}_h(\mathbf{X}) = \mathbf{X} + K_1[\sin 2\xi\ \mathbf{e}_\theta + \sin(\pi(r - r_1))]\ \mathbf{e}_\xi \quad \text{on} \quad \Gamma_{h,r_1}. \tag{80}$$

Once again, optimal quadratic convergence in the L_2 norm is confirmed by figure 8(b).

It was remarked in section 4 that we could use as boundary conditions for points on Γ_h, the value prescribed at its closest neighbor in Γ. For the example of the annulus, this corresponds to using as boundary conditions,the analytical extension

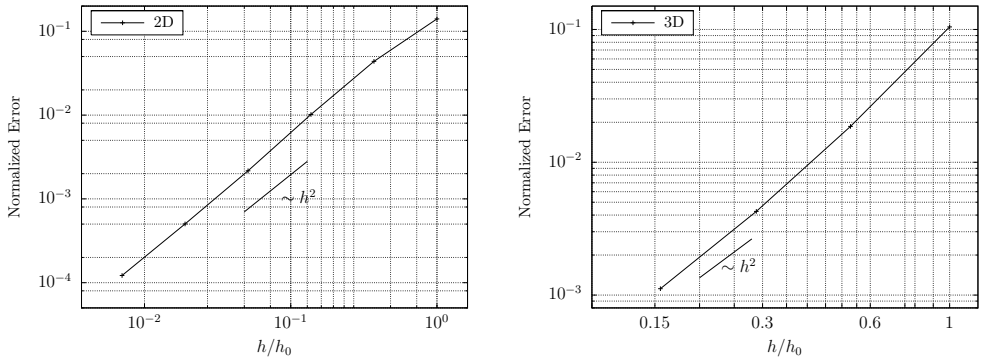
$$\overline{\boldsymbol{\varphi}}_h(\mathbf{X}) = \mathbf{X} \qquad \text{on} \quad \Gamma_{h,r_0} \quad \text{and}$$
$$\overline{\boldsymbol{\varphi}}_h(\mathbf{X}) = (r_1 + 0.2)\ \mathbf{e}_r \quad \text{on} \quad \Gamma_{h,r_1}. \tag{81}$$

Figure 8(a) shows that this extension also leads to optimal convergence rates.



(a) (b)

Fig. 9. Convergence of solution computed using a numerical extension of Dirichlet boundary conditions on Γ_{h,r_1} for the two and three dimensional example. The convergence rate is approximately two in both cases.



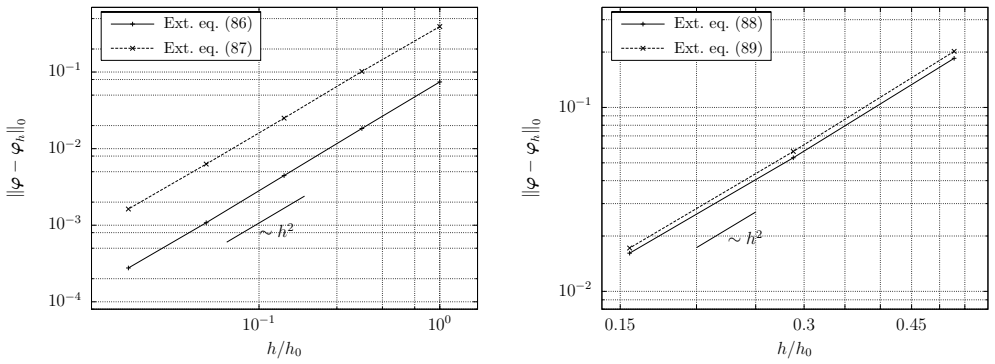
(a) (b)

Fig. 10. Quadratic convergence of area(/length) of Γ_h to that of Γ in the annulus and spherical shell examples. The errors are normalized by the exact areas(/lengths).

In the examples that follow, we will use the natural extension for the Dirichlet boundary conditions on Γ_{h,r_0} as in equations (71) and (76) for the annulus and the spherical shell respectively, while trying other extensions on Γ_{h,r_1}.

We now impose as boundary condition on Γ_{h,r_1}, the numerical extension of the prescribed boundary conditions on Γ_{r_1}. Recall that the numerical extension is a linear function on Γ_h^E whose value at the point $\Gamma_h^E \cap e_i$ equals the prescribed value at $\Gamma \cap e_i$ for edges e_i belonging to element E. For the tetrahedral elements, when the cut element is a wedge of type b, this agreement is over three of the four edges intersected by Γ and Γ_h. Figure 9 shows that once again, the solution converges with the optimal order for both the two and three dimensional example.

Next, we inspect how the proposed method works when using analytical and numerical extensions for Neumann boundary conditions. To this end, we consider a new set of boundary conditions- we now impose Dirichlet boundary



(a) (b)

Fig. 11. Quadratic convergence of numerical solution when using analytical extensions for Neumann boundary conditions on Γ_{h,r_1} in the annulus and spherical shell examples.

conditions on Γ_{r_0} and Neumann boundary conditions on Γ_{r_1}. For the example of the annulus, the new set of boundary conditions to be satisfied along with the mechanical equilibrium condition are

$$\overline{\boldsymbol{\varphi}}(\mathbf{X}) = \mathbf{X} \qquad \text{on} \quad \Gamma_{r_0} \quad \text{and} \tag{82}$$

$$\overline{\mathbf{T}}(\mathbf{X}) = 2((\lambda+\mu)A_2 - \mu A_1/24.5)\ \mathbf{e}_r \quad \text{on} \quad \Gamma_{r_1}. \tag{83}$$

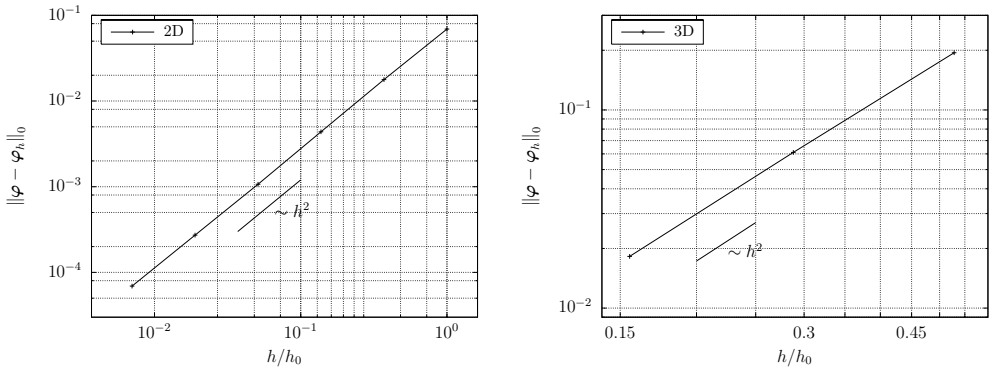
Similarly, for the example of the spherical shell, we consider the following boundary conditions

$$\overline{\boldsymbol{\varphi}}(\mathbf{X}) = \mathbf{X} + K_0 \sin 2\xi\ \mathbf{e}_\theta \quad \text{on} \quad \Gamma_{r_0} \quad \text{and} \tag{84}$$

$$\overline{\mathbf{T}}(\mathbf{X}) = K\mu r_1 \sin 2\xi\ \mathbf{e}_\theta \quad \text{on} \quad \Gamma_{r_1}. \tag{85}$$

Since the material is linear elastic and the traction imposed on Γ_{r_1} is as computed in (70) and (75) for the two examples respectively, the exact solution to these two problems are unaltered. That the components of the boundary Γ_{r_0} and Γ_{r_1} are disconnected comes in quite handy because we can independently choose what kind of boundary conditions to impose on each component. As indicated earlier, we continue to use the analytical extensions given in (71) and (76) for the Dirichlet boundary conditions on Γ_{r_0}. But before that, we make sure that the surface area (or length in 2D) of Γ_h approaches that of Γ with the right order. Since Γ_h is the zero level set of the *linear* interpolant of ϕ, we expect the area/length of to converge quadratically, which is indeed the case as shown in figure 10.

