http://dx.doi.org/10.1016/j.jcp.2019.109123

Recovering Elastic Inclusions by Shape Optimization Methods with Immersed Finite Elements *

Ruchi Guo † Tao Lin ‡ Yanping Lin §

Abstract

This article presents a finite element method on a fixed mesh for solving a group of inverse geometric problems for recovering the material interface of a linear elasticity system. A partially penalized immersed finite element method is used to discretize both the elasticity interface problems and the objective shape functionals accurately regardless of the shape and location of the interface. Explicit formulas for both the velocity fields and the shape derivatives of IFE shape functions are derived on a fixed mesh and they are employed in the shape sensitivity framework through the discretized adjoint method for accurately and efficiently computing the gradients of objective shape functions with respect to the parameters of the interface curve. The shape optimization for solving an inverse geometric problem is therefore accurately reduced to a constrained optimization that can be implemented efficiently within the IFE framework together with a standard optimization algorithm. We demonstrate features and advantages of the proposed IFE-based shape optimization method by several typical inverse geometric problems for linear elasticity systems.

Keywords: Inverse problems, Elasticity systems, Inclusions reconstruction, Discontinuous Lamé parameters, Shape optimization, Immersed finite element methods.

1 Introduction

Elasticity inverse problems have attracted attentions because of their appearances in quite a few applications such as elasticity imaging used for non-destructive testing of mechanical properties/structures in geophysics exploration and medical diagnosis [9, 29, 64]. In general, elasticity inverse problems can be categorized into inverse geometric problems and parameter estimation problems. An inverse geometric problem is for reconstructing geometric structures inside an elastic body, such as inclusions, cavities (rigid inclusions) and cracks [6, 7, 8, 42, 55], and a parameter estimation problem is for the determination of distributed parameters, such as elastic moduli and mass density [24, 25, 30, 71]. It is well known that these inverse problems are usually ill-posed due to the lack of information. We refer readers to the review article [15] for various inverse problems in the context of elasticity and the related mathematical or numerical techniques.

The purpose of this article is to develop a numerical method for recovering inclusions in an elastic body with different elastic properties than the surrounding materials through an IFE-based shape optimization method on a fixed mesh. Without loss of generality, consider an elastic body Ω that is partitioned by an interface Γ into two subdomains Ω^- and Ω^+ filled with different materials, and we assume the buried inclusion occupies the subdomain Ω^-. Due to the changes in elastic properties across the material interface Γ, and without loss of generality, the Young's modulus E and Poisson's ratio ν in Ω are assumed to be piecewise constant functions:

$$E(X) = E^s, \quad \nu(X) = \nu^s, \quad X \in \Omega^s, \quad s = \pm, \tag{1}$$

*This research was partially supported by GRF 15301714/15327816 of HKSAR, and Polyu AMA-JRI

†Department of Mathematics, Virginia Tech, Blacksburg, VA 24061 (ruchi91@vt.edu)

‡Department of Mathematics, Virginia Tech, Blacksburg, VA 24061 (tlin@vt.edu)

§Department of Applied Mathematics, Hong Kong Polytechnic University, Kowloon, Hong Kong, China (yanping.lin@polyu.edu.hk)

where $X = (x_1, x_2)^t$ with the superscript t for the transpose. Correspondingly, the Lamé parameters in Ω are also piecewise constants:

$$\begin{aligned} &\lambda(X) = \lambda^s(X), \quad \mu(X) = \mu^s(X), \quad X \in \Omega^s, \quad s = \pm, \\ \text{with} \qquad &\lambda^s(X) = \frac{E^s \nu^s}{(1+\nu^s)(1-2\nu^s)}, \quad X \in \Omega^s, \qquad \mu^s(X) = \frac{E^s}{2(1+\nu^s)}, \quad X \in \Omega^s, \quad s = \pm. \end{aligned} \tag{2}$$

Assume that a set of body forces and boundary conditions $\mathbf{f}^k, \mathbf{g}_D^k, \mathbf{g}_N^k, k = 1, 2, \cdots, K$ simulating unrelated mechanisms are applied on this elastic body Ω, then the corresponding displacement fields $\mathbf{u}^k = (u_1^k, u_2^k)^t, k = 1, 2, \cdots, K$ are determined by the following direct problem for the linear elasticity system:

$$-\operatorname{div} \sigma(\mathbf{u}^k) = \mathbf{f}^k, \quad \text{in } \Omega^- \cup \Omega^+, \tag{3a}$$

$$\mathbf{u}^k = \mathbf{g}_D^k, \quad \text{on } \partial\Omega_D^k, \quad \text{and} \quad \sigma(\mathbf{u}^k)\mathbf{n} = \mathbf{g}_N^k \quad \text{on } \partial\Omega_N^k, \tag{3b}$$

with the stress tensor $\sigma(\mathbf{u}) = (\sigma_{ij}(\mathbf{u}))_{i,j=1,2}$ and strain tensor $\epsilon(\mathbf{u}) = \epsilon_{ij}(\mathbf{u})$:

$$\sigma_{ij}(\mathbf{u}) = \lambda(\nabla \cdot \mathbf{u})\delta_{i,j} + 2\mu\epsilon_{ij}(\mathbf{u}), \qquad \epsilon_{ij}(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{4}$$

where $\overline{\partial\Omega_D^k} \cup \overline{\partial\Omega_N^k} = \partial\Omega$, and $\mathbf{n}$ is the outward normal vector to $\partial\Omega$. In addition, the displacement fields $\mathbf{u}^k$'s are assumed to satisfy jump conditions across the material interface Γ:

$$[\mathbf{u}^k]_\Gamma := \mathbf{u}^{k,+} - \mathbf{u}^{k,+} = \mathbf{0}, \quad [\sigma(\mathbf{u}^k)\mathbf{n}]_\Gamma := \sigma^+(\mathbf{u}^{k,+})\mathbf{n} - \sigma^-(\mathbf{u}^{k,-})\mathbf{n} = \mathbf{0}, \quad k = 1, \cdots, K, \tag{5}$$

where $\mathbf{n}$ is normal to Γ.

In many applications, the values of material parameters are known a-priori, such as the elastic properties of tissue and bone in medical problems [59, 67]. Hence, assuming the Young's modulus E^s, ν^s, $s = \pm$, are given, our focus in this work is to numerically solve the inverse geometric problem for recovering the shape and location of the material interface Γ in the linear elasticity systems (3)-(5) by only boundary measurements or measurements close to the boundary through a shape optimization method [40, 72]. Shape optimization methods have been applied to various inverse geometric problems such as the electric impedance tomography (EIT), the related electric resistance tomography (ERT), electrical capacitance tomography (ECP) [12, 23, 73], and the optical tomography [1]. In particular, the authors in [71] studied simultaneous identification of both Lamé parameters and a circular inclusion provided boundary measurements. Obstacle identification problems are studied in [8, 17] also with boundary measurements. In [42], the authors proposed an algorithm to simultaneously recover both the inclusions of general shape and Lamé parameters but with full internal measurements. We would like to mention that the authors in [7] studied an inclusion reconstruction problem close to the one we investigate in this article but with a different jump condition across interface which can be understood as extra measurements. In addition, the inverse problem considered here has a certain similarity to the electrical impedance tomography governed by scalar elliptic equations which seems to be a popular topic in the literature. Nevertheless, to the best of our knowledge, there are relatively fewer publications on the inverse problem in the specific form discussed in the present article.

By the shape optimization approach, we formulate the inverse geometric problem as finding the material interface Γ^* such that

$$\Gamma^* = \operatorname{argmin} \mathcal{J}(\mathbf{u}^1(\Gamma), \mathbf{u}^2(\Gamma), \cdots, \mathbf{u}^K(\Gamma), \Gamma), \tag{6}$$

$$\text{with} \quad \mathcal{J}(\mathbf{u}^1(\Gamma), \mathbf{u}^2(\Gamma), \cdots, \mathbf{u}^K(\Gamma), \Gamma) = \int_{\Omega_d} J(\mathbf{u}^1(\Gamma), \mathbf{u}^2(\Gamma), \cdots, \mathbf{u}^K(\Gamma); X, \Gamma) dX, \tag{7}$$

where the displacement fields $\mathbf{u}^k$, $1 \leqslant k \leqslant K$ satisfy the elasticity systems described by (3)-(5), but the shape functional $\mathcal{J}$, the related integrand J, the data/measurement region $\Omega_d \subseteq \overline{\Omega}$ depend on the data available for recovering the material interface. In general, the shape functionals should achieve the minimal values when the

interface is the exact one. In practice, the solutions $\mathbf{u}^k$, $1 \leqslant k \leqslant K$, will be used to compute the simulated measurements which are further used to be compared with the actual measurements. Then the construction of such shape functionals satisfies the following principle: the minimal value is achieved at an interface when the simulated measurements agree with the actual measurements. We refer readers to [15] for a detailed discussion of this principle and some examples as well as other more sophisticated techniques in construction. In this article, some representative examples of shape functionals will be given in Section 4 for various typical applications.

A good numerical method for the shape optimization (6)-(7) requires a reliable approximation to the displacement fields $\mathbf{u}^k$, $k = 1, \cdots, K$ for which one needs to solve the direct problems (3)-(5) with discontinuous coefficients. Traditional finite element methods can be used to solve for $\mathbf{u}^k$, $k = 1, \cdots, K$ from (3)-(5) provided that the mesh keeps updating to conform with the modified interface in each iteration in the shape optimization; otherwise, such a direct solver may produce unsatisfactory results [11]. Shape optimization methods based on conforming meshes are referred as the Lagrange approach in some references [21], and we refer readers to [12, 14, 71, 73] for the application of this approach to some inverse geometric problems. An obvious drawback of a Lagrange type shape optimization method is its computational cost to resolve the interface by the mesh again and again in the optimization iterations, especially when the interface becomes complicated.

In addition, since there is no explicit mathematical rules relating the mesh nodes and the design variables for general auto mesh generators, it usually needs some special approximation techniques and requires a global computation on Ω [21] to construct a velocity field within the Lagrange framework. This procedure is often burdensome and time consuming. Also, according to [20], an inappropriate choice of velocity field for mesh updating may result in a distorted mesh leading to inaccurate finite element solutions for the displacement fields. Here, a velocity field is the derivatives of spatial variables X with respect to the design variables [21], i.e., the parameters determining the interface curve. It is needed for computing the shape sensitivity which is a necessary ingredient for minimizing the shape functional by a gradient-type numerical optimization algorithm.

Therefore, direct problem solvers that can handle discontinuous coefficients on a fixed mesh are desirable alternatives for shape optimization applications, and shape optimization methods based on a fixed mesh are referred as the Eulerian methods in some literatures [53]. One group of the Eulerian type shape optimization methods [1, 17, 19, 23] use a local averaging method [18] to smooth the discontinuous coefficients such that standard finite element methods can be applied on a fixed mesh. Another group of Eulerian type shape optimization methods employ special finite difference/element methods such as the immersed interface methods (IIM) in [46] for cavity reconstruction in scalar elliptic equations and the extended finite element methods (XFEM) in [61, 68, 75] for crack detection.

In this article, we propose an Eulerian type shape optimization method based on the immersed finite elements (IFE) [32, 33, 35, 56, 57, 58] for solving the inverse geometric problem of the linear elasticity system described by (3)-(5). In the proposed method, we parameterize the interface curve Γ and use the coordinates of its control points as the design variables in the shape optimization. We discretize the elasticity interface (direct) problems (3)-(5) by a partially penalized IFE (PPIFE) method [36] using the IFE space developed in [35]. Regardless of the interface location in a chosen mesh, the numerical displacement fields produced by this IFE method are optimal approximations to the displacement fields described by (3)-(5). Consequently, the objective functional (7) in many applications, such as those reported in Section 4, is also discretized optimally by the IFE solution on the chosen fixed mesh. These optimal discretizations are then naturally used to reduce the shape optimization described by (6)-(7) subject to the linear elasticity system (3)-(5) to a fully discrete constrained optimization problem.

Furthermore, the fixed mesh used by the IFE methods enables us to introduce a velocity field that vanishes over all the non-interface elements. And the intrinsic dependence of IFE shape functions on the interface allows us to derive explicit formulas for their shape derivatives which are also non-zero only on interface elements. Since the number of interface elements is only in the order of $\mathcal{O}(h^{-1})$ in a shape regular mesh and the union of all interface elements form a small band around the interface for a fine mesh, these two fundamental components together with a discretized adjoint method can be efficiently implemented in the standard material derivative formula for accurately and efficiently computing the shape sensitivities. Moreover, the interface parameterization in the proposed method results in an optimization algorithm to search the target interface in

a relatively small space in order for a quick and good reconstruction. We believe this can be understood as a "hidden" regularization mechanism of the proposed method.

We believe, these features, i.e., (a) accurate discretization for both the elasticity systems and the cost functional on a fixed mesh, (b) the accurate and efficient computation for the shape sensitivities of the cost function on a fixed mesh, and (c) the self/implicit regularization mechanism, collectively distinguish the proposed IFE-based shape optimization method from other methods in the literature.

This article consists of four additional sections. In the next section, we provide a brief review of the vector IFE shape functions and the PPIFE method for the direct elasticity interface problems. In Section 3, we propose the IFE-based shape optimization for the inverse geometric problems for the elasticity system. Key ingredients for the IFE-based shape optimization such as the velocity fields on interface independent mesh, the shape derivatives of IFE shape functions, and the material derivatives of objective functions will be derived in this section. Section 4 presents numerical examples for a class of typical inverse problems to demonstrate the strength and versatility of the proposed IFE-based shape optimization method. The last presents some conclusion remarks.

2 An IFE Method for the Elasticity Interface Direct Problems

In this section, we first describe our parameterization approach of the interface as it is a key component of the proposed shape optimization algorithm. Then based on this parameterization, we proceed to recall the IFE space developed in [35] according to the jump conditions (5), and then recall the PPIFE method [36] for the discretization of the elasticity interface problems (3)-(5) on an interface independent mesh. We will put the PPIFE discretization in a matrix form suitable for shape optimization.