Extension of Neumann Boundary Conditions: While there is a subtle difference in the definition of Γ_h^d and Γ_h^τ (see section 4), the analytical extension method for traction and displacement boundary conditions are identical. We consider the following two analytical extensions for the traction on Γ_{h,r_1} in the



(a) (b)

Fig. 12. Numerical extension for traction boundary conditions on Γ_{h,r_1}. Like when using an analytical extension, optimal convergence rates are achieved.

example of the annulus

$$\overline{\mathbf{T}}_h(\mathbf{X}) = 2[(\lambda+\mu)A_2 - \mu A_1/24.5]\ \mathbf{e}_r, \tag{86}$$
$$\overline{\mathbf{T}}_h(\mathbf{X}) = 2[(\lambda+\mu)A_2 - \mu A_1/r^2]\ \mathbf{e}_r + (r_1 - r)[(\lambda+\mu)A_2 + \mu A_1/r^2]\ \mathbf{e}_\theta. \tag{87}$$

While the first extension simply evaluates the expression in (83) on Γ_{h,r_1}, the second one is quite arbitrary.

For the example of the spherical shell, we consider the following two extensions for the traction boundary condition on Γ_{h,r_1}

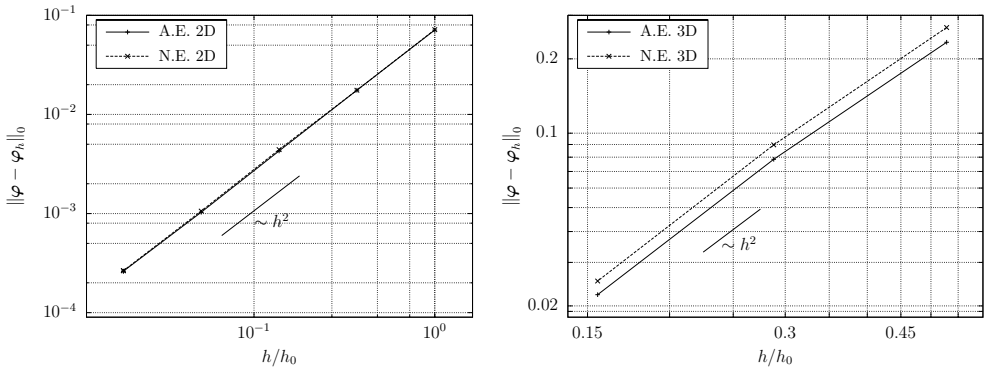
$$\overline{\mathbf{T}}_h(\mathbf{X}) = K\mu r_1 \sin 2\xi\ \mathbf{e}_\theta, \tag{88}$$
$$\overline{\mathbf{T}}_h(\mathbf{X}) = K\mu[r \sin 2\xi\ \mathbf{e}_\theta - (r_1 - r)\sin^2\xi]\ \mathbf{e}_\xi. \tag{89}$$

Figure 11 shows the ideal convergence solutions computed using the extensions given above.

Similarly, figure 12 shows the convergence of the computed solution using a numerical extension for the tractions.

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. Since we defined the sets Γ_h^d and Γ_h^τ such that their union is Γ_h and their intersection has zero mesure, this case should be similar to the ones considered previously. Of course, because of using extensions of the exact boundary conditions and since the exact solution is smooth, there isn't a singularity at the points where both Neumann and Dirichlet boundary conditions are prescribed.

For the example of the annulus, we consider the following boundary conditions



(a) (b)

Fig. 13. Extensions for a combination of Dirichlet and Neumann boundary conditions on Γ_{r_1}. A.E. and N.E. refer to analytical and numerical extensions respectively. The quadratic convergence in both the examples comes as no surprise.

on Γ_{r_1}

$$\overline{\boldsymbol{\varphi}}(\mathbf{X}) = 6\mathbf{X}/5 \quad \text{when } \mathbf{X}\cdot\mathbf{e}_2 \leq \sqrt{5}, \tag{90}$$

$$\overline{\mathbf{T}}(\mathbf{X}) = 2((\lambda+\mu)A_2 - \frac{\mu A_1}{24.5})\ \mathbf{e}_r \quad \text{when } X\cdot\mathbf{e}_2 > \sqrt{5}. \tag{91}$$

Similarly, for the example of the spherical shell, on Γ_{h,r_1} we impose the following boundary conditions

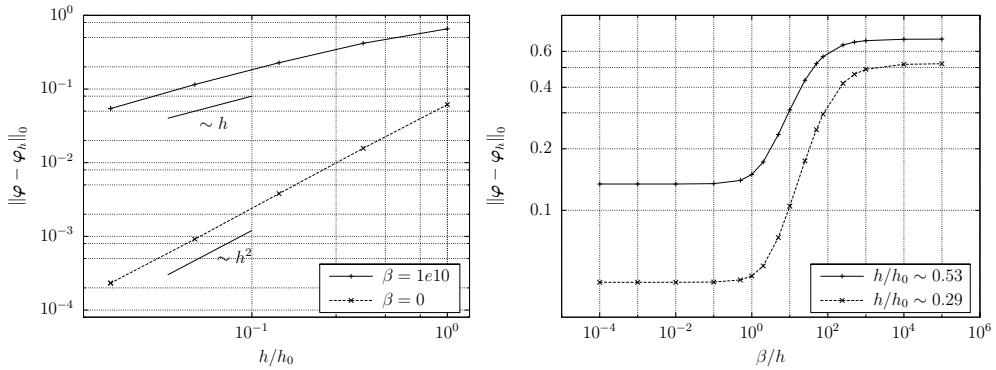
$$\overline{\boldsymbol{\varphi}}(\mathbf{X}) = \mathbf{X} + K_1 \sin 2\xi\ \mathbf{e}_\theta \quad \text{when } \mathbf{X}\cdot\mathbf{e}_3 \leq \sqrt{1.25}, \tag{92}$$

$$\overline{\mathbf{T}}(\mathbf{X}) = K\mu r_1 \sin 2\xi\ \mathbf{e}_\theta \quad \text{when } \mathbf{X}\cdot\mathbf{e}_3 > \sqrt{1.25}. \tag{93}$$

As analytical extensions for this set of boundary conditions, we evaluate the prescribed functions on the sets Γ^d_{h,r_1} and Γ^τ_{h,r_1} for each example. The numerical extension is also defined in a straightforward way. Figure 13 shows the expected results.

5.4 Stabilization

When using a DG method, the question of stabilization of jumps naturally comes up. The parameter β in (**??**) is used to penalize discontinuities in the solution. Even though we have needed to penalize discontinuities in any of the simulations presented so far, it would be interesting to see the outcome of penalizing jumps. This is particularly important because we anticipate that DG-IBM overcomes boundary locking precisely because discontinuities in the solution are permitted close to the boundary. Indeed, figure 14(a) corroborates this understanding by revealing a suboptimal $\mathcal{O}(h)$ convergence when using $\beta = 1e10$ in the example of the annulus. A similar situation is reflected in



(a) (b)

Fig. 14. Effect of stabilization on the rate of convergence. Shown on the left are the convergence curves for the example of the annulus with and without stabilization. Penalization of jumps results in a loss of an order of convergence. Shown on the right is 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.

the 3D example. Figure 14 shows the 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 to the $\mathcal{O}(h^2)$ convergence for β that is small or zero.

This does not however, rule out the possibility of the method requiring stabilization. Other examples or numerical fluxes for the DG method could be crafted that require penalization of jumps.

Next, we present two applications that showcase the use DG-IBM for simulations over complex domains in elasticity.

5.5 Numerical Examples with Nonlinear Elastic Materials

Compression of a Femur bone:
As an illustration of the proposed method's capability to handle realistic geometries, we consider an example of a femur bone sample. While boundary fitting meshes are readily available for geometries far more complicated than the bone model studied here, we simplify the problem significantly by meshing a parallelopiped enclosing the bone and letting the DG-IBM framework to 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

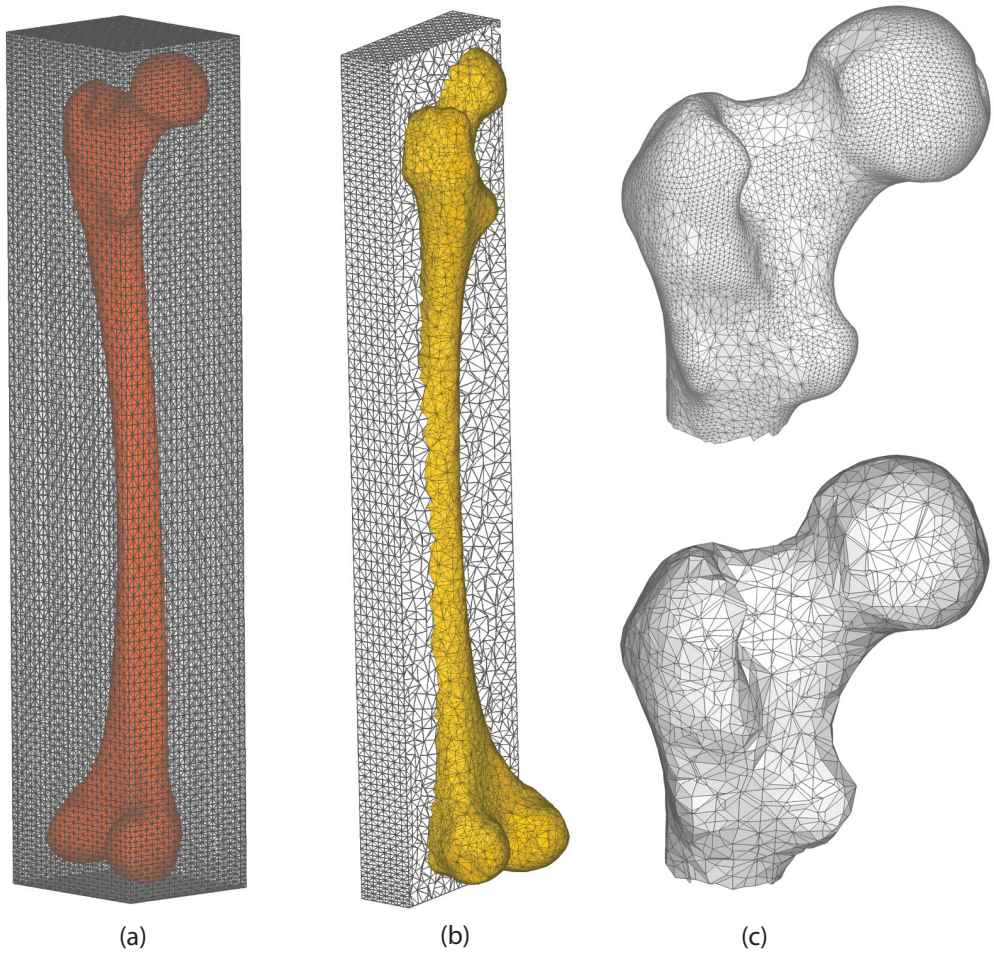


Fig. 15. Approximation of a femur bone in a non-conforming mesh. Shown on the far left is the exact surface immersed in a parallelopiped. The parallelopiped is meshed using tetrahedra. The signed distance function is evaluated only at the nodes of the mesh. The zero isosurface of the resulting function is shown in the middle. Shown on the right are the exact and approximated geometry for the head of the bone.

model. But with an immersed boundary method like ours, the last step is not required; the tessellation of the surface is used to (approximately) reconstruct the domain in a possibly non-conforming mesh. It has been estimated in [23] that about 80% of the time in finite element analysis is spent in building a boundary fitting mesh. In this way, we bypass a time consuming step almost completely.

The domain of the problem considered here us the region enclosed by a piecewise smooth surface obtained by profiling a femur bone model. The surface is represented as a tessellation of triangles. The enclosing domain $\mathcal{B}$ is a parallelopiped that is *easily* meshed with about 159,100 tetrahedral elements. The signed distance function to the surface is computed at the nodes of the non-conforming mesh and an approximation of the geometry constructed in the usual way. The resulting approximation of the domain is shown in figure 15.