In this work, we employ the cubic spline to parameterize the interface curve because of its accuracy, versatility, and popularity. Given a series of points $Y_j = (y_1^j, y_2^j)$, $j = 1, 2, \cdots, N$, with a positive integer N, we use their coordinates to form a vector $\boldsymbol{\alpha} = (y_1^1, \cdots, y_1^N, y_2^1, \cdots, y_2^N)$. On a reference interval $I = [0, 1]$, we divide I into $N-1$ subintervals: $I_j = [(j-1)/(N-1), j/(N-1)]$, $j = 1, 2, \cdots, N-1$. Then the interface Γ corresponding to the points $\{Y_j\}_{j=1}^N$ is the parametric curve $(y_1(\tau), y_2(\tau)), \tau \in [0, 1]$ such that on each sub-interval I_j, $j = 1, 2, \cdots, N-1$,

$$y_l(t) = y_l^j(t) := a_l^j t^3 + b_l^j t^2 + c_l^j t + d_l^j, \quad l = 1, 2. \tag{8}$$

Here the coefficients a_l^j, b_l^j, c_l^j and $d_l^j, 1 \le j \le N-1, l = 1, 2$ are constructed such that the function $y_l(t), l = 1, 2$ is the cubic spline interpolations of the corresponding coordinates in $Y_j = (y_1^j, y_2^j)$, $j = 1, 2, \cdots, N$ and satisfy the end point matching condition, i.e.,

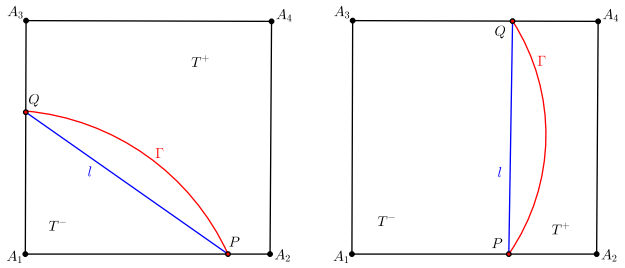
$$(y_l^1)^k(0) = (y_l^{N-1})^k(1), \quad j = 1, 2, \cdots, N-2, \quad l = 1, 2, \quad k = 0, 1, 2.$$

It well known that, as cubic splines, functions $y_l(t), l = 1, 2$ have continuous derivatives up to the second order. This so called front-tracking approach is also used in [37] for the discretization of interface. By this set-up, we therefore denote the interface curve as $\Gamma = \Gamma(t, \boldsymbol{\alpha})$, $t \in [0, 1]$, and we note that the shape and location of Γ depends and only depends on the points $Y_j = (y_1^j, y_2^j)$, $j = 1, 2, \cdots, N$, i.e., the vector $\boldsymbol{\alpha}$. Following [13], we call Y_j the control points and call the entries of $\boldsymbol{\alpha}$ the design variables. We note that the design variables are basically the coordinates of control points and they will be used as variables for shape optimization. We refer readers to numerical results presented in Figures 3-6 for a visualization of these control points. For the sake of simplicity, in the rest of this article, we will denote the vector $\boldsymbol{\alpha} = (\alpha_i)_{i \in \mathcal{D}}$ with $\mathcal{D} = \{1, 2, \cdots, 2N\}$. Also we emphasize that the proposed method works readily with other parameterizations.

Without loss of generality, we consider a rectangular domain Ω and let $\mathcal{T}_h$ be a fixed Cartesian mesh for this domain. The sets of nodes and interior nodes are denoted by $\mathcal{N}_h = \{X_1, X_2, \cdots, X_{|\mathcal{N}_h|}\}$ and $\mathring{\mathcal{N}}_h$, respectively. We call $T \in \mathcal{T}_h$ an interface element if the interior of T intersects the interface $\Gamma(t, \boldsymbol{\alpha})$; otherwise, we call it a non-interface element. Similarly, we define the interface and non-interface edges. Denote $\mathcal{T}_h^i$ ($\mathcal{E}_h^i$) and $\mathcal{T}_h^n$ ($\mathcal{E}_h^n$) as the sets of interface and non-interface elements (edges), respectively. Furthermore, let $\mathring{\mathcal{E}}_h^i$ be the set of interior interface edges.

We start from recalling the vector bilinear IFE space on a mesh $\mathcal{T}_h$ discussed in [35]. On each element $T = \square A_1A_2A_3A_4 \in \mathcal{T}_h$, we define an index set $\mathcal{I} = \{1, 2, 3, 4\}$. Let $\psi_{i,T}^{non}, i \in \mathcal{I}$ be the standard scalar Lagrange bilinear shape functions associated with the vertices of T, and let $\boldsymbol{\psi}_{i,T}^{non}, i = 1, \cdots, 8$ be the corresponding vector bilinear shape functions on T [16], i.e., $\boldsymbol{\psi}_{i,T}^{non}(A_j) = [\delta_{i,j}, 0]^t$, $\boldsymbol{\psi}_{i+4,T}^{non}(A_j) = [0, \delta_{i,j}]^t$, $i, j \in \mathcal{I}$. Further denote the matrix shape functions: $\Psi_{i,T}^{non} = [\boldsymbol{\psi}_{i,T}^{non}, \boldsymbol{\psi}_{i+4,T}^{non}]$, $i \in \mathcal{I}$. Then the local IFE space $\mathbf{S}_h(T)$ on every non-interface element $T \in \mathcal{T}_h^n$ is defined as

$$\mathbf{S}_h(T) = \text{Span}\{\boldsymbol{\psi}_{i,T}^{non},\ i = 1, \cdots, 8\} = [\mathbb{Q}_1(T)]^2\,. \tag{9}$$



(a) Case 1 interface element (b) Case 2 interface element

Figure 1: Typical bilinear elements

On an interface element $T \in \mathcal{T}_h^i$ with the edge length h, we denote the two interface-mesh intersection points of ∂T and the interface $\Gamma(t, \boldsymbol{\alpha})$ by $P = (x_P, y_P)^t$ and $Q = (x_Q, y_Q)^t$, and let l be the line connecting P and Q whose normal and tangential vectors are as follows:

$$\bar{\mathbf{n}} = \frac{1}{||P - Q||}(y_P - y_Q, -(x_P - x_Q))^t, \quad \bar{\mathbf{t}} = \frac{1}{||P - Q||}(x_P - x_Q, y_P - y_Q)^t. \tag{10}$$

Then the equation for the line l is $L(X) = 0$ with $L(X) = \bar{\mathbf{n}} \cdot (X - P)$. Denote T^+ and T^- as the two sub-elements partitioned by the line l, see the illustrations in Figure 1. We divide $\mathcal{I}$ into two subsets $\mathcal{I}^- = \{i : A_i \in T^-\}$ and $\mathcal{I}^+ = \{i : A_i \in T^+\}$. We recall from [35] that the vector bilinear IFE function with the nodal value $\mathbf{v}_i = (v_i^1, v_i^2)^t$ at the node A_i, $i \in \mathcal{I}$ assumes the format of piecewise polynomials:

$$\boldsymbol{\psi}_T^{int}(X) = \begin{cases} \boldsymbol{\psi}_T^{int,-}(X) = \boldsymbol{\psi}_T^{int,+}(X) + L(X)\mathbf{c}_0 & \text{if} \quad X \in T^-, \\ \boldsymbol{\psi}_T^{int,+}(X) = \displaystyle\sum_{i \in \mathcal{I}^+} \Psi_{i,T}^{non}(X)\mathbf{v}_i + \sum_{i \in \mathcal{I}^-} \Psi_{i,T}^{non}(X)\mathbf{c}_i & \text{if} \quad X \in T^+, \end{cases} \tag{11}$$

where the coefficients $\mathbf{c}_0 = (c_0^1, c_0^2)^t$, $\mathbf{c}_i = (c_i^1, c_i^2)^t$, $i \in \mathcal{I}^-$ are determined by

$$\mathbf{c}_0 = K^{-1}\hat{\sigma}(\boldsymbol{\psi}_T^{int,+})(F_0)\bar{\mathbf{n}}, \quad \text{with} \quad K = R \begin{bmatrix} (\lambda^- + 2\mu^-) & 0 \\ 0 & \mu^- \end{bmatrix} R^t, \quad R = [\bar{\mathbf{n}}, \bar{\mathbf{t}}], \tag{12}$$

$$\text{and} \quad \mathbf{c}_i = \mathbf{v}_i - L(A_i)R^t\Xi^{-1}R\sum_{j \in \mathcal{I}} \left(v_j^1\hat{\sigma}(\boldsymbol{\psi}_{j,T}^{non})(F_0)\bar{\mathbf{n}} + v_j^2\hat{\sigma}(\boldsymbol{\psi}_{j+4,T}^{non})(F_0)\bar{\mathbf{n}}\right), \quad i \in \mathcal{I}^-, \tag{13}$$

$$\text{with} \quad \hat{\sigma}(\mathbf{p}) = (\hat{\sigma}_{ij}(\mathbf{p}))_{1 \leqslant i,j \leqslant 2}, \quad \hat{\sigma}_{ij}(\mathbf{p}) = (\lambda^+ - \lambda^-)(\nabla \cdot \mathbf{p})\delta_{i,j} + 2(\mu^+ - \mu^-)\epsilon_{ij}(\mathbf{p}), \quad \forall \mathbf{p} \in [H^1(T)]^2, \tag{14}$$

$$\text{and} \quad \Xi = \begin{bmatrix} (\lambda^- + 2\mu^-) & 0 \\ 0 & \mu^- \end{bmatrix} + \begin{bmatrix} ((\lambda^+ - \lambda^-) + 2(\mu^+ - \mu^-))g_n(F_0) & (\lambda^+ - \lambda^-)g_t(F_0) \\ (\mu^+ - \mu^-)g_t(F_0) & (\mu^+ - \mu^-)g_n(F_0) \end{bmatrix}, \tag{15}$$

$$\text{wtih} \quad g_n(F_0) = \sum_{i \in \mathcal{I}^-} L(A_i)\nabla\psi_{i,T}^{non}(F_0)\bar{\mathbf{n}}, \quad g_t(F_0) = \sum_{i \in \mathcal{I}^-} L(A_i)\nabla\psi_{i,T}^{non}(F_0)\bar{\mathbf{t}}, \tag{16}$$

where $\nabla\psi_{i,T}^{non}$ are understood as 1-by-2 vectors, and $F_0 = F_0(P, Q)$ is a specific point on the line l specified in [35] depending on P and Q to guarantee the matrix Ξ is invertible such that $\mathbf{c}_i$ in (13) is well defined. This

format is derived such that the two polynomial components in the IFE shape functions in (11) can satisfy the nodal value condition $\boldsymbol{\psi}_T^{int}(A_i) = \mathbf{v}_i$, $i \in \mathcal{I}$ and the following approximate jump conditions:

$$\begin{cases} \boldsymbol{\psi}_T^{int,-}|_l = \boldsymbol{\psi}_T^{int,+}|_l, \\ \sigma^+(\boldsymbol{\psi}_T^{int,-})(F_0)\ \bar{\mathbf{n}} = \sigma^-(\boldsymbol{\psi}_T^{int,+})(F_0)\ \bar{\mathbf{n}}. \end{cases} \tag{17}$$

Here, we present these detailed formulas in (12)-(16) in order to derive the formulas of shape derivatives of IFE functions later. Setting $\mathbf{v}_i$ to be $[1,0]^t$, $[0,1]^t$ or $[0,0]^t$ suitably in the formulation (11)-(13), we obtain the vector IFE shape functions $\boldsymbol{\psi}_{i,T}^{int}(X)$, $i = 1, \cdots, 8$ on the interface element T such that $\boldsymbol{\psi}_{i,T}^{int}(A_j) = [\delta_{i,j}, 0]^t$ and $\boldsymbol{\psi}_{i+4,T}^{int}(A_j) = [0, \delta_{i,j}]^t$, $i, j \in \mathcal{I}$. Then, the local IFE space on each interface element $T \in \mathcal{T}_h^i$ is defined as

$$\mathbf{S}_h(T) = \text{Span}\{\boldsymbol{\psi}_{i,T}^{int},\ \boldsymbol{\psi}_{i+4,T}^{int},\quad i \in \mathcal{I}\}. \tag{18}$$

From now on, for simplicity, we only use $\boldsymbol{\psi}_{i,T} = [\psi_{i,T}^1, \psi_{i,T}^2]^t$, $i = 1, \cdots, 8$, to denote the IFE shape functions on either an interface or non-interface element T. Then, using (9) and (18), we define the global vector bilinear IFE space:

$$\mathbf{S}_h(\Omega) = \left\{\mathbf{v} \in [L^2(\Omega)]^2\ :\ \mathbf{v}|_T \in \mathbf{S}_h(T),\ \forall T \in \mathcal{T}_h \text{ and } \mathbf{v} \text{ is continuous at each } X \in \mathcal{N}_h\right\}, \tag{19}$$

and the associated space $\mathbf{S}_h^0(\Omega) = \{\mathbf{v} \in \mathbf{S}_h(\Omega)\ :\ \mathbf{v}(X) = \mathbf{0},\ \forall X \in \mathcal{N}_h \cap \partial\Omega_D\}$. Let $\boldsymbol{\phi}_{2i-1}(X)$ and $\boldsymbol{\phi}_{2i}(X)$ be the global IFE basis functions associated with the node $X_i \in \mathcal{N}_h$ such that $\boldsymbol{\phi}_{2i-1}(X_j) = [\delta_{i,j}, 0]^t$, $\boldsymbol{\phi}_{2i}(X_j) = [0, \delta_{i,j}]^t$, $i, j = 1, \cdots, |\mathcal{N}_h|$. Then, we have

$$\mathbf{S}_h(\Omega) = \text{Span}\{\boldsymbol{\phi}_{2i-1}(X), \boldsymbol{\phi}_{2i}(X), i = 1, 2, \cdots, |\mathcal{N}_h|\}.$$

We now describe how to use the bilinear IFE space to discretize the elasticity interface forward problems described by (3)-(5). As usual, letting $\otimes$ be the Kronecker product, we adopt the operators $[\cdot]_e$, $\underline{[\cdot]}_e$ and $\{\cdot\}_e$ for every $\mathbf{v} \in \mathbf{S}_h(\Omega)$ on each interface edge:

$$[\mathbf{v}]_e = (\mathbf{v}|_{T^1} + \mathbf{v}|_{T^2}),\quad \underline{[\mathbf{v}]}_e = (\mathbf{v}^t|_{T^1} \otimes \mathbf{n}_e^1 + \mathbf{v}^t|_{T^2} \otimes \mathbf{n}_e^2),\quad \{\sigma(\mathbf{v})\}_e = (\sigma(\mathbf{v}|_{T^1}) + \sigma(\mathbf{v}|_{T^2}))/2,\quad \text{if } e \in \mathring{\mathcal{E}}_h^i, \tag{20}$$

$$\text{and}\quad [\mathbf{v}]_e = \mathbf{v}|_T,\quad \underline{[\mathbf{v}]}_e = \mathbf{v}^t|_T \otimes \mathbf{n}_e,\quad \{\sigma(\mathbf{v})\}_e = \sigma(\mathbf{v}|_T),\quad \text{if } e \in \mathcal{E}_h^i \cap \partial\Omega, \tag{21}$$

where T^1 and T^2 are the two elements sharing e, the normal vector $\mathbf{n}_e^1 = -\mathbf{n}_e^2$ points to T^2 from T^1 if $e \in \mathring{\mathcal{E}}_h^i$ in (20), and T is the element containing e, $\mathbf{n}_e$ is the outward normal vector to $\partial\Omega$ if $e \in \mathcal{E}_h^i \cap \partial\Omega$ in (21).