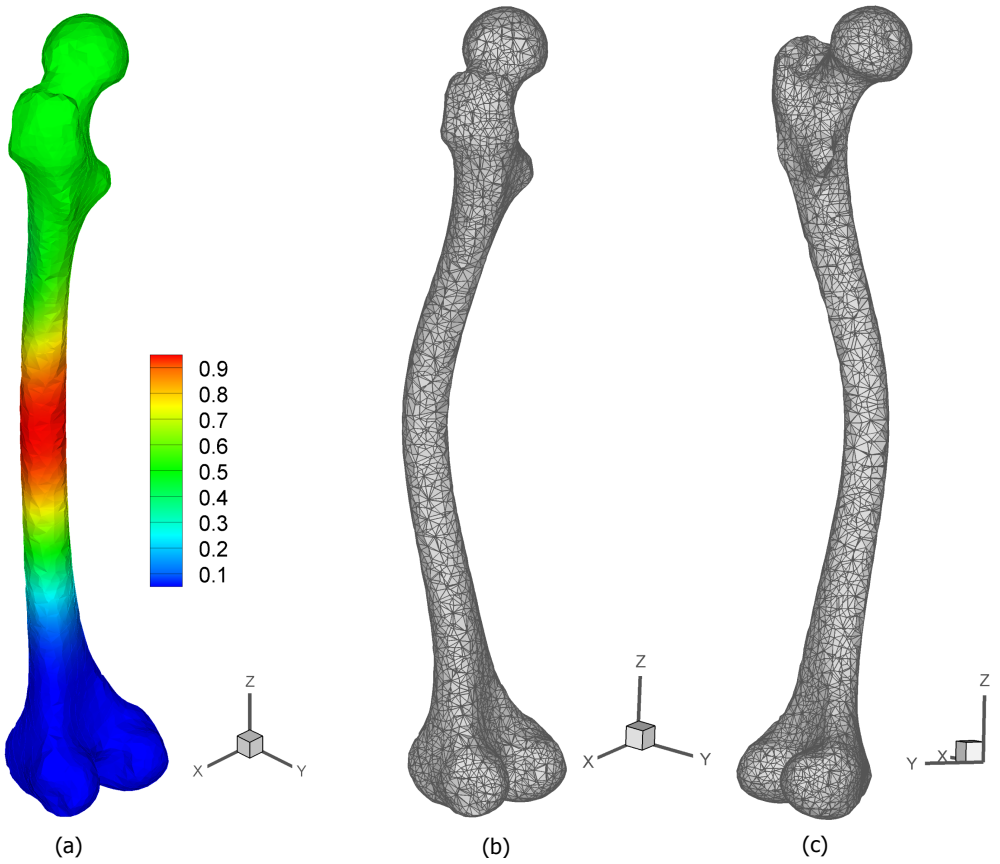


Fig. 16. Displacement contours and deformation of the femur bone as a result of fixing the bottom and imposing downward displacements at the top. The contours of the norm of the displacement field are shown in (a). The deformed configuration of the bone is shown in (b) and (c). Note that in (b) and (c), the displacement field was scale by a factor of two for visualization purposes.

The bone is modeled as a Neohookean material as in equation 60 with both Lamé constants equal to 1. 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 as

$$\overline{\boldsymbol{\varphi}}_h(\mathbf{X}) = \begin{cases} \mathbf{X} & \text{if } \mathbf{X}\cdot\mathbf{e}_z \leq 8 \text{ units} \\ \mathbf{X} - 1.0\ \mathbf{e}_3 & \text{if } (\mathbf{X}\cdot\mathbf{e}_z \geq 40 \text{ units.} \end{cases} \tag{94}$$

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. The displacement imposed at the head is about 2.1% of the length. Figure 16 plots the resulting deformation showing how the bone has buckled. Once again, we recall that these discontinuities are introduced by the numerical method even though the solution is continuous.

Application to Image based Geometries:
Image based geometries are encountered in many situations, perhaps most

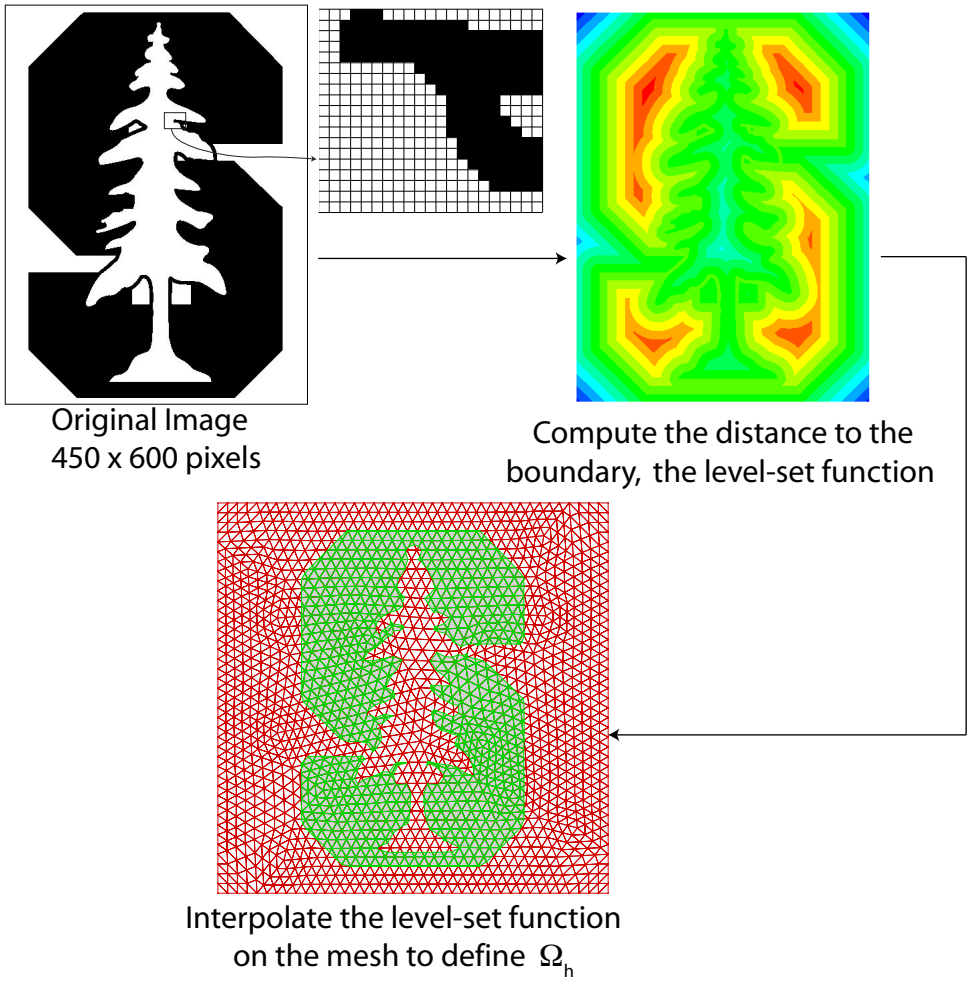


Fig. 17. Step towards the generation of image–based geometries. The starting point is a black and white image made of pixels, each one of them painted with one of two colors. A level-set function is constructed by computing the signed distance of each point to the closest pixel with an opposite color. This function is in turn interpolated over the background mesh. The zero level 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.

often in medical imaging. A possibly undesirable feature of extracting geometries from images is that the boundary is naturally jagged, or pixelated. These issues can particularly hinder simple mesh generation, particularly with three dimensional images. However, as we show next, the immersed boundary method is well suited to handle such cases.

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. The image is assumed to contain rectangular Cartesian grids in which each cell, or pixel, has one of

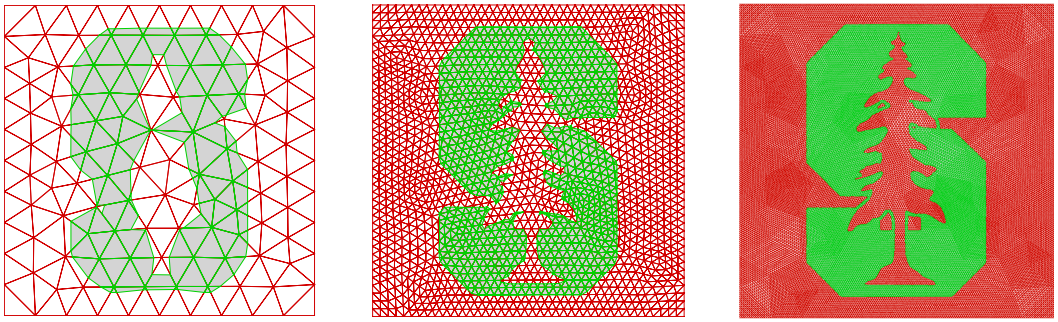


Fig. 18. Approximate domains computed with progressively finer meshes obtained by subdiving each triangle in the first mesh into four similar ones.

two possible values- say 0 for white and 1 for black. The particular geometry that we consider here is shown in figure 17.

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 the white pixel zero at the boundary and negative everywhere else (points that lie in a black pixel). The approximate domain determined by the interpolant of the level set function, also shown in figure 17. It is clear that if the mesh $\mathcal{T}_h$ over which the level set function is interpolated is much finer that the pixel size, the domain is naturally smoothed out. In such applications, highly refined meshes may be unnecessary because most often, the geometry obtained from the image is already an approximation of the exact one. This approximation sets the accuracy limit of the simulation.

For this example, the initial background mesh is shown in gray in figure 18. Finer meshes are obtained by simply 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 figure 19

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

$$\overline{\boldsymbol{\varphi}}_h(\mathbf{X}) = \frac{\mathbf{X}}{\|\mathbf{X}\|}\cos(5\theta - \frac{\pi}{2}), \tag{95}$$

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 $\mathcal{B}$ employed in the previous 2D simulations. 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 properties are taken to be homogeneous and non-linear elastic

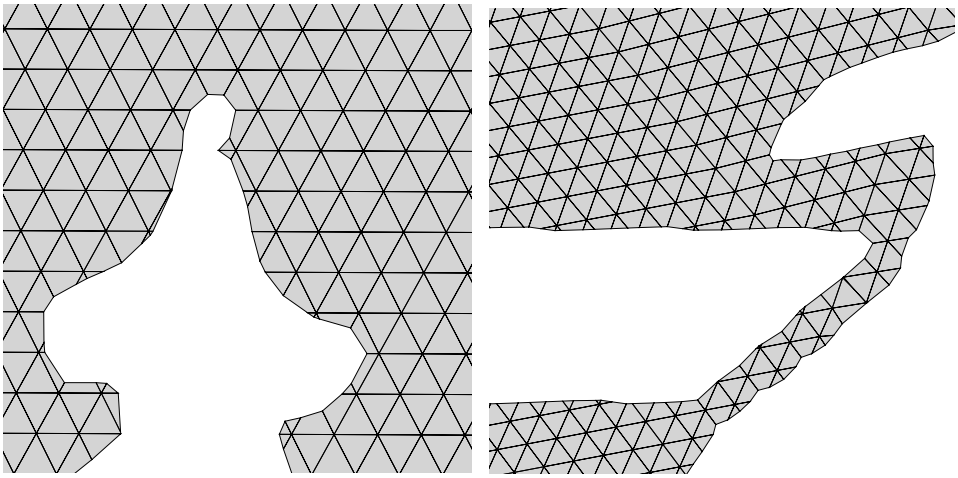


Fig. 19. Details of the approximate domain for the finest mesh used, before deformation.

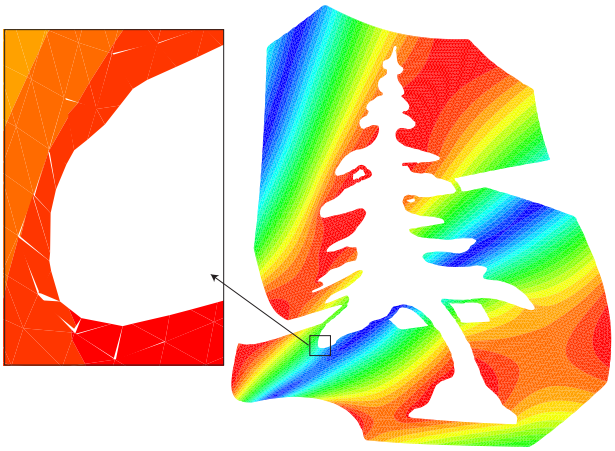


Fig. 20. Nonlinear elastic deformations of the image-based geometry. The contour plot shown is based on the Euclidean norm of the displacement field at each point. On the left, an enlarged version near the boundary, where the jumps in displacements can be easily observed.

whose constitutive equation is given by equation 60. The resulting elastic deformation is shown in figure 20. The presence of jumps near the boundary are also shown therein.

6 Summary

Motivated to create a numerical method to solve boundary value problems on complex domains, we defined a class of asymptotically consistent methods which have guaranteed convergence properties. These methods are distinguished by the fact that there is considerable freedom in the various ap-

proximations involved. For example, the domain of the problem may be approximated in any way as long as $|\Omega \setminus \Omega_h| + |\Omega_h \setminus \Omega|$ is $\mathcal{O}(h)$ or smaller. This immediately suggests that it should be possible to come up with a simple asymptotically consistent method, which is precisely what we did.

The DG-IBM framework presented here has the distinguishing feature that it determines numerical approximations to the exact solution using a simple mesh, which may not conform to the complex problem domain. This drastically reduces the cost of creating boundary fitting meshes. In order to ensure that the numerical solution so computed converges to the exact solution with an optimal rate, we used specific approximations of the domain, function space and the boundary conditions. The domain was approximated using the linear interpolant of a level set function and a discontinuous Galerkin approximation was used in the elements cut by the boundary to avoid boundary locking. These ideas were discussed in detail in [27].

However, imposing nonhomogeneous boundary conditions on the approximate boundary is far from obvious, especially without explicitly compute correspondences between the exact and approximate boundaries (which is required when using curved elements). We addressed two issues- we defined where and what to impose as Dirichlet and Neumann boundary conditions on the approximate boundary. These boundary conditions were defined such that a better approximation of the problem domain implied a better approximation of the boundary conditions and hence a more accurate numerical solution.

Using numerous simlations, the proposed numerical extension as well the analytical extension method for boundary conditions were tested with two and three dimensional problems in elasticity. For a combination of Dirichlet and Neumann boundary conditions, the numerical solution was observed to converge to the exact one with the optimal rate in the L_2 norm, inspiring confidence in the extension procedure. Two nonlinear elasticity examples on complex domains, namely the femur and the Stanford "S", provide a glimpse of the novel applications DG-IBM can be used for.