Then, with the bilinear IFE space, we employ the symmetric PPIFE (SPPIFE) method [36] to discretize the elasticity interface direct problems (3)-(5) as follows: find $\mathbf{u}_h^k \in \mathbf{S}_h(\Omega)$, $k = 1, \cdots, K$, such that

$$a_h(\mathbf{u}_h^k, \mathbf{v}_h) = L_f^k(\mathbf{v}_h),\quad \forall \mathbf{v}_h \in \mathbf{S}_h^0(\Omega),\quad \mathbf{u}_h^k(X) = \mathbf{g}_D^k(X),\quad \forall X \in \mathcal{N}_h \cap \partial\Omega_D^k, \tag{22}$$

$$\begin{aligned} \text{where}\quad a_h(\mathbf{u}_h^k, \mathbf{v}_h) &= \sum_{T \in \mathcal{T}_h} \int_T 2\mu\epsilon(\mathbf{u}_h^k) : \epsilon(\mathbf{v}_h) dX + \sum_{T \in \mathcal{T}_h} \int_T \lambda(\nabla \cdot \mathbf{u}_h^k)(\nabla \cdot \mathbf{v}_h) dX \\ &- \sum_{e \in \mathcal{E}_h^i \setminus \partial\Omega_N^k} \int_e \{\sigma(\mathbf{u}_h^k)\}_e : \underline{[\mathbf{v}_h]}_e ds - \sum_{e \in \mathcal{E}_h^i \setminus \partial\Omega_N^k} \int_e \{\sigma(\mathbf{v}_h)\}_e : \underline{[\mathbf{u}_h^k]}_e ds + \sum_{e \in \mathcal{E}_h^i \setminus \partial\Omega_N^k} \frac{\rho}{|e|} \int_e [\mathbf{u}_h^k]_e \cdot [\mathbf{v}_h]_e ds, \end{aligned} \tag{23}$$

$$\begin{aligned} \text{and}\quad L_f^k(\mathbf{v}_h) &= \sum_{T \in \mathcal{T}_h} \int_T \mathbf{f}^k \cdot \mathbf{v}_h dX + \sum_{e \in \partial\Omega_N^k} \int_e \mathbf{g}_N^k \cdot \mathbf{v}_h ds \\ &- \sum_{e \in \mathcal{E}^i \cap \partial\Omega_D^k} \int_e (\sigma(\mathbf{v}_h)\mathbf{n}_e) \cdot \mathbf{g}_D^k ds + \sum_{e \in \mathcal{E}^i \cap \partial\Omega_D^k} \frac{\rho}{|e|} \int_e \mathbf{v}_h \cdot \mathbf{g}_D^k ds, \end{aligned} \tag{24}$$

and we choose $\rho = 10 \max\{\lambda^\pm, \mu^\pm\}$. We note that in (23)-(24), the penalties are only added on interface edges to alleviate the adverse impacts possibly caused by the discontinuity of IFE functions across the interface edges.

Also the involved IFE space has the same degrees of freedom as the standard continuous finite element space defined on the same mesh. This feature makes it advantageous in moving interface problems because both the location and total number of degrees of freedom do not change. Therefore, the PPIFE method to be used in the proposed IFE-based shape optimization is essentially different from the interior penalty DG methods in [10, 28]. We refer readers to [36, 56] for the detailed discussions of this IFE scheme.

We now put this IFE method in a matrix form in order to describe how it can be used in the shape optimization for the inverse geometric problems. By the standard assembling procedure, we can generate the matrix $\tilde{\mathbf{A}} = (a_{i,j})_{i,j=1}^{2|\mathcal{N}_h|}$ associated with the bilinear form defined in (23) by the following local matrices associated with elements and edges of $\mathcal{T}_h$:

$$\mathbf{K}_T^1 = \left(\int_T 2\mu\epsilon(\boldsymbol{\psi}_{p,T}) : \epsilon(\boldsymbol{\psi}_{q,T})dX\right)_{p,q}, \quad \mathbf{K}_T^2 = \left(\int_T \lambda(\nabla\cdot\boldsymbol{\psi}_{p,T})(\nabla\cdot\boldsymbol{\psi}_{q,T})dX\right)_{p,q}, \quad \forall T \in \mathcal{T}_h, \tag{25a}$$

$$\mathbf{E}_e^{r_1r_2} = \left(\int_e \sigma(\boldsymbol{\psi}_{p,T^{r_1}}) : (\boldsymbol{\psi}_{q,T^{r_2}} \otimes \mathbf{n}_e^{r_2})ds\right)_{p,q}, \quad \mathbf{G}_e^{r_1r_2} = \left(\frac{\rho}{|e|}\int_e \boldsymbol{\psi}_{p,T^{r_1}} \cdot \boldsymbol{\psi}_{q,T^{r_2}} ds\right)_{p,q}, \quad \forall e \in \mathcal{E}_h^i, \tag{25b}$$

where for $e \in \mathring{\mathcal{E}}_h^i$, $r_1, r_2 = 1, 2$, and for $e \in \mathcal{E}_h^i \cap \partial\Omega$, $r_1 = r_2 = 0$, then $\mathbf{n}_e^0 = \mathbf{n}_e$ and $T^0 = T$, which all have the same meaning in the notations (20) and (21). Here and from now on, we use $p, q = 1, \cdots, 8$ for the entries in these matrices in (25). Similarly, the load vector $\tilde{\mathbf{F}}^k = (f_i^k)_{i=1}^{2|\mathcal{N}_h|}$ corresponding to the linear form defined in (24) is assembled from the following local vectors:

$$\mathbf{F}_T^k = \left(\int_T \mathbf{f}^k \cdot \boldsymbol{\psi}_{p,T} dX\right)_p, \qquad \forall T \in \mathcal{T}_h, \tag{26a}$$

$$\mathbf{B}_e^k = \left(\int_e (\sigma(\boldsymbol{\psi}_{p,T})\mathbf{n}_e)\cdot \mathbf{g}_D^k ds\right)_p, \quad \mathbf{C}_e^k = \frac{\rho}{|e|}\left(\int_e \mathbf{g}_D^k \cdot \boldsymbol{\psi}_{p,T} ds\right)_p, \qquad \forall e \in \mathcal{E}_h^i \cap \partial\Omega_D, \tag{26b}$$

$$\mathbf{N}_e^k = \left(\int_e \mathbf{g}_N^k \cdot \boldsymbol{\psi}_{p,T} ds\right)_p, \qquad \forall e \in \mathcal{E}_h^i \cap \partial\Omega_N. \tag{26c}$$

where, again, we use $p = 1, \cdots, 8$ for the entries in these vectors.

When the k-th ($1 \le k \le K$) interface direct problem has the boundary condition such that $\left|\partial\Omega_D^k\right| \neq 0$, we let $\mathcal{N}_h^m = \{X_i \in \mathcal{N}_h \mid X_i \in \mathring{\mathcal{N}}_h \cup \partial\Omega_N\}$ and assume that the nodes in $\mathcal{N}_h^m$ are ordered first. Then the IFE solution $\mathbf{u}_h^k$ from the SPPIFE scheme (22)-(24) can be expressed as

$$\mathbf{u}_h^k(X) = \sum_{i=1}^{|\mathcal{N}_h^m|}\left(u_{2i-1}^k\boldsymbol{\phi}_{2i-1}(X) + u_{2i}^k\boldsymbol{\phi}_{2i}(X)\right) + \sum_{i=|\mathcal{N}_h^m|+1}^{|\mathcal{N}_h|}\left(g_{D,1}^k(X_i)\boldsymbol{\phi}_{2i-1}(X) + g_{D,2}^k(X_i)\boldsymbol{\phi}_{2i}(X)\right), \tag{27}$$

where $\mathbf{g}_D^k = [g_{D,1}^k\ g_{D,2}^k]^t$ is the Dirichlet boundary condition in (3b). In such a case, the Dirichlet boundary condition is implemented through the vector $\tilde{\mathbf{A}}_b^{m,k} = (a_{b,i}^k)_{i=1}^{2|\mathcal{N}_h|} = \tilde{\mathbf{A}}\begin{bmatrix}\mathbf{0} & \mathbf{b}_D^k\end{bmatrix}^t$, where $\mathbf{0}$ is the $1 \times 2\left|\mathcal{N}_h^m\right|$ zero vector and $\mathbf{b}_D^k = [g_{D,1}^k(X_{|\mathcal{N}_h^m|+1}), g_{D,2}^k(X_{|\mathcal{N}_h^m|+1}), \cdots, g_{D,1}^k(X_{|\mathcal{N}_h|}), g_{D,2}^k(X_{|\mathcal{N}_h|})] \in \mathbb{R}^{1\times 2(|\mathcal{N}_h|-|\mathcal{N}_h^m|)}$. With these preparations, we can put the SPPIFE method described by (22)-(24) in the following matrix form:

$$\mathbf{A}^{m,k}\bar{\mathbf{u}}^{m,k} = \mathbf{F}^{m,k}, \quad \text{with } \mathbf{A}^{m,k} = (a_{i,j})_{i,j=1}^{2|\mathcal{N}_h^m|}, \ \mathbf{F}^{m,k} = (f_i^k)_{i=1}^{2|\mathcal{N}_h^m|} - (a_{b,i}^k)_{i=1}^{2|\mathcal{N}_h^m|}, \tag{28}$$

where $\bar{\mathbf{u}}^{m,k} = [u_1^k, u_2^k, \cdots, u_{2|\mathcal{N}_h^m|}^k]^t$.

When the k-th interface direct problem has the pure Neumann boundary condition, i.e., $\partial\Omega_N^k = \partial\Omega$, vectors assembled from (26b) are not needed and (27) does not have the second summation. For uniquely determining the solution, as usual, we need the extra conditions: $\int_\Omega u_i^k dX = \omega_i$, $i = 1, 2$ and $\int_\Omega \mathrm{rot}(\mathbf{u}^k)dX = \omega_3$ imposed through three Lagrange multipliers $\lambda_1, \lambda_2, \lambda_3$. Let $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^t$, then the matrix form for the SPPIFE method in this case is expressed as

$$\mathbf{A}^n\bar{\mathbf{u}}^{n,k} = \mathbf{F}^{n,k}, \quad \text{with } \mathbf{A}^n = \begin{bmatrix}\tilde{\mathbf{A}} & \mathbf{R}\\ \mathbf{R}^t & \mathbf{0}\end{bmatrix}, \ \mathbf{F}^{n,k} = \begin{bmatrix}\tilde{\mathbf{F}}^k\\ \boldsymbol{\omega}\end{bmatrix}, \tag{29}$$

where $\bar{\mathbf{u}}^{n,k} = [u_1^k, u_2^k, \cdots, u_{2|\mathcal{N}_h|}^k, \lambda_1, \lambda_2, \lambda_3]^t$ and $\mathbf{R} = [\mathbf{R}^1, \mathbf{R}^2, \mathbf{R}^3] \in \mathbb{R}^{|\mathcal{N}_h|\times 3}$ is assembled from the following local vectors

$$\mathbf{R}_T^1 = \left(\int_T \psi_{p,T}^1 dX\right)_p, \quad \mathbf{R}_T^2 = \left(\int_T \psi_{p,T}^2 dX\right)_p, \quad \mathbf{R}_T^3 = \left(\int_T \mathrm{rot}(\boldsymbol{\psi}_{p,T}) dX\right)_p, \quad \forall T \in \mathcal{T}_h. \tag{30}$$

Based on (28) and (29), we summarize the SPPIFE discretization (22)-(24) with the mixed or pure Neumann boundary conditions into a unified matrix form for all K elasticity interface direct problems as follows:

$$\mathbf{A}^k \bar{\mathbf{u}}^k = \mathbf{F}^k, \quad k = 1, 2, \cdots, K, \tag{31}$$

$$\text{with } \bar{\mathbf{u}}^k = \begin{cases} \bar{\mathbf{u}}^{m,k}, \\ \bar{\mathbf{u}}^{n,k} \end{cases} \quad \mathbf{A}^k = \begin{cases} \mathbf{A}^{m,k}, \\ \mathbf{A}^{n,k}, \end{cases} \quad \mathbf{F}^k = \begin{cases} \mathbf{F}^{m,k}, & \text{for the boundary condition } |\partial\Omega_D^k| \neq 0, \\ \mathbf{F}^{n,k}, & \text{for a pure Neumann boundary condition.} \end{cases}$$

We emphasize that, on a fixed mesh, the size and algebraic structure of the symmetric positive definite matrices $\mathbf{A}^k$s in (31) remain the same as the interface $\Gamma(t, \boldsymbol{\alpha}), t \in [0, 1]$ evolves due to variation in the design variable $\boldsymbol{\alpha}$.

To facilitate simple discussions, we further rewrite (27) in a matrix-vector form:

$$\mathbf{u}_h^k(X) = \boldsymbol{\Phi}_m(X)\bar{\mathbf{u}}^k + \boldsymbol{\Phi}_D(X)\bar{\mathbf{g}}_D, \tag{32}$$

where $\bar{\mathbf{u}}^k$ is given in (31), and

$$\bar{\mathbf{g}}_D = [g_{D,1}^k(X_{|\mathcal{N}_h^m|+1}), g_{D,2}^k(X_{|\mathcal{N}_h^m|+1}), \cdots, g_{D,1}^k(X_{|\mathcal{N}_h|}), g_{D,2}^k(X_{|\mathcal{N}_h|})]^t, \tag{33}$$

$\boldsymbol{\Phi}_m(X)$ and $\boldsymbol{\Phi}_D(X)$ are 2-by-$2|\mathcal{N}_h^m|$ and 2-by-$2(|\mathcal{N}_h| - |\mathcal{N}_h^m|)$ matrix functions:

$$\boldsymbol{\Phi}_m = \left[\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_{2|\mathcal{N}_h^m|}\right], \quad \boldsymbol{\Phi}_D = \left[\boldsymbol{\phi}_{2|\mathcal{N}_h^m|+1}, \cdots, \boldsymbol{\phi}_{2|\mathcal{N}_h|}\right], \quad \text{and denote } \boldsymbol{\Phi} = [\boldsymbol{\Phi}_m, \boldsymbol{\Phi}_D]. \tag{34}$$

Again, when the direct problem has a pure Neumann boundary condition, (32) does not have the second term.