Apart from the encouraging numerical results, there are a lessons to be learnt from the discussions presented in this paper. For a start, DG-IBM provides an excellent example of a method that computes good approximations to the exact solution even though almost none of the data in the given problem may be used exactly, not the domain, not even the boundary conditions. The problem of having to extend boundary conditions to an approximate boundary, which may not be an interpolant of the exact one, is commonly encountered in both finite element methods as well as in many immersed boundary methods. The elegant solution we have proposed- using functions that can contain discontinuities near the boundary and imposing boundary conditions that are extensions of the exact ones while strictly imposing Dirichlet boundary con-

ditions, may be suitable for many other numerical methods as well.

While we have not justified why the DG-IBM formulation presented here is asymptotically consistent, a companion paper showing the same will be communicated shortly. At the same time, we highlight that this is just a specific example of an asymptotically consistent method. There is a whole class of such methods and perhaps one more suitable than the other for a given application. For example, it is trivial to show that the standard finite element method falls into this category as well. A good understanding of the underlying mathematics will no doubt help tailor newer and higher order numerical schemes that can be used to tackle challenging engineering problems.

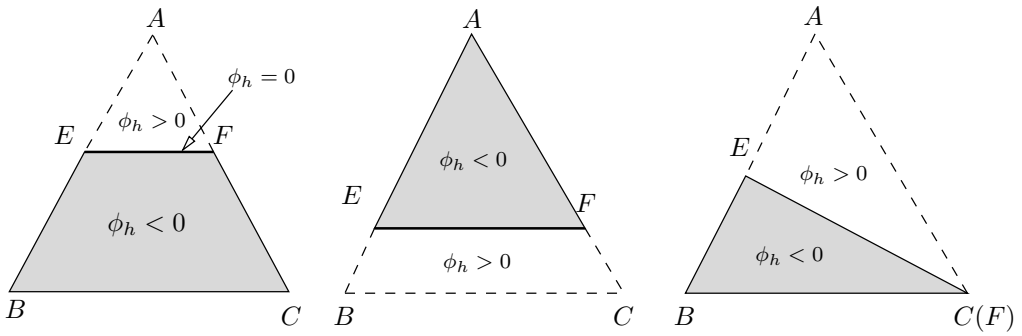


Fig. 21. Intersection of a triangle element with a straight segment (zero level sets of ϕ_h). The "cut" elements are either triangles or quadrilaterals.

Appendix- An Implementation of DG-IBM

A simple implementation of DG-IBM using linear triangle and tetrahedral elements for two and 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 [27] and is included here for the sake of completeness.

Element Boundary intersections:

As mentioned earlier, Γ_h is composed of linear segments in 2D and plane sections in 3D domains. Figure 21 depicts line-triangle intersections possible in 2D and figure 22 shows the plane-tetrahedron intersections possible in 3D. Referring to $E \cap \Omega_h$ for $E \in \mathcal{Q}_h$ as the "cut element", we see that cut elements are either triangles or quadrilaterals in 2D. In 3D, cut elements possible are tetrahedra, wedges with six nodes and pyramids having five nodes. Also note that Γ_h^E is a triangle for tetrahedra, pyramids and wedges of type (a), and a quadrilateral for type (b) wedges.

Basis Functions The standard finite element basis functions are used for elements belonging to $\mathcal{R}_h$. For elements in $\mathcal{Q}_h$ 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 Γ_h^E. Hence we use the first shape function as the level set function itself. The remaining shape functions interpolate displacements at points of Γ_h^E.

For the linear triangle elements belonging to $\mathcal{Q}_h$, we have seen that Γ_h^E is a

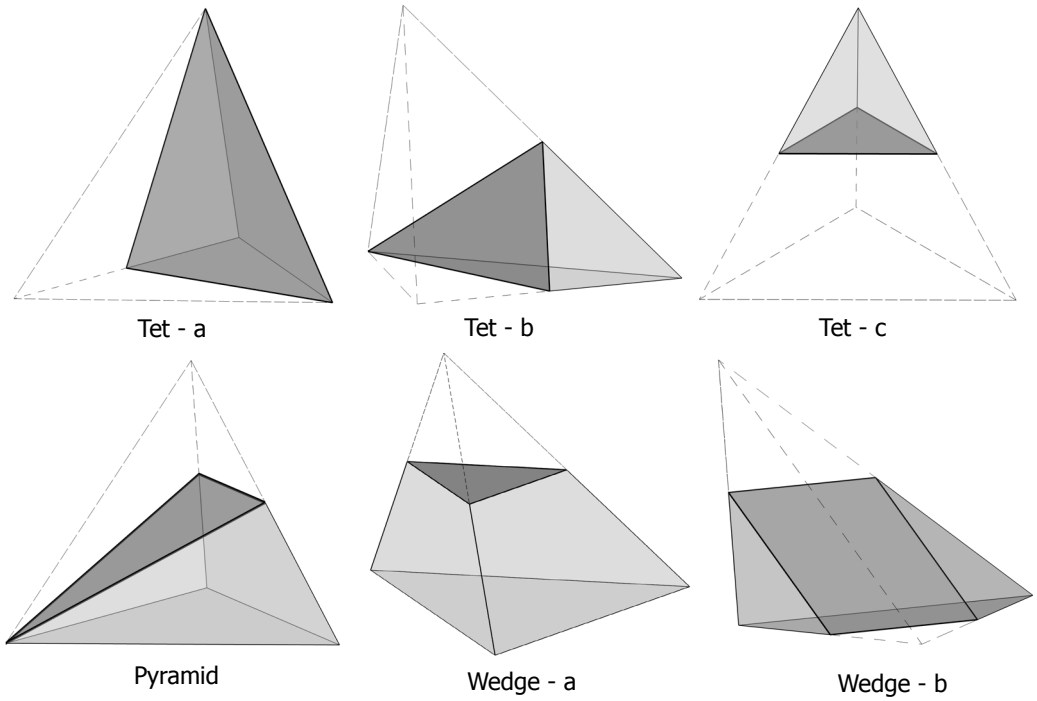


Fig. 22. Types of plane-tetrahedron intersections; the plane represents the zero level sets of ϕ_h. The "cut" element can be tetrahedra, pyramids or wedges. Note that Γ_h^E is a triangle excepts in the case of the cut element being a wedge of type (b).