3 IFE Shape Optimization

In this section, we use the IFE functions and the PPIFE method (31) discussed in the previous section to discretize the shape optimization described by (6) and (7) subject to the governing interface problems (3)-(5) for solving the inverse geometric problem. We will also derive the formulas for the velocity field and the shape derivatives of IFE shape functions which are two key ingredients for calculating the shape sensitivities for the discretized shape optimization.

3.1 Discretization of Shape Optimization Problem

In shape optimization, in terms of the continuous mechanism, the movement of the interface deforms the shape of each subdomain of Ω accordingly. Therefore, each point $X \in \Omega$ can be tacitly considered to depend on the interface curve Γ, i.e., they are functions of Γ, see [21, 69]. Hence, we denote $X = X(\Gamma) = X(\Gamma(\boldsymbol{\alpha})) = X(\boldsymbol{\alpha})$ to emphasize the dependence of X on the design variable $\boldsymbol{\alpha}$.

In addition, the IFE shape functions intrinsically depend on $\boldsymbol{\alpha}$ not only through $X(\boldsymbol{\alpha})$ but also through those coefficients $c_0 = c_0(\boldsymbol{\alpha})$, $\mathbf{c}_i = \mathbf{c}_i(\boldsymbol{\alpha})$ and the function $L(X) = L(X; \boldsymbol{\alpha})$ in (12)-(13) of the IFE shape functions because of their dependence on points P and Q where the interface $\Gamma(\boldsymbol{\alpha})$ intersects with the edges of the interface element T. These dependencies on $\boldsymbol{\alpha}$ motivate us to rewrite a global IFE function as $\boldsymbol{\phi}_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ in which $X(\boldsymbol{\alpha})$ indicates the dependence of $\boldsymbol{\phi}_i$ on $\boldsymbol{\alpha}$ through X, and the second $\boldsymbol{\alpha}$ emphasizes the dependence of $\boldsymbol{\phi}_i$ on $\boldsymbol{\alpha}$ through those coefficients in IFE shape functions. Furthermore, we note that the matrices and vectors

in (25) and (26) are all defined by the integration of the IFE functions $\boldsymbol{\phi}_i(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})$; therefore, by similar considerations, we denote $\mathbf{A}^k = \mathbf{A}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})$ and $\mathbf{F}^k = \mathbf{F}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})$. Consequently, the vector solutions to the IFE equations (31) also depend on $\boldsymbol{\alpha}$, and we denote them by $\bar{\mathbf{u}}^k = \bar{\mathbf{u}}^k(\boldsymbol{\alpha}), 1 \le k \le K$ so that we express the IFE solutions (32) as

$$\mathbf{u}_h^k(\boldsymbol{\alpha}) = \mathbf{u}_h^k(\bar{\mathbf{u}}^k(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}),\boldsymbol{\alpha}) = \boldsymbol{\Phi}_m(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})\bar{\mathbf{u}}^k(\boldsymbol{\alpha}) + \boldsymbol{\Phi}_D(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})\bar{\mathbf{g}}_D. \tag{35}$$

Then, using the IFE solutions to approximate the displacement functions leads to the following disretized integrand for the objective functional in (7):

$$J_h(\bar{\mathbf{u}}^1(\boldsymbol{\alpha}), \bar{\mathbf{u}}^2(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}),\boldsymbol{\alpha}) \coloneqq J(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), \cdots, \mathbf{u}_h^K(\boldsymbol{\alpha}); X(\boldsymbol{\alpha}), \Gamma(\cdot,\boldsymbol{\alpha})). \tag{36}$$

Then, the discretized integrand defined in (36) can be used to define a discretized objective function:

$$\mathcal{J}_h(\bar{\mathbf{u}}_h^1(\boldsymbol{\alpha}), \bar{\mathbf{u}}_h^2(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}_h^K(\boldsymbol{\alpha}),\boldsymbol{\alpha}) = \int_{\Omega_d} J_h(\bar{\mathbf{u}}_h^1(\boldsymbol{\alpha}), \bar{\mathbf{u}}_h^2(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}_h^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}),\boldsymbol{\alpha}) dX. \tag{37}$$

We recall that the IFE solutions $\mathbf{u}_h^k$, $k = 1, \cdots, K$, can approximate the exact solutions $\mathbf{u}^k$ of the direct interface problems (3)-(5) with an optimal convergence rate regardless of the interface location on a fixed mesh [36]. This means that the IFE equations (31) are optimal disretizations of the elasticity systems (3)-(5). Furthermore, for many applications such as those demonstrated in Section 4, we can show that $\mathcal{J}_h(\bar{\mathbf{u}}^1(\boldsymbol{\alpha}), \bar{\mathbf{u}}^2(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}),\boldsymbol{\alpha})$ is also an optimal approximation to the continuous objective functional (7) in the shape optimization, see the derivation in (57) for an example. Hence, the shape optimization procedure described by (6) and (7) and the optimal discretizations for its components motivate us to propose an IFE-based shape optimization method on a fixed mesh of Ω for the inverse geometric problem: find the design parameter $\boldsymbol{\alpha}^*$ by carrying out the constrained optimization:

$$\begin{aligned} \boldsymbol{\alpha}^* = \operatorname{argmin} \mathcal{J}_h(\boldsymbol{\alpha}), \quad & \mathcal{J}_h(\boldsymbol{\alpha}) \coloneqq \mathcal{J}_h(\bar{\mathbf{u}}_h^1(\boldsymbol{\alpha}), \bar{\mathbf{u}}_h^2(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}_h^K(\boldsymbol{\alpha}),\boldsymbol{\alpha}), \\ \text{subject to} \quad & \mathbf{A}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})\bar{\mathbf{u}}^k(\boldsymbol{\alpha}) - \mathbf{F}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha}) = \mathbf{0}, \quad k = 1, 2, \cdots, K. \end{aligned} \tag{38}$$

This optimal design parameter $\boldsymbol{\alpha}^*$ leads to a parametric curve $\Gamma(\boldsymbol{\alpha}^*)$ that is expected to be a good approximation to Γ^*, the solution to the inverse geometric problem produced by the shape optimization (6) and (7).

It is well known that the inverse problems concerned in this article are ill-conditioned [8, 7, 47]. By choosing a reasonably small number of control points, our algorithm actually searches the target interface curve in a small space on fixed meshes. Therefore, the parameterization $\Gamma(\boldsymbol{\alpha})$ in the optimization problem (38) can be interpreted as a "hidden" regularization for the inverse problem. Indeed, the numerical experiments presented in Section 4 show that the reconstructions produced by the proposed IFE-based shape optimization method are quite satisfactory without additional regularization such as the geometric regularization [46], total variation regularization [23] and Tikhonov regularization [50]. We will discuss this feature in more details in Subsection 5.

In the following subsections, we will derive the velocity fields and the shape derivatives of IFE shape functions, and we will then proceed to discuss how these two key ingredients can be used to efficiently and accurately implement the formula for the gradient of $\mathcal{J}_h(\boldsymbol{\alpha})$ within the IFE framework so that common numerical optimization algorithms such as descent direction methods and trust region methods [27, 62] can be employed to solve this constrained optimization problem.

3.2 A Velocity Field on A Cartesian Mesh

As described above, when the interface $\Gamma(\boldsymbol{\alpha})$ evolves in the optimization, each subdomain changes its shape and each point also moves accordingly, and this feature is reflected by the notation $X(\boldsymbol{\alpha})$, see the discussions at the beginning of Subsection 3.1. In this subsection, we follow the idea in [37, 48, 53, 60] to develop the derivative of $X(\boldsymbol{\alpha})$ with respect to the interface variation, i.e., we construct the velocity fields $\frac{\partial X(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}$, on a Cartesian mesh,

which is a critical ingredient [40] for computing the shape sensitivities of the discrete cost function $\mathcal{J}_h(\boldsymbol{\alpha})$ in the proposed IFE-based method.

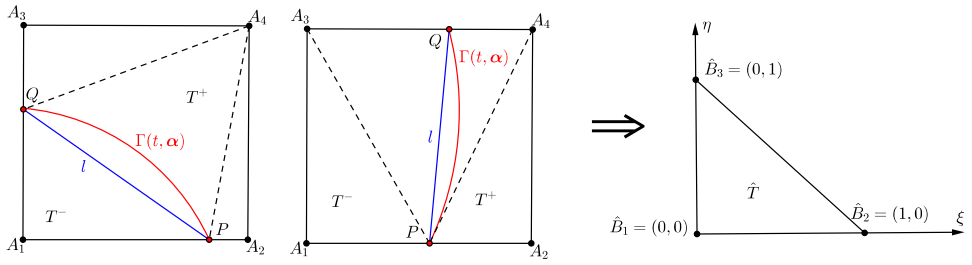


Figure 2: Interface elements and partitioning

On every non-interface element, since the spatial point X can be considered to be a constant function of $\boldsymbol{\alpha}$, we obviously have $\frac{\partial X}{\partial \boldsymbol{\alpha}} = \mathbf{0}$. For the velocity on interface elements, let us consider a typical interface element $T = \square A_1A_2A_3A_4$ with the interface-mesh intersection points of $\Gamma(t,\boldsymbol{\alpha})$ and ∂T denoted by $P = P(\boldsymbol{\alpha})$ and $Q = Q(\boldsymbol{\alpha})$, see Figure 2 for an illustration. For simplicity's sake, we assume P, Q do not coincide with the vertices of T, all the results here are readily generalized to the case that the interface passes through the mesh nodes. Note that the derivative of $P = P(\boldsymbol{\alpha})$ and $Q = Q(\boldsymbol{\alpha})$ with respect to the design variables $\boldsymbol{\alpha}$ is already described by Lemma 3.1 in [37] which is actually the velocity at these two points. This derivative only depends on the chosen parameterization for the interface curve for which we also refer readers to [37] for a detailed discussion. Since the vertices of T are fixed, the basic idea for constructing a suitable velocity field is to establish a relation between the interior points of T and the intersection points P and Q, and use the movement of P, Q to guide to the movement of other interior points. For this purpose, with out loss of generality, we only discuss the **Case 1** interface element illustrated by the 1st sketch in Figure 2, the **Case 2** interface element illustrated by the 2nd sketch in Figure 2 can be handled similarly. We partition T into 4 subelements: $T_1 = \triangle A_1PQ$, $T_2 = \triangle A_2A_4P$, $T_3 = \triangle A_3QA_4$, $T_4 = A_4QP$, illustrated by the 1st sketch in Figure 2, and let $\hat{T} = \triangle \hat{B}_1\hat{B}_2\hat{B}_3$ be the usual reference triangle with vertices $\hat{B}_1 = [0,0]^t$, $\hat{B}_2 = [1,0]^t$, $\hat{B}_1 = [0,1]^t$, shown by the 3rd sketch in Figure 2. Then, following the idea used in [40, 48, 60], we can use the standard affine mappings from the reference element $\hat{T}$ to $T_m, m = 1,2,3,4$ to describe the dependence of the spatial point X on the design variable $\boldsymbol{\alpha}$ as follows:

$$X(\boldsymbol{\alpha}) = \mathcal{F}_m(\boldsymbol{\alpha},\xi,\eta) = \mathbf{J}_m(\boldsymbol{\alpha}) \begin{bmatrix} \xi \\ \eta \end{bmatrix} + A_m, \quad \text{for} \quad \begin{bmatrix} \xi \\ \eta \end{bmatrix} \in \hat{T}, \quad m = 1,2,3,4, \tag{39}$$

in which the Jacobian matrices $\mathbf{J}_m(\boldsymbol{\alpha})$ are given by

$$\begin{aligned} \mathbf{J}_1(\boldsymbol{\alpha}) &= [P(\boldsymbol{\alpha}) - A_1, Q(\boldsymbol{\alpha}) - A_1], \quad \mathbf{J}_2(\boldsymbol{\alpha}) = [A_4 - A_2, P(\boldsymbol{\alpha}) - A_2], \\ \mathbf{J}_3(\boldsymbol{\alpha}) &= [Q(\boldsymbol{\alpha}) - A_3, A_4 - A_3], \quad \mathbf{J}_4(\boldsymbol{\alpha}) = [Q(\boldsymbol{\alpha}) - A_4, P(\boldsymbol{\alpha}) - A_4]. \end{aligned}$$

Note that $X(\boldsymbol{\alpha})$ in (39) is a piecewise differentiable function on each subelement $T_m \subset T$, $m = 1,2,3,4$. Also, the formulas for calculating $D_{\alpha_j}P(\boldsymbol{\alpha})$, $D_{\alpha_j}Q(\boldsymbol{\alpha})$, $j \in \mathcal{D}$, i.e., the velocity at these two points, have been established in Lemma 3.1 in [37]. Therefore, the derivatives of X with respect to α_j, $j \in \mathcal{D}$, can be calculated by

$$D_{\alpha_j}X(\boldsymbol{\alpha}) = (D_{\alpha_j}\mathbf{J}_m(\boldsymbol{\alpha}))\mathbf{J}_m^{-1}(\boldsymbol{\alpha})(X(\boldsymbol{\alpha}) - A_m) \quad \text{for } X(\boldsymbol{\alpha}) \in T_m \subseteq T,\ m = 1,2,3,4, \tag{40}$$

$$\begin{aligned} \text{where} \quad & D_{\alpha_j}\mathbf{J}_1(\boldsymbol{\alpha}) = \left[D_{\alpha_j}P, D_{\alpha_j}Q\right], \quad D_{\alpha_j}\mathbf{J}_2(\boldsymbol{\alpha}) = \left[\mathbf{0}, D_{\alpha_j}P\right], \\ & D_{\alpha_j}\mathbf{J}_3(\boldsymbol{\alpha}) = \left[D_{\alpha_j}Q, \mathbf{0}\right], \quad D_{\alpha_j}\mathbf{J}_4(\boldsymbol{\alpha}) = \left[D_{\alpha_j}Q, D_{\alpha_j}P\right]. \end{aligned} \tag{41}$$

Thus, combining the zero velocity field on non-interface elements and the formulas in (40) on interface elements, we piecewisely define the velocity field $\mathbf{V}^j$ corresponding to the j-th design variable α_j, $j \in \mathcal{D}$ as:

$$\mathbf{V}^j(X) = \begin{cases} \mathbf{V}_T^j(X) = \mathbf{0}, & \text{for } X \in T \in \mathcal{T}_h^n \ , \\ \mathbf{V}_T^j(X) = (D_{\alpha_j}\mathbf{J}_m(\boldsymbol{\alpha}))\mathbf{J}_m^{-1}(\boldsymbol{\alpha})(X(\boldsymbol{\alpha}) - A_m), & \text{for } X \in T \in \mathcal{T}_h^i \text{ and } X \in T_m, m = 1,2,3,4. \end{cases} \tag{42}$$

We note that the velocity in (42) in each sub-triangle can be considered as a linear interpolation of those velocities on intersection points and element vertices. Furthermore, we note that the proposed velocity fields (42) are close to the one constructed in [37], the difference is on the element shape. Also the velocity fields (42) share properties similar to those given in Theorem 3.1 in [37] which is copied below for reader's convenience:

Theorem 3.1. *The velocity fields $\mathbf{V}^j(X)$, $j \in \mathcal{D}$, defined in (42) are such that:*

(P1) *on each interface element $T = \triangle A_1A_2A_3A_4 \in \mathcal{T}_h^i$, there hold*

$$\mathbf{V}_T^j|_{A_iP} = \frac{\|X - A_i\|}{\|P - A_i\|} D_{\alpha_j}P, \ i = 1,2,4 \qquad \mathbf{V}_T^j|_{A_iQ} = \frac{\|X - A_i\|}{\|Q - A_i\|} D_{\alpha_j}Q, \ i = 1,3,4 \tag{43a}$$

$$\mathbf{V}_T^j|_{PQ} = \frac{\|X - Q\|}{\|P - Q\|} D_{\alpha_j}P + \frac{\|X - P\|}{\|P - Q\|} D_{\alpha_j}Q, \tag{43b}$$

$$\mathbf{V}_T^j|_{A_3A_4} = \mathbf{0}, \qquad \mathbf{V}_T^j|_{A_2A_4} = \mathbf{0}, \tag{43c}$$

$$\operatorname{div}(\mathbf{V}_{T_m}^j) = \operatorname{tr}\left((D_{\alpha_j}\mathbf{J}_m)\mathbf{J}_m^{-1}\right), \quad m = 1,2,3,4; \tag{43d}$$

(P2) *$\mathbf{V}^j \in H^1(\Omega)$ and $supp(\mathbf{V}^j) \subseteq \bigcup_{T \in \mathcal{T}_h^i} T$;*

(P3) *$\mathbf{V}^j|_e$ has the same direction as the edge e, if $e \in \mathcal{E}_h^i$.*

3.3 Shape Derivatives of IFE Shape Functions

In shape optimization, it is inevitable to consider the rate of change of a quantity with respect to the design variable $\boldsymbol{\alpha}$ not directly through the spatial variable $X(\boldsymbol{\alpha})$ itself considered as a function of the design variable, and this rate of change is referred as the shape derivative in the literature [40]. The shape derivatives of the scalar IFE shape functions have been presented in [37] for solving shape optimization problems governed by the scalar elliptic equations. In this subsection, we derive the formulas for the shape derivatives of the vector IFE shape functions which are needed for computing the gradient of the cost function in the IFE shape optimization described by (38).

First of all, on each non-interface element, every IFE shape function is actually the usual finite element shape function that is just a polynomial independent of the design variable; hence, the shape derivative of an IFE shape function should be zero on all the non-interface elements. On every interface element T, we write an IFE shape functions as as $\boldsymbol{\psi}_{i,T}^{int}(X) = \boldsymbol{\psi}_{i,T}^{int}(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}), 1 \le i \le 8$ where again the first variable $X(\boldsymbol{\alpha})$ reflects the influence of the design variable $\boldsymbol{\alpha}$ on the IFE shape function $\boldsymbol{\psi}_{i,T}^{int}$ according to the formula $X = X(\boldsymbol{\alpha})$ given in (39), and the second variable $\boldsymbol{\alpha}$ is for the influence of $\boldsymbol{\alpha}$ on the IFE shape function $\boldsymbol{\psi}_{i,T}^{int}$ through its coefficients according to (12) and (13). Hence the shape derivatives of $\boldsymbol{\psi}_{i,T}^{int}$ are denoted as the partial derivatives $\frac{\partial}{\partial \alpha_j}\boldsymbol{\psi}_{i,T}^{int}(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}), j \in \mathcal{D}$.

The formulas (11)-(13) for the IFE shape functions suggest that, to compute their shape derivatives, we only need to calculate the derivatives of $L(X)$, $\mathbf{c}_i$, $i \in \mathcal{I}^-$, and $\mathbf{c}_0$ with respect to the design variables $\boldsymbol{\alpha}$. By inspecting the formulas (12)-(16), we note that they are functions of intersection points P and Q; so the desired shape derivatives should only rely on the velocity at these two intersection points given by Lemma 3.1 in [37]. More specifically, given a tensor $\mathbf{r} \in \mathbf{R}^{r \times 1}$ which could be the coefficients $\mathbf{c}$, c_0 or the function $L(X)$ in the IFE function (11), its shape derivatives can be decoupled by the standard chain rule:

$$\frac{\partial \mathbf{r}}{\partial \alpha_j} = \frac{\partial \mathbf{r}}{\partial P} \cdot D_{\alpha_j}P + \frac{\partial \mathbf{r}}{\partial Q} \cdot D_{\alpha_j}Q \tag{44}$$

where $\frac{\partial \mathbf{r}}{\partial P}$ and $\frac{\partial \mathbf{r}}{\partial Q}$ are understood as r-by-2 row vector. Since the derivatives $\frac{\partial P}{\partial \alpha_j}$ and $\frac{\partial Q}{\partial \alpha_j}$ are already known (Lemma 3.1 in [37]), we herein only need to derive the formulas of $\frac{\partial \mathbf{r}}{\partial P}$ and $\frac{\partial \mathbf{r}}{\partial Q}$ with $\mathbf{r} = \mathbf{c}, c_0$ and $L(X)$ for the IFE functions used in this article. Their calculations are simply based on standard multivariable calculus together with the chain rule. Here, in order for any interesting readers to directly implement the proposed shape optimization algorithm, we present the explicit formulas of these derivatives in Appendix A.

Finally, we put (65), (69), (66) in the Appendix A together with the generic formula (44) to have the following formula for the shape derivatives of the IFE shape functions on an interface element T:

$$\frac{\partial \boldsymbol{\psi}_T^{int}(X,\boldsymbol{\alpha})}{\partial \alpha_j} = \begin{cases} \dfrac{\partial \boldsymbol{\psi}_T^{int,-}(X,\boldsymbol{\alpha})}{\partial \alpha_j} = \dfrac{\partial \boldsymbol{\psi}_T^{int,+}(X,\boldsymbol{\alpha})}{\partial \alpha_j} + \dfrac{\partial L(X,\boldsymbol{\alpha})}{\partial \alpha_j}\mathbf{c}_0 + L(X,\boldsymbol{\alpha})\dfrac{\partial \mathbf{c}_0}{\partial \alpha_j}, & \text{if} \quad X \in T^-, \\ \dfrac{\partial \boldsymbol{\psi}_T^{int,+}(X,\boldsymbol{\alpha})}{\partial \alpha_j} = \displaystyle\sum_{i\in\mathcal{I}^-} \Psi_{i,T}^{non}(X)\frac{\partial \mathbf{c}_i}{\partial \alpha_j}, & \text{if} \quad X \in T^+, \end{cases} \tag{45}$$

for every $j \in \mathcal{D}$.

3.4 Shape Sensitivities for The IFE Shape Optimization

In this subsection, we apply the results established in the previous subsections to derive the formulas for calculating the shape sensitivity or the gradient of the discretized objective function $\mathcal{J}_h(\boldsymbol{\alpha})$ in (38) with respect to the design variable $\alpha_j, j \in \mathcal{D}$.

By Lemma 3.3 in [40], we have the following formula for the material derivatives of $\mathcal{J}_h(\boldsymbol{\alpha})$:

$$D_{\alpha_j}\mathcal{J}_h(\boldsymbol{\alpha}) = \sum_{k=1}^{K}\left(\frac{\partial \mathcal{J}_h}{\partial \bar{\mathbf{u}}^k}\cdot D_{\alpha_j}\bar{\mathbf{u}}^k\right) + \int_{\Omega_d}\frac{\partial J_h}{\partial \alpha_j}dX + \int_{\Omega_d}\nabla J_h\cdot\mathbf{V}^j dX + \int_{\Omega_d} J_h \text{ div}\left(\mathbf{V}^j\right)dX. \tag{46}$$

In this formula, we note that $\mathbf{V}^j$ is given in (42), $\text{div}\left(\mathbf{V}^j\right)$ is given in (43d) computed element-wisely, and $\frac{\partial \mathcal{J}_h}{\partial \bar{\mathbf{u}}^k} = \int_{\Omega_d}\frac{\partial J_h}{\partial \bar{\mathbf{u}}^k}dX$, ∇J_h, $\frac{\partial J_h}{\partial \alpha_j}$ and J_h are problem dependent but they are usually easy to compute as demonstrated by the representative examples provided in the next section. For the first term on the right of (46), by the standard discretized adjoint method [31], we have

$$\left(\frac{\partial \mathcal{J}_h}{\partial \bar{\mathbf{u}}^k}\right)\cdot D_{\alpha_j}\bar{\mathbf{u}}^k = \mathbf{Y}^k\cdot\left(D_{\alpha_j}\mathbf{F}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha}) - D_{\alpha_j}\mathbf{A}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})\ \bar{\mathbf{u}}^k(\boldsymbol{\alpha})\right), \quad k = 1,\cdots,K, \tag{47}$$

where $\mathbf{Y}^k$ is solved from

$$\left(\mathbf{A}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})\right)^T\mathbf{Y}^k = \frac{\partial \mathcal{J}_h}{\partial \bar{\mathbf{u}}^k}. \tag{48}$$

The matrix $D_{\alpha_j}\mathbf{A}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha}), j \in \mathcal{D}$ in formula (47) is the material derivative of the stiffness matrix $\mathbf{A}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})$, and it is assembled from the corresponding material derivative of the local matrices listed in (25) through the standard element-by-element procedure. Furthermore, the material derivatives of those local matrices can be straightforwardly derived in terms of the formulas of the velocity fields and the shape derivatives of the IFE shape functions presented in the previous subsections, see Appendix B for these formulas. Similarly, $D_{\alpha_j}\mathbf{F}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})$ is the material derivative of the load vector $\mathbf{F}^k(X(\boldsymbol{\alpha}),\boldsymbol{\alpha})$, and it is made by the corresponding material derivative of the local vectors listed in (26), formulas for these material derivatives of the local vectors are given in Appendix C.

3.5 Implementation of the IFE Shape Optimization Method

We now finish this section by summarizing the discussions and derivations above into the following IFE-based shape optimization algorithm for solving the inverse geometric problems for the elasticity interface problems.

Algorithm The IFE-based Shape Optimization Algorithm

1: Choose a mesh and choose an initial value for the design variable $\boldsymbol{\alpha}$.
2: Loop until convergence
i: Prepare data:
a: use the design variable $\boldsymbol{\alpha}$ to generate a parametric curve $\Gamma(t, \boldsymbol{\alpha})$ as the current numerical interface.
b: find the interface-mesh intersection points, interface edges, and the interface elements.
ii: Prepare matrices and vectors for the IFE systems and compute the cost function:
a: use IFE local matrices/vectors given in (25), (26), and (30) to assemble the global matrices and vectors $\mathbf{A}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ and $\mathbf{F}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ in the IFE systems (31).
b: compute the PPIFE solutions $\bar{\mathbf{u}}^k, 1 \leq k \leq K$ by (31), and use them to compute $\mathcal{J}_h(\boldsymbol{\alpha})$ in (38).
iii: Compute the shape sensitivities:
a: generate the velocity fields $\mathbf{V}^j$ (42), $j \in \mathcal{D}$, and shape derivatives of IFE shape functions (45);
b: assemble the global matrices $D_{\alpha_j}\mathbf{A}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ and vectors $D_{\alpha_j}\mathbf{F}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$;
c: use (47) to compute $\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k} \cdot D_{\alpha_j}\mathbf{u}_h^k$ for $k = 1, \cdots, K$;
d: compute the terms $\int_{\Omega_d} \frac{\partial J_h}{\partial \alpha_j} dX$, $\int_{\Omega_d} \nabla J_h \cdot \mathbf{V}^j dX$ and $\int_{\Omega_d} J_h \mathrm{div}(\mathbf{V}^j) dX$ according to the specific form of the given objective functional;
e: compute the material derivatives of the objective function $\mathcal{J}_h(\boldsymbol{\alpha})$ according to (46).
iv: Use $\mathcal{J}_h(\boldsymbol{\alpha})$ and its material derivatives to update the design variable $\boldsymbol{\alpha}$ by a chosen gradient-based optimization algorithm.
3: End loop

In step 2ia of the Algorithm, we follow the procedure described at the beginning of Section 2 to generate a cubic spline as a parametric interface $\Gamma(t, \boldsymbol{\alpha})$. We repeat this procedure at each iteration according to the updated design variable $\boldsymbol{\alpha}$. Therefore, the design variable $\boldsymbol{\alpha}$ consisting of coordinates of control points totally controls the evolutions of shape and location of the interface. Namely, we only need to update $\boldsymbol{\alpha}$ to move the interface curve.

In step 2ib, the mesh for finite element computation is generated a-priori and fixed during optimization. On this fixed mesh, we only need to update those interface-mesh intersection points, the related edges, and elements at each iteration. The interface edges and elements are determined by whether their nodes are inside or outside the interface. Once the interface edges are found, then the Cartesian coordinates and parametric coordinates of the interface-mesh intersection points are computed by applying the standard Newton's method to the equation formed by the parametric function (8) of the interface and the linear functions of each interface edge.

In step 2iia, the global stiffness matrices $\mathbf{A}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ and load vectors $\mathbf{F}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ inherit most of their entries from their counterparts in the previous iteration, we only need to reassemble those entries involving the interface elements in the current and previous iterations. This feature makes the assembling process for the global stiffness matrices and load vectors very efficient in a shape optimization, and this is one advantage of IFE method for moving interface problems.