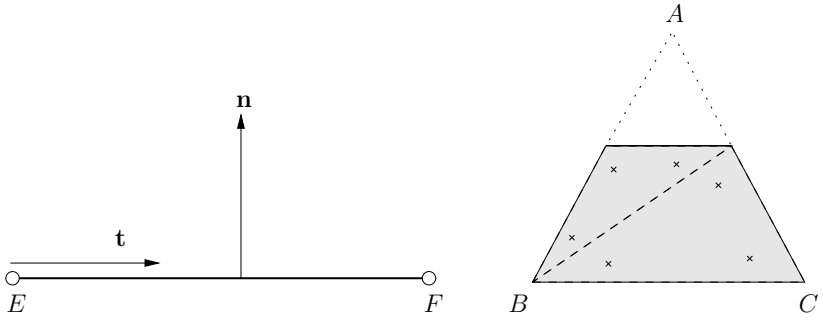


Fig. 23. Illustration for shape functions and quadrature rules in cut triangles. In elements of $\mathcal{Q}_h$, Γ_h^E is a straight segment EF. The shape functions N_E and N_F interpolate displacements at points E and F respectively while N_n is zero on EF. Shown on the right is the quadrature rules for a quadrilateral cut element. The quadrilateral is divided into two triangles (only for the purpose of integration) and a three point rule used in each.

straight segment EF. The basis functions for these elements are chosen as

$$\begin{aligned} N_n &= (\mathbf{x}_E - \mathbf{x}) \cdot \mathbf{n}, \\ N_E &= (\mathbf{x} - \mathbf{x}_F) \cdot \mathbf{t} \quad \text{and} \\ N_F &= (\mathbf{x} - \mathbf{x}_E) \cdot \mathbf{t}, \end{aligned} \tag{96}$$

where $\mathbf{n}$ is the unit normal to segment EF and $\mathbf{t}$ is a unit vector along EF as

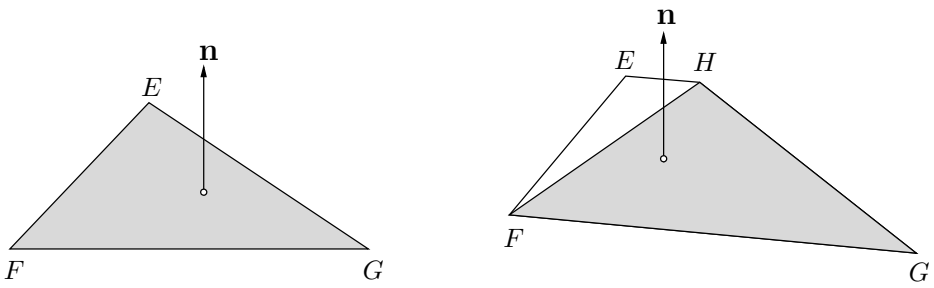


Fig. 24. For tetrahedral elements cut by the boundary, N_E, N_F and N_G interpolate the displacements at points E, F and G. Shape function N_n is zero on Γ_h^E.

shown in figure 23.

For linear tetrahedral elements cut by the boundary, we saw that Γ_h^E is either a triangle or a quadrilateral. The first shape function is one that is zero on this plane. In the case of Γ_h^E being a triangle, the remaining three shape functions are ones interpolate displacements at the vertices of the triangle. In the case of a quadrilateral, we interpolate the displacements at the vertices of the triangle formed by the vertices that has the largest area. Let the final three shape functions interpolate the displacements at vertices E, F and G on Γ_h^E. The four shape functions are then

$$\begin{aligned} N_n &= (\mathbf{x} - \mathbf{x_E}) \cdot \mathbf{n}, \\ N_E &= \frac{1}{\Delta EFG}(\mathbf{x} - \mathbf{x_F}) \cdot (\mathbf{x_F} - \mathbf{x_G}) \times \mathbf{n}, \\ N_F &= \frac{1}{\Delta EFG}(\mathbf{x} - \mathbf{x_G}) \cdot (\mathbf{x_G} - \mathbf{x_E}) \times \mathbf{n} \quad \text{and} \\ N_G &= \frac{1}{\Delta EFG}(\mathbf{x} - \mathbf{x_E}) \cdot (\mathbf{x_E} - \mathbf{x_F}) \times \mathbf{n}, \end{aligned} \tag{97}$$

where $\Delta EFG = (\mathbf{x_E} - \mathbf{x_F}) \cdot (\mathbf{x_F} - \mathbf{x_G}) \times \mathbf{n}$ and $\mathbf{n}$ is the unit normal to the plane of Γ_h^E, see figure 24.

Quadrature rules: Standard quadrature rules can be used for elements of $\mathcal{R}_h$. Some additional care is required for elements of $\mathcal{Q}_h$ because they require the evaluation of lifting operators and hence integration of quadratic polynomials over cut elements. Elements of $\mathcal{M}_h$ also require quadrature rules for integration over faces that are shared with other elements (once again to compute lifting operators). We outline an integration scheme to perform all the necessary integrals exactly and efficiently.

In the 2D case, a three point rule is used for elements in $\mathcal{R}_h$ and for triangular cut elements of $\mathcal{Q}_h$. When the cut element is a quad, the quad is divided into two triangles and three point rules used over each of them. Note that this division is solely for the purpose of integration, see figure 23. A two point rule is used for integration over faces of elements in $\mathcal{M}_h$ to compute

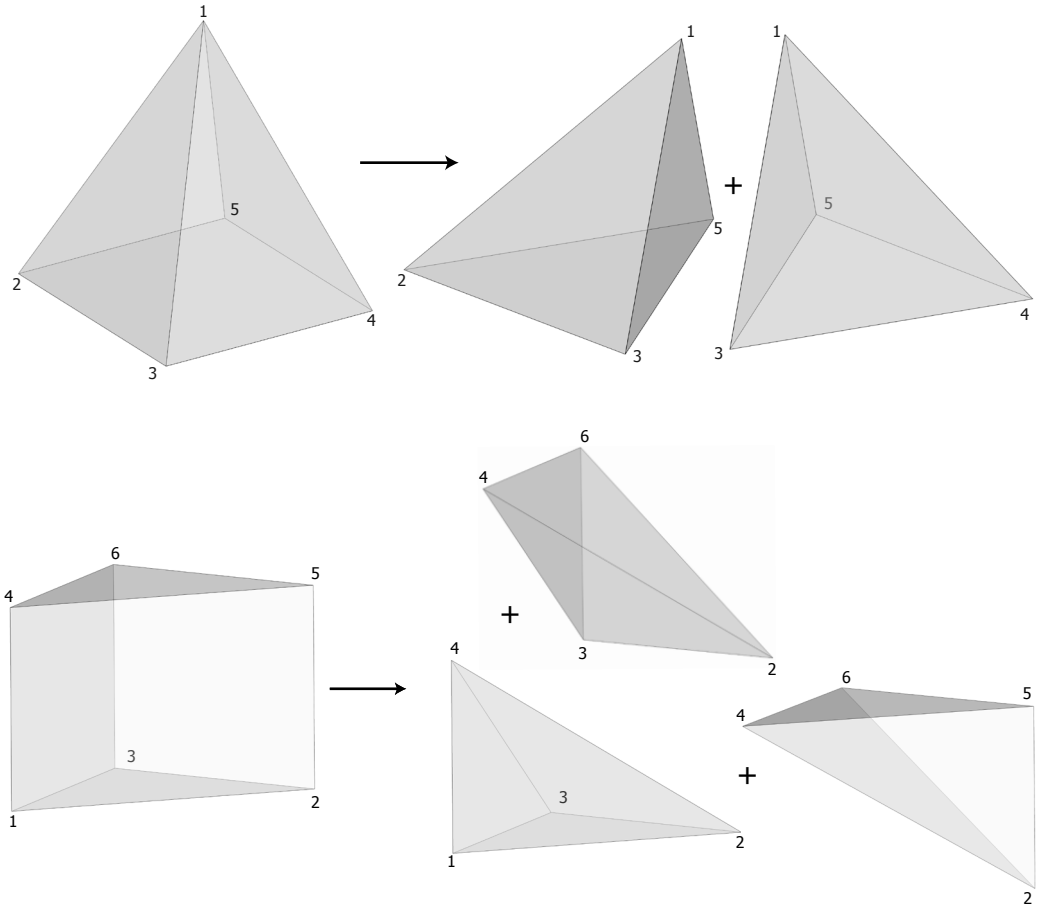


Fig. 25. For the purpose of integration, pyramids are subdivided into two and wedges into three tetrahedra. A four point rule is used in each tetrahedron.