For the computations of shape sensitivities in step iii of the Algorithm, the two critical ingredients, i.e., the velocity fields and shape derivatives of IFE shape functions, are computed accurately by formulas (42) and (45). In addition, their computations are very local in the sense that they only need to be computed on interface elements because they vanish on all non-interface elements. Consequently, the assemblage for $D_{\alpha_j}\mathbf{A}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ and $D_{\alpha_j}\mathbf{F}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ is performed only over interface elements and interface edges by using the explicit formulas for their local counterparts given in Appendix B and Appendix C. Furthermore, the terms $\int_{\Omega_d} \frac{\partial J_h}{\partial \alpha_j} dX$, $\int_{\Omega_d} \nabla J_h \cdot \mathbf{V}^j dX$ and $\int_{\Omega_d} J_h \mathrm{div}(\mathbf{V}^j) dX$ in (46) also only need to be computed on $\Omega_d \cap (\cup_{T \in \mathcal{T}_h^i} T)$ since the involved velocity fields are zero outside the interface elements. Therefore, compared to Lagrange approaches in the literature that generally require global computations over all the elements/edges for sensitivity analysis [21], the shape sensitivity computation in the IFE-based shape optimization algorithm can be executed very efficiently because the number of interface elements/edges is only in the order of $O(h^{-1})$ compared to the total

number of elements/edges in the order of $O(h^{-2})$.

We also note that the proposed IFE-based shape optimization algorithm is suitable for a highly parallel implementation from two perspectives. Firstly, computing the material derivatives with respect to each individual design variable is independent with each other; hence, these computations can be done very efficiently in parallel. Secondly, in the case that there are K available measurements, the elasticity systems (3)-(5) need to be solved K times in general, but they can be also solved efficiently in parallel.

In summary, these features together with the optimal accuracy of the IFE systems and the resulted optimal accuracy of discretized objective functions, regardless of the interface location in a chose mesh, strongly indicate the competitiveness of the proposed IFE-based shape optimization method for the inverse geometric problems of the planar elasticity interface problems.

4 Numerical Examples

In this section, we apply the IFE-based shape optimization algorithm developed in the previous sections to recover the material interface in three groups of representative inverse problems that are derived from applications such as ultrasound elastography in medical diagnosis [9, 29, 64] where the displacement fields caused by the applied stress on the boundary are measured to detect the changes of mechanic properties inside tissue. We choose these examples to demonstrate the versatility of this IFE-based method for dealing with a wide spectrum of inverse interface problems in terms of different types of data and objective functionals. Of course, the application of the proposed algorithm is not limited to the examples discussed here.

All the examples to be presented are posed on the domain $\Omega = (-1, 1) \times (-1, 1)$ on which an interface-independent rectangular Cartesian mesh will be used. As for the optimization algorithm, we use the standard BFGS optimization algorithm, a well known quasi-Newton method [62], to update the design variables $\boldsymbol{\alpha}$ in the shape optimization iterations. For the data to be used to recover the interface in the numerical examples, we consider the displacement measurements provided in a domain $\Omega_d = \Omega_\theta$ defined by:

$$\Omega_\theta = \{X \in \Omega \ : \ \text{dist}(X, \partial\Omega) \leqslant \theta\}, \quad \theta \in [0, 1] \tag{49}$$

which consists of points in the vicinity of $\partial\Omega$, in particular, $\Omega_1 = \Omega$ and $\Omega_0 = \partial\Omega$. For $0 < \theta \leqslant 1$, we show Ω_θ on the plots by shaded region, see Figures 3-10. Note that when $\theta = 0$, the shaded data region reduces to the boundary. Similar measurement domains were used in [46].

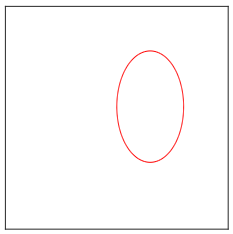
A suitable number of control points to generate the cubic spline used in the shape optimization is one of the keys for a successful reconstruction. Overall, we believe the choice of this number may be effected by two factors: the geometric complexity of the unknown interface and ill-posedness nature of inverse problems. On one hand, ideally, more control points enable us to numerically represent more complicated interface curves with higher resolution, and thus this number may depend on some a-priori information of the actual interface. On the other hand, the considered problem of reconstructing elastic inclusions is ill-posed and even the uniqueness remains an issue, i.e., whether the single or multiple boundary measurements can uniquely determine the interface is still open [2, 3]. We also refer readers to [44, 52] for the ill-posedness issue in related inverse problems for scalar elliptic equations which is more documented. Alternatively, instead of pursuing highly accurate reconstruction for arbitrary interface, following the general ideas described in [2, 3], we think it can be more stable and efficient to just recover a few geometric parameters of the unknown interface. Thus, we expect that a relatively small number of control points may actually help for a successful reconstruction which can be considered as certain self-regularization mechanism of the proposed parameterized shape optimization algorithm. We remark that this is also one of the advantages compared with interface-capturing methods in the literature.

In the presented numerical experiments, we all choose the number of control points to be 20 (40 design variables) which is not only enough to represent a large class of interface curves in many real applications but also yields a searching space with a moderate dimension for the purpose of efficient reconstruction. See the plots in Figures 3-6 for an illustration of these control points.

We consider three interface shapes defined by the zero level set function $S(x_1, x_2) = 0$ and the corresponding data functions $\mathbf{u}_\theta^k$ as follows. It can readily verified that these functions $\mathbf{u}_\theta^k, 1 \leq k \leq K$ are exact solutions to

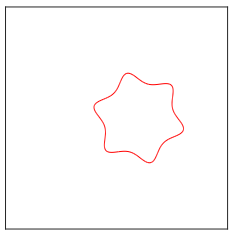
the interface problems of the linear system (3)-(5). However, these functions are employed only to generate the data $\mathbf{f}^k, \mathbf{g}_D^k, \mathbf{g}_N^k, k = 1, 2, \cdots, K$ for interface direct problems (3)-(5) and the data measurements on $\Omega_d = \Omega_\theta$ for inverse geometric problems. The proposed IFE-based shape optimization algorithm does not use these functions $\mathbf{u}_\theta^k, 1 \le k \le K$ outside data region $\Omega_d = \Omega_\theta$.

Case 1 (Ellipse) : $S(x_1, x_2) = (x_1 - 0.3)^2/0.3^2 + (y - 0.1)/0.5^2 - 1$, and



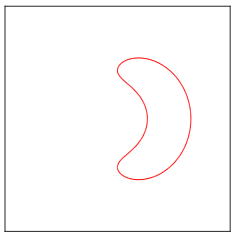
$$\mathbf{u}_\theta^1 = \begin{cases} \begin{bmatrix} \frac{0.0225}{\lambda^-}(S+1) \\ \frac{0.0225}{\lambda^-}(S+1) \end{bmatrix} & \text{if } X \in \Omega^-, \\ \begin{bmatrix} \frac{0.0225}{\lambda^+}(S+1) + 0.0225\left(\frac{1}{\lambda^-} - \frac{1}{\lambda^+}\right) \\ \frac{0.0225}{\lambda^+}(S+1) + 0.0225\left(\frac{1}{\lambda^-} - \frac{1}{\lambda^+}\right) \end{bmatrix} & \text{if } X \in \Omega^+; \end{cases} \tag{50}$$

Case 2 (Star): $S(x_1, x_2) = ((x_1 - 0.2)^2 + x_2^2)^2(1 + 0.5\sin(6\arctan(x_2/(x_1 - 0.2)))) - 0.015$, and



$$\mathbf{u}_\theta^k = \begin{cases} \begin{bmatrix} \frac{(2.1-x_1^2-x_2^2)^{\beta_1^k}}{\lambda^-}S \\ \frac{(2.1-x_1^2-x_2^2)^{\beta_2^k}}{\lambda^-}S \end{bmatrix} & \text{if } X \in \Omega^-, \\ \begin{bmatrix} \frac{(2.1-x_1^2-x_2^2)^{\beta_1^k}}{\lambda^+}S \\ \frac{(2.1-x_1^2-x_2^2)^{\beta_2^k}}{\lambda^+}S \end{bmatrix} & \text{if } X \in \Omega^+, \end{cases} \quad 1 \le k \le K. \tag{51}$$

Case 3 (Kidney): $S(x_1, x_2) = (2((x_1 + 0.2)^2 + x_2^2) - (x_1 + 0.2))^2 - (0.7(x_1 + 0.2)^2 + x_2^2) + 0.15$, and



$$\mathbf{u}_\theta^k = \begin{cases} \begin{bmatrix} \frac{(2.1-x_1^2-x_2^2)^{\beta_1^k}}{\lambda^-}S \\ \frac{(2.1-x_1^2-x_2^2)^{\beta_2^k}}{\lambda^-}S \end{bmatrix} & \text{if } X \in \Omega^-, \\ \begin{bmatrix} \frac{(2.1-x_1^2-x_2^2)^{\beta_1^k}}{\lambda^+}S \\ \frac{(2.1-x_1^2-x_2^2)^{\beta_2^k}}{\lambda^+}S \end{bmatrix} & \text{if } X \in \Omega^+, \end{cases} \quad 1 \le k \le K. \tag{52}$$

These interfaces to be recovered have distinct geometric features: The interface curve in **Case 1** is a simple ellipse with a convex shape, the one in **Case 2** has an alternating local convexity and concavity, and the one in **Case 3** has a large concave portion. The exponents β_1^k and β_2^k in the data functions (51), (52) in **Case 2** and **Case 3** are for generating different data, i.e., multiple stress data and the corresponding displacement measurements for each individual interface. To be specific, we employ the following values for β_1^k and β_2^k:

(β_1^1, β_2^1)	(β_1^2, β_2^2)	(β_1^3, β_2^3)	(β_1^4, β_2^4)	(β_1^5, β_2^5)	(β_1^6, β_2^6)	(β_1^7, β_2^7)
$(0, 0)$	$(1, 0)$	$(0, 1)$	$(2, 0)$	$(0, 2)$	$(3, 0)$	$(0, 3)$

(53)

$$(\beta_1^k, \beta_2^k) = \begin{cases} (3 + (k-6)/4, 0) & \text{if k is an even integer,} \\ (0, 3 + (k-7)/4) & \text{if k is an odd integer,} \end{cases} \quad k \geqslant 8. \tag{54}$$

These different data functions simulate the unrelated mechanisms applied on an elastic body in the elastography which cause uncorrelated echogenicity and stiffness, and then provide new information about the structure of the elastic body [64]. In our numerical experiments, it is observed that using multiple measurements indeed

enables the proposed IFE-based shape optimization algorithm to produce a better reconstruction, especially when the target interface has a challenging shape. On the other hand, we note that K measurements correspond to K different boundary value problems in (3), and thus, the shape optimization is more expensive when we use more measurements in the objective functional because we need to solve more direct problems in each iteration. When multiple measurements are available, one approach we employ here to reduce the computational cost is to start the optimization with only a single measurement and gradually use other available measurements as the shape optimization proceeds.

4.1 Internal Displacement Measurements

In this group of examples, we consider the inverse geometric problem in which there are $K \geqslant 1$ internal displacement measurements $\mathbf{u}_\theta^k$, $k = 1, \cdots, K$, available on Ω_θ defined in (49). The shape functional for the inverse geometric problem is in an output-least-squares form as follows:

$$\mathcal{J}(\mathbf{u}^1(\Gamma), \cdots, \mathbf{u}^K(\Gamma), \Gamma) = \sum_{k=1}^{K} \int_{\Omega_\theta} \|\mathbf{u}^k - \mathbf{u}_\theta^k\|^2 dX, \tag{55}$$

where $\mathbf{u}^k$ is the solution to the k-th elasticity interface problem described by (3)-(5) with the pure Dirichlet boundary condition given by the measurements $\mathbf{g}_D^k = \mathbf{u}_\theta^k|_{\partial\Omega}$. This shape functional also has been used in [46] for a similar inverse problem. It is easy to see that the simulated solutions $\mathbf{u}^k$ can match the actual measurements $\mathbf{u}_\theta^k$ when Γ is the actual interface, and this certainly satisfies the general principle for the choice of the cost functional discussed in Section 1. We also note that it employs all the available data in its formulation to enhance the reconstruction. In addition, we will see in the following that it requires only one adjoint equation to solve for each iteration and each pair of data regardless of number of design variables. These properties make the shape functional suitable for the considered inverse problems and measurements. But for the case $\theta = 0$, i.e., the data is only given on the boundary, a different shape functional in contrast to (55) will be employed for reconstruction with other features, see the details in Subsection 4.3.

Then, the proposed IFE-based shape optimization method for reconstructing the material interface is to seek the design variable $\boldsymbol{\alpha}^*$ by carrying out the following constrained optimization:

$$\begin{aligned}
&\boldsymbol{\alpha}^* = \operatorname{argmin} \mathcal{J}_h(\boldsymbol{\alpha}), \quad \mathcal{J}_h(\boldsymbol{\alpha}) = \int_{\Omega_\theta} J_h(\bar{\mathbf{u}}^1(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) dX, \\
\text{with} \quad &J_h(\bar{\mathbf{u}}^1(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \sum_{k=1}^{K} \|\boldsymbol{\Phi}_m(X(\alpha), \alpha)\bar{\mathbf{u}}^k(\boldsymbol{\alpha}) + \boldsymbol{\Phi}_D(X(\alpha), \alpha)\bar{\mathbf{g}}_D - \mathbf{u}_\theta^k\|^2, \\
\text{subject to} \quad &\mathbf{A}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})\bar{\mathbf{u}}^k(\alpha) - \mathbf{F}^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \mathbf{0}, \quad k = 1, 2, \cdots, K,
\end{aligned} \tag{56}$$

where $\bar{\mathbf{u}}^k(\boldsymbol{\alpha})$, $\bar{\mathbf{g}}_D$, $k = 1, \cdots, K$, and $\boldsymbol{\Phi}_m(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$, $\boldsymbol{\Phi}_D(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ are given in (31) and (35) with $\mathcal{N}_h^m = \mathring{\mathcal{N}}_h$. We note that the discretized objective objective function in (56) is an optimal discretization of its continuous counterpart (55) because

$$\begin{aligned}
&\left|\mathcal{J}(\mathbf{u}^1(\Gamma), \cdots, \mathbf{u}^K(\Gamma), \Gamma) - \mathcal{J}_h(\bar{\mathbf{u}}^1(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha})\right| \\
=&\left|\sum_{k=1}^{K} \left(\int_{\Omega_\theta} \|\mathbf{u}^k - \mathbf{u}_\theta^k\|^2 dX - \int_{\Omega_\theta} \|\mathbf{u}_h^k - \mathbf{u}_\theta^k\|^2 dX \right)\right| \\
=&\left|\sum_{k=1}^{K} \left(\int_{\Omega_\theta} (\mathbf{u}^k - \mathbf{u}_h^k) \cdot (\mathbf{u}^k + \mathbf{u}_h^k - \mathbf{u}_\theta^k) dX \right)\right| \leqslant Ch^2,
\end{aligned} \tag{57}$$

where in the last inequality above we have used the Hölder's inequality and the optimality for $\mathbf{u}_h^k$ approximating $\mathbf{u}^k$ [36], and the constant C is independent of how the interface cuts each element. For the objective function