the lifting operators. For the 3D case, a four point rule is used for integration over tetrahedral domains. Pyramids and wedges are subdivided into two and three tetrahedra respectively, see figure 25. A four point rule is used in each tetrahedron resulting in a 8 point rule for pyramids and a 12 point rule for wedges. Three quadrature points are used for integration over triangular faces and four points for quad faces.

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.

References

[1] R. Adams and J. Fournier. *Sobolev Spaces*. Academix Press, 2003.

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

[3] I. Babuška. The finite element method with Lagrangian multipliers. *Numerische Mathematik*, 20:179–192, 1973.

[4] I. Babuška. The finite element method with penalty. *Mathematics of Computation*, 27:221–228, 1973.

[5] I. Babuška, U. Banerjee, and J. Osborn. Survey of meshless and generalized finite element methods: A unified approach. *Acta Numerica*, 12:1–125, 2003.

[6] H. Barbosa and T.J.R. Hughes. The finite element method with Lagrange multipliers on the boundary: Circumventing the Babuška-Brezzi condition. *Computer Methods in Applied Mechanics and Engineering*, 85:109–128, 1991.

[7] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47(45), 1996.

[8] Ted Belytschko, Nicolas Moës Chandu Parimi, N.Sukumar, and Shuji Usui. Structured extended finite element method for solids defined by implicit surfaces. *International Journal for Numerical Methods in Engineering*, 56:609–635, 2002.

[9] A. Berger. l^2 error estimates for finite elements with interpolated boundary conditions. *Numer. Math.*, 21:345–349, 1973.

[10] A. Berger, R. Scott, and G. Strang. Approximate boundary conditions in the finite element method. In *Symposia Mathematica*, volume X, pages 757–761. Academic Press, New York, 1972.

[11] J.J. Blair. Higher order approximations to the boundary conditions for the finite element method. *Mathematics of Computation*, 30(134):250–262, April 1976.

[12] J.H. Bramble and J.T. King. A robust finite element method for nonhomogeneous dirichlet problems in domains with curved boundaries. *Mathematics of Computation*, 63(207):1–17, June 1994.

[13] S.C. Brenner and Scott L.R. *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, and 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*, pages 67–76, New York, NY, USA, 2001. ACM.

[15] P.G. Ciarlet and P.A. Raviart. Interpolation theory over curved elements, with applications to finite element methods. *cmame*, 1:217–249, 1972.

[16] D. Clarke, M. Salas, and H. Hassan. Euler calculations for multi-elements airfoils using cartesian grids. *AIAA Journal*, 24:1128–1166, 1986.

[17] C. Engwer and P. Bastian. A discontinuous galerkin method for simulations in complex domains. Preprint(http://www/ub.uniheidelberg.de/archiv/5707/), 2005.

[18] R. Fedwik and X.D. Liu. *Innovative methods for Numerical Solutions of Partial differential equations*, chapter The Ghost fluid method for Viscous flows, pages 111–143. World Scientific Publishing, New Jersey, 2002.

[19] S. Fernández-Méndez and A. Huerta. Imposing essential boundary conditions in mesh–free methods. *Computer Methods in Applied Mechanics and Engineering*, 193:1257–1275, 2004.

[20] R. Glowinski, T.-W. Pan, T. Hesla, and D. Joseph. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *Int. J. Multiphase Flow*, 25:755–794, 99.

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

[22] B. Griffith and C. Peskin. On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems. *J. Comput. Phys.*, 208:75–105, 2005.

[23] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

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

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

[26] M. Lenoir. Optimal Isoparametric Finite Elements and Error Estimates for Domains Involving Curved Boundaries. *SIAM Journal on Numerical Analysis*, 23(3):562–580, 1986.

[27] A. Lew and G. Buscaglia. A discontiuous-galerkin based immersed boundary method. *Int. J. Numerical Methods in Engineering*, 2007. in press.

[28] X.D. Liu and C.T. Sideris. Convergence of the ghost fluid method for elliptic equations with interfaces. *Mathematics of Computation*, 72(244):1731–1746, 2003.

[29] 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.

[30] S. Mauch. *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. PhD thesis, California Institute of Technology, 2003.

[31] Moës, N., Béchet, E., and M. Tourbier. Imposing Dirichlet boundary conditions in the extended finite element method. *Int. J. Numer. Meth. Eng.*, 67:1641–1669, 2006.

[32] Sukumar N. and Moran B.and Belytschko T. The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering*, 43:839–887, 1998.

[33] C. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25:220–252, 1977.

[34] V.L. Rvachev, T.I. Sheiko, V. Shapiro, and I. Tsukanov. Completeness of RFM solution structures. *Computational Mechanics*, 25:305–316, 2000.

[35] V.L. Rvachev, T.I. Sheiko, V. Shapiro, and I. Tsukanov. Transfinite interpolation over implicityly defined sets. *Computational Mechanics*, 18:195–220, 2001.

[36] R. Scott. *Finite Element Techniques for Curved Boundaries*. PhD thesis, M.I.T, 1973.

[37] R. Scott. Intrpolcated boundary conditions in the finite element method. *siam*, 12(3):404–427, June 1975.

[38] V. Shapiro. Real functions for representations of rigid solids. *Computer Aided Geometric Design*, 11, 1994.

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

[40] G. Strang and A. Berger. The change in solution due to change in domain. AMS Symposium on Partial Differential Equations, Berkeley, 1971.

[41] A. TenEyck and A. Lew. Discontinuous galerkin methods for nonlinear elasticity. *Int. J. Numerical Methods in Engineering*, 67:1204–1243, 2006.

[42] V. Thomée. Polygonal domain approximation in dirichlet's problem. *J. Inst. Maths Applics*, 11:33–44, 1973.

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

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

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

[46] Krongauz Y. and Belytschko T. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering*, 131:133–145, 1996.

[47] M. Zlámal. Curved elements in the finite element method. I. *SIAM J. Numerical Analysis*, 10(1):229–240, 1973.

[48] M. Zlámal. Curved elements in the finite element method. II. *SIAM J. Numerical Analysis*, 11(2):347–362, 1973.

[49] M. Zlámal. The finite element method in domains with curved boundaries. *Int. J. Numerical Methods in Engineering*, 5:367–373, April 1973.