in (56), the computations for $\nabla J_h = [\partial_{x_1} J_h, \partial_{x_2} J_h]$, $\frac{\partial J_h}{\partial \alpha_j}$, $j \in \mathcal{D}$, and $\frac{\partial \mathcal{J}_h}{\partial \bar{\mathbf{u}}^k}$ can be carried with the following formulas :

$$\partial_{x_i} J_h = 2\left(\mathbf{\Phi}_m \bar{\mathbf{u}}^k + \mathbf{\Phi}_D^k \bar{\mathbf{g}}_D\right)^t \left(\partial_{x_i}\mathbf{\Phi}_m \bar{\mathbf{u}}^k + \partial_{x_i}\mathbf{\Phi}_D \bar{\mathbf{g}}_D - \partial_{x_i}\mathbf{u}_\theta^k\right), \quad i = 1, 2, \tag{58}$$

$$\frac{\partial J_h}{\partial \alpha_j} = 2\left(\mathbf{\Phi}_m \bar{\mathbf{u}}^k + \mathbf{\Phi}_D^k \bar{\mathbf{g}}_D\right)^t \left((\frac{\partial}{\partial \alpha_j}\mathbf{\Phi}_m)\bar{\mathbf{u}}^k + (\frac{\partial}{\partial \alpha_j}\mathbf{\Phi}_D)\bar{\mathbf{g}}_D\right), \quad j \in \mathcal{D}, \tag{59}$$

$$\frac{\partial \mathcal{J}_h}{\partial \bar{\mathbf{u}}^k} = 2\left(\int_{\Omega_\theta} \mathbf{\Phi}_m^t \mathbf{\Phi} \begin{bmatrix} \bar{\mathbf{u}}^k \\ \bar{\mathbf{g}}_D \end{bmatrix} dX - \int_{\Omega_\theta} \mathbf{\Phi}_m^t \mathbf{u}_\theta^k dX\right), \tag{60}$$

where $\frac{\partial}{\partial \alpha_j}\mathbf{\Phi}_m$ and $\frac{\partial}{\partial \alpha_j}\mathbf{\Phi}_D$ are formed by the shape derivatives of IFE shape functions which only need to be computed through (45) for those global basis functions whose supports are overlapped with interface elements, since the shape derivatives of IFE shape functions on non-interface elements vanish. The formulas (58)-(60) are implemented in the material derivative formula (46) for efficiently and accurately computing the shape sensitivities with respect to the interface variation.

The Lamé parameters $\lambda^+ = 1$, $\mu^+ = 2$ and $\lambda^- = 20$, $\mu^- = 40$ are used in these examples. For the displacement data provided on the whole domain, i.e., $\Omega_\theta = \Omega_1 = \Omega$, Figures 3, 4, and 5 present approximate interfaces in solid blue color together with the target interfaces in dotted red color. In order to give readers a visualization how the control points generate interface curve or guide its movement, in these plots we also show the location of control points by blue $*$. These results indicate that the IFE-based shape optimization algorithm converges quickly and recovers the target interfaces well with just a single displacement data, i.e., $K = 1$ in (55). They also demonstrate that the proposed IFE-based shape optimization algorithm can capture the shape sensitivities accurately and it can handle the large interface shape change on a fixed mesh independent of the interface movement.

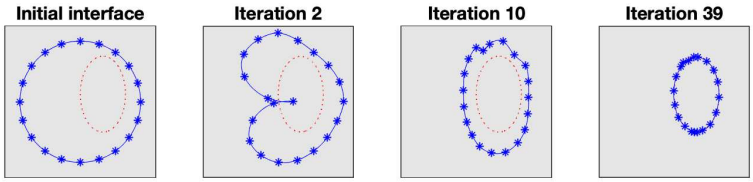


Figure 3: Elliptic interface with a single measurement on Ω

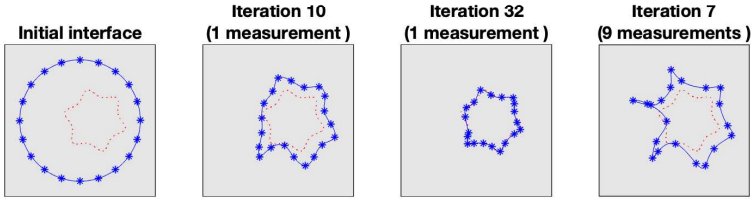


Figure 4: Star shape interface with a single measurement on Ω

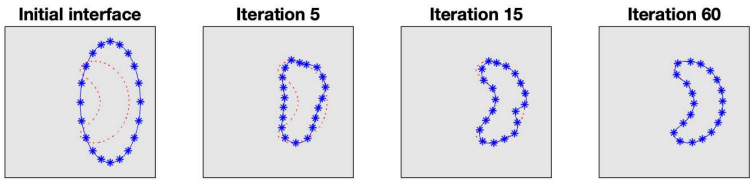


Figure 5: Kidney shape interface with a single measurement on Ω

Figure 6 presents numerical results for recovering the target interface in **Case 2** with displacement data provided on the data region $\Omega_{0.5}$ shown by the shaded region. We note that the majority of the target interface is outside the data region $\Omega_{0.5}$. Even with only a single measurement, i.e., $K = 1$, the target interface is recovered quite satisfactorily to a certain extend after 37 iterations. After that, we employ $K = 9$ measurements and the reconstruction is indeed improved. These numerical results demonstrate that more measurements can provide more information about the structure inside an elastic body for the proposed IFE-based shape optimization algorithm to consequently produce a better reconstruction.

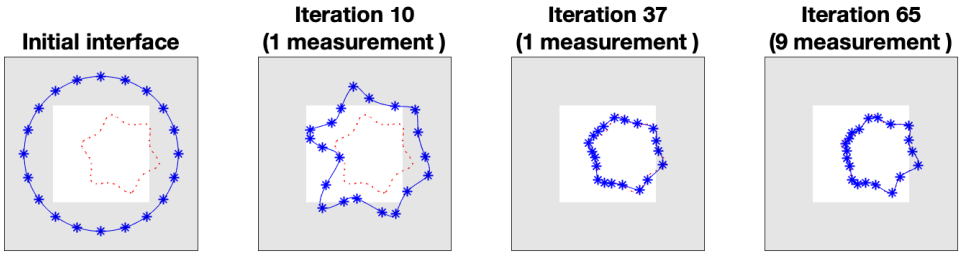


Figure 6: Star shape interface with a single and multiple measurements on $\Omega_{0.5}$

4.2 Boundary Stress Data and Internal Displacement Measurements

In this group of examples, we consider the inverse geometric problems in which the boundary stress data $\mathbf{g}_N^k$ and internal displacement measurement $\mathbf{u}_\theta^k$ are given for $K \geq 1$ such that shape functional is

$$\mathcal{J}(\mathbf{u}^1(\Gamma), \cdots, \mathbf{u}^K(\Gamma), \Gamma) = \sum_{k=1}^{K} \int_{\Omega_\theta} \|\mathbf{u}^k - \mathbf{u}_\theta^k\|^2 dX, \tag{61}$$

where $\mathbf{u}^k$, $k = 1, \cdots, K$ are solutions to (3)-(5) with the pure Neumann boundary conditions given by the stress data $\mathbf{g}_N^k$. This functional is similar to (55) such that formulas for discretized J_h, $\mathcal{J}_h$ and their related derivatives ∇J_h, $\frac{\partial J_h}{\partial \alpha_j}$, $j \in \mathcal{D}$, $\frac{\partial \mathcal{J}_h}{\partial \bar{\mathbf{u}}^k}$ are similar to (56)-(60). Also it is an optimal discretization of the continuous one of which the derivation is similar to (57). The Lamé parameters used in these examples are $\lambda^+ = 1$, $\mu^+ = 2$ and $\lambda^- = 20$, $\mu^- = 40$.

To avoid redundancy and also to show how the reconstructed curve matches the exact interface more clearly, we will not show control points in plots in this and the next section. Figures 7 and 8 present numerical results for interface curves in **Case 1** and **Case 2** with the displacement measurement given on $\Omega_{0.1}$ which does not cover the target curves. For the simple elliptic target interface, the proposed IFE-based shape optimization algorithm is able to produce a satisfactory reconstruction with just one stress-displacement data. However, for the more complicated star shape interface, with one stress-displacement data, the algorithm becomes staggered after 32 iterations. Then we restart the iteration with 9 stress-displacement data and the IFE-based shape optimization produces a numerical interface more satisfactory.

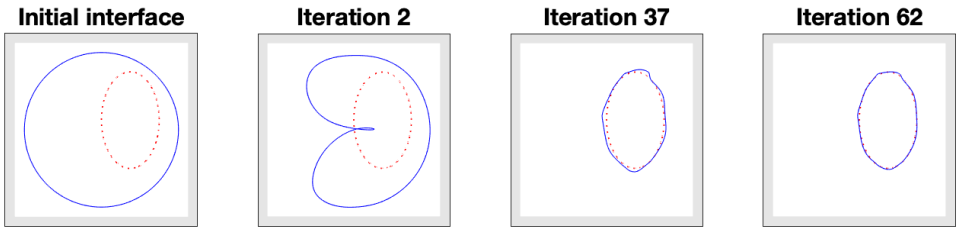


Figure 7: Elliptic interface with a single measurement on $\Omega_{0.1}$

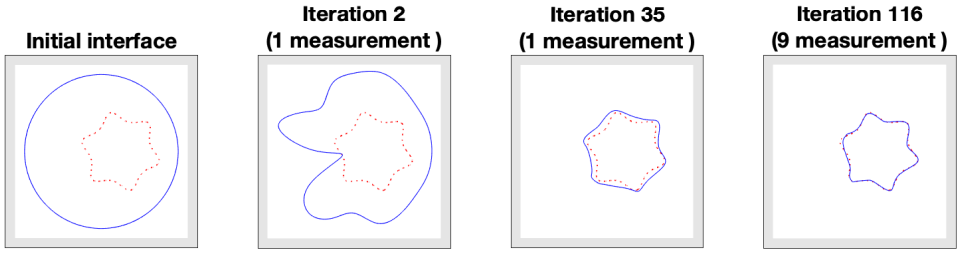


Figure 8: Star shape interface with single/multiple measurement on $\Omega_{0.1}$

Figures 9 and 10 present numerical curves generated with stress-displacement data for a kidney shape target interface in **Case 3** that has a large concave portion. The results demonstrate that using more measurements in reconstruction leads to a better numerical interface. We note that the reconstruction with displacement measurement on $\Omega_{0.2}$ is better than the one from $\Omega_{0.1}$ suggesting that the larger size/amount of data leads to a better reconstruction.

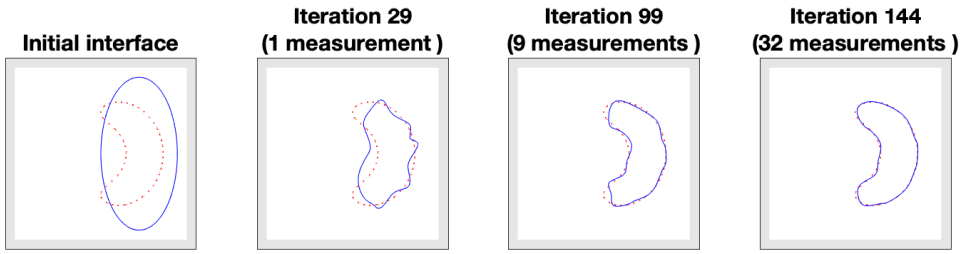


Figure 9: Kidney shape interface with single/multiple measurements on $\Omega_{0.1}$

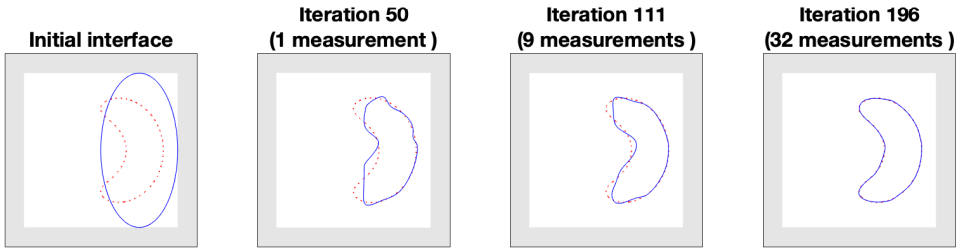
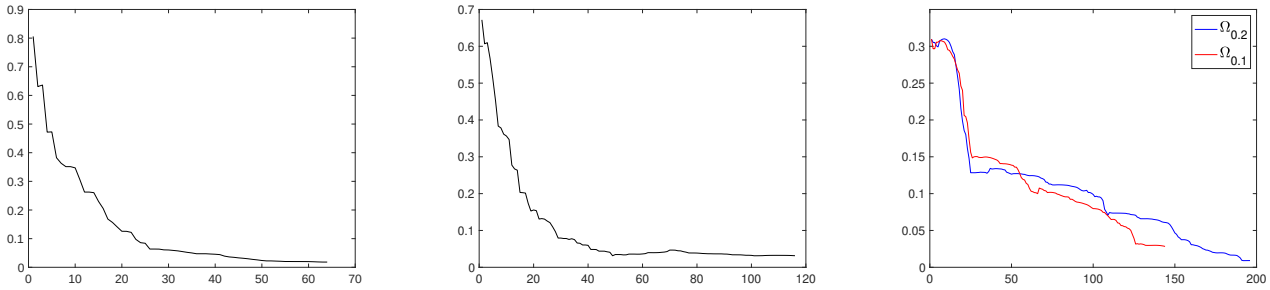


Figure 10: Kidney shape interface with single/multiple measurements on $\Omega_{0.2}$

For this group of examples, we also show the convergence the proposed algorithm by investigating the Hausdorff distance between the numerical curve and exact interface. The Hausdorff distance is also used in [7] as a criteria to terminate the iteration. We recall that the Hausdorff distance between two arbitrary sets $\mathcal{A}$ and $\mathcal{B}$ is defined as follows:

$$d_H(\mathcal{A},\mathcal{B}) = \min_{a\in\mathcal{A}}\{\min_{b\in\mathcal{B}} d(a,b)\}, \tag{62}$$

where d is certain distance. In our case, we let $\mathcal{A}$ and $\mathcal{B}$ be the numerical curve and target interface, respectively, and let $d(\cdot,\cdot)$ be the standard Euclid distance in the plane. The evolution of Haudorff distance for the experiments above are presented in Figure 11. In these plots, we can clearly observe a quick convergence at the beginning of optimization. We believe this may be the consequence that the proposed method can accurately capture the shape sensitivities.



(a) Elliptic interface (b) Star shape interface (c) Kidney shape interface

Figure 11: Hausdorff distance vs iterations

4.3 Stress-Displacement Data on The Boundary

In this group of examples, we consider the inverse problems to recover the material interface from the stress and the related displacement data both measured on the boundary such that a Kohn-Vogelius type functional [54, 63] is used in the shape optimization:

$$\mathcal{J}(\mathbf{u}_D^1(\Gamma),\cdots,\mathbf{u}_D^K(\Gamma),\mathbf{u}_N^1(\Gamma),\cdots,\mathbf{u}_N^K(\Gamma),\Gamma)=\sum_{k=1}^{K}\int_{\Omega}\|\mathbf{u}_D^k-\mathbf{u}_N^k\|^2dX, \tag{63}$$

where $\mathbf{u}_N^k$ are solutions to (3)-(5) with the pure Neumann boundary conditions given by the stress data $\mathbf{g}_N^k$, while $\mathbf{u}_D^k$ are solutions to (3)-(5) with the pure Dirichlet boundary conditions given by the displacement measurements $\mathbf{g}_D^k=\mathbf{u}_\theta^k|_{\partial\Omega}$.

Again, it is easy to verify that the shape functional (63) vanishes when the interface is the actual one. Other shape functionals can also be formulated to use the available data. For instance, a shape functional similar to (61) with $\theta=0$ can be used in this situation, i.e., considering the mismatch on the boundary between the solution with the Neumann boundary condition and Dirichlet data or the mismatch between the solution with the Dirichlet boundary condition and Neumann data. We refer readers to [39] for a comparison between these two types of shape functionals for a similar inverse problem with only boundary data available. We prefer the Kohn-Vogelius type functional (63) rather than the other type since our numerical experiments suggest that, in general, it can produce a better reconstruction. We believe the following two reasons might contribute to the better performance. Firstly, when the shape functional is defined by integration over the boundary such as the one in (61), it suffers from the well known loss of accuracy of finite element solutions restricted to the boundary. Secondly, the self-regularizing feature of the Kohn-Vogelius type functional as mentioned in [41] may help improve the reconstruction. On the other hand, we note that, in order to use the Kohn-Vogelius type functional (63), we need to solve two direct interface problems and two adjoint problems in each iteration and for each pair of stress-displacement data, and this is twice more expensive than using a shape functional like (61).

The proposed IFE-based shape optimization method for reconstructing the material interface is to seek the

design variable $\boldsymbol{\alpha}^*$ by carrying out the following constrained optimization:

$$
\begin{aligned}
&\boldsymbol{\alpha}^* = \operatorname{argmin} \mathcal{J}_h(\boldsymbol{\alpha}), \quad \mathcal{J}_h(\boldsymbol{\alpha}) = \int_{\Omega_d} J_h(\bar{\mathbf{u}}_D^1(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}_D^K(\boldsymbol{\alpha}), \bar{\mathbf{u}}_N^1(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}_N^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) dX, \\
&\text{with} \quad J_h(\bar{\mathbf{u}}_D^1(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}_D^K(\boldsymbol{\alpha}), \bar{\mathbf{u}}_N^1(\boldsymbol{\alpha}), \cdots, \bar{\mathbf{u}}_N^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\
&\qquad = \sum_{k=1}^{K} \left\| \boldsymbol{\Phi}_m(X(\alpha), \alpha) \bar{\mathbf{u}}_D^k(\boldsymbol{\alpha}) + \boldsymbol{\Phi}_D(X(\alpha), \alpha) \bar{\mathbf{g}}_D - \boldsymbol{\Phi}(X(\alpha), \alpha) \bar{\mathbf{u}}_N^k(\boldsymbol{\alpha}) \right\|^2, \\
&\text{subject to} \quad \mathbf{A}_D^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \bar{\mathbf{u}}_D^k(\alpha) - \mathbf{F}_D^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \mathbf{0}, \quad k = 1, 2, \cdots, K, \\
&\qquad \mathbf{A}_N^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \bar{\mathbf{u}}_N^k(\alpha) - \mathbf{F}_N^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \mathbf{0}, \quad k = 1, 2, \cdots, K,
\end{aligned}
\tag{64}
$$

where $\mathbf{A}_D^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})/\mathbf{F}_D^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ and $\mathbf{A}_N^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})/\mathbf{F}_N^k(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ are the matrices/vectors in the IFE system (31) corresponding to the Dirichlet and Neumann boundary conditions, respectively. The formulas for ∇J_h, $\frac{\partial J_h}{\partial \alpha_j}$, $j \in \mathcal{D}$ and $\frac{\partial \mathcal{J}_h}{\partial \bar{\mathbf{u}}_D^k}$, $\frac{\partial \mathcal{J}_h}{\partial \bar{\mathbf{u}}_N^k}$ are similar to (58)-(60). The optimality of discretized shape functional in this case follows from a similar derivation as (57). All the numerical results in this group of examples are generated with the Lamé parameters $\lambda^- = 1$, $\lambda^+ = 2$, $\mu^- = 5$ and $\mu^+ = 10$.

Figure 12 presents numerical interfaces constructed by the IFE-based shape optimization algorithm for the elliptic interface curve in **Case 1**. Even though only one stress-displacement date on the boundary is used, the numerical interface quickly captures the basic shape of the target interface after about 15 iterations, and it converges to a much better approximation after 51 iterations.

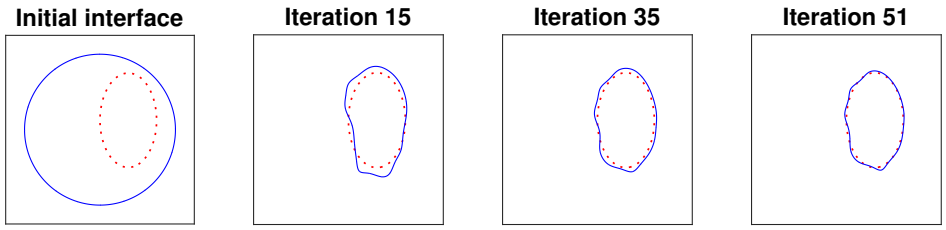


Figure 12: Elliptic interface with single measurement

Numerical results for the more complicated star shape target interface in **Case 2** are presented in Figure 13. With just a single stress-displacement data, the shape optimization quickly (after only 17 iteration) generates a reconstruction quite satisfactory to a certain sense. Then, from the numerical interface at iteration 17, we restart the iteration with additional 8 measurements so that the IFE-based shape optimization produces a good reconstruction at iteration 57.

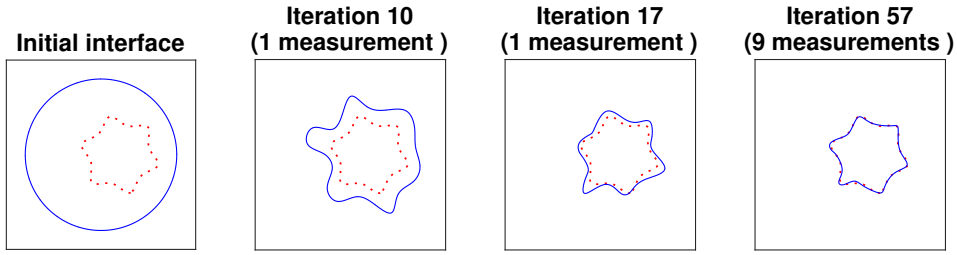


Figure 13: Star shape interface with single/multiple measurements

For the kidney shape target interface in **Case 3** with a large concave portion, with just one stress-displacement data after 41 iterations, the algorithm produces a numerical interface that approximates the target curve well over the portion close to the boundary, but less satisfactory for the concave part in the middle of the domain. With more stress-displacement data, the algorithm generates better numerical interfaces even

in the concave part. This example indicates that the proposed algorithm is able to recover the interface with a large concave portion with sufficient stress-displacement data pairs measured only on the boundary.

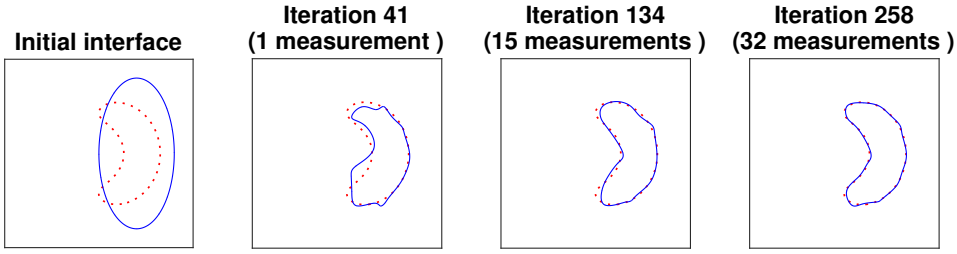


Figure 14: Kidney shape interface with single/multiples measurements

We finally present two experiments on the reconstruction of the star shape interface with the Dirichlet data polluted by Gaussian noise. Figure 15 and Figure 16 present reconstruction results for data whose noise level are set to be 1% and 5%, respectively. We obverse the noise in the data has a strong impact on the reconstruction results when only a single measurement is available, see the iteration 60 and iteration 24 shown in the third plot in Figures 15 and 16, compared with their counterparts in Figure 13. However, we note that, with multiple measurements that nevertheless contain noise, the proposed IFE-based shape optimization algorithm can still produces a reasonably good reconstruction comparable to the one presented in Figure 13 which is generated with data without noise. Therefore, in a certain sense, we believe the proposed algorithm is robust with respect to the noise.

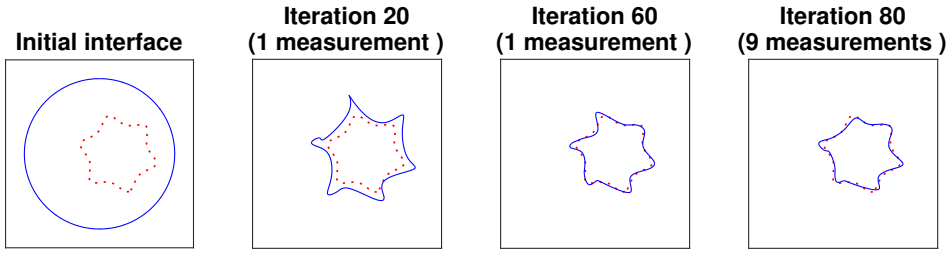


Figure 15: Star shape interface with 1% noise

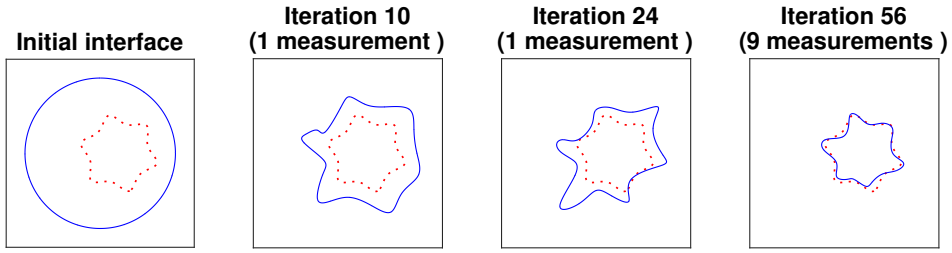


Figure 16: Star shape interface with 5% noise

4.4 Observations And Comparisons

Because of the ill-posedness of the considered inverse problems, it is known, see [49, 74] for example, that the portion of the true interface curve close to the center of the domain is in general more difficult to retrieve from the data on the boundary than the portion closer to the boundary. We have also observed such a phenomenon in our numerical experiments. But we note that the proposed algorithm has the built-in mechanism that can take advantage of multiple pairs of boundary data to produce a reasonably good approximation to the interface curve near the center.

We note that the local minima of the shape functions in (61) and (63) may appear when the interface totally moves out of the domain Ω or deteriorates into one point, and then the derivatives of shape functions vanish. Since the considered shape functions are highly non-linear and non-convex, other types of local minima may also exit which is a well-known issue for gradient type methods for inverse problems [15]. We also refer readers to [19, 46] for discussions on the related issue of various numerical methods. Moreover we believe some local minima may be intrinsic in the sense that the uniqueness of the considered inverse problem itself is a theoretical concern [2, 3]. As mentioned in [19, 46, 15], these pitfalls may be circumvented by some a-prior information of the true interface, i.e., a relatively good initial guess, or certain special numerical techniques.

However in our numerical experiments, we have observed that the numerical curve is rarely trapped at any local minima far away from the true interface. As a matter of fact, in our extensive experiments, we have observed that the numerical curve in general can quickly move into the neighborhood of the exact interface which means that method can efficiently recover the essential feature of the exact interface within just a few iterations, see, for example, Figure 11 and related discussions. Moreover, we note that the proposed method has no explicit regularization added in the shape functional to ensure convergence or to avoid local minima. Similar feature is also observed in [39] for a time-dependent inverse problem. We speculate that this is mainly thanks to a "hidden" regularization mechanism of the proposed algorithm which can be understood from the following two perspectives. The first one is the low dimensionality of the searching space due to the chosen relatively small number of control points (also see the discussion at the beginning of this section) for the numerical interface. Another one is the capability of the proposed algorithm to search for the interface in a manner of a topological equivalence which can be understood as a geometric restriction naturally imposed. In other words, we believe it is less possible for the numerical interface to be trapped at a local minima if the search is performed in a relatively small space, and we think the topology information of the true interface can benefit the proposed method for solving the considered geometric inverse problem.

This "hidden" regularization is significant in contrast to the interface capturing methods in the literature, such as level-set methods, which in general rely on some explicit regularization techniques to guarantee convergence, such as the geometric regularization [38, 46, 70], total variation regularization [19, 23], and the Tikhonov regularization [50]. The advantage of these methods is the capability to represent a large group of interface curves and to handle topological changes. But they have to search for the interface in a much larger space, and consequently need much more iterations to converge in general. Most of these works in the literature are for scalar elliptic interface problems such as the electrical impedance tomography (EIT). We note that an algorithm similar to the one proposed here has already been applied to scalar elliptic interface problems in [37] where we can observe similar features and a significant improvement on the efficiency and accuracy for reconstruction in some representative situations. In addition, we refer readers to [7] for an application of a level-set method to an elasticity inverse problem as an unstrict comparison because of its slightly different set-up. We would like to mention that the method in [7] is unable to capture the concave-in portion of a non-convex interface, while the reconstruction results of the proposed method for a star shape interface and a kidney shape interface are quite satisfactory in the concave-in portion.

As a comparison in parametrization for the interface, we note the shape optimization methods in the route of interface tracking approaches for shape representation [26, 39, 43, 65, 66]. Most of these methods parameterize/represent the interface shape by Fourier series or harmonic functions which in some situations are limited to star shape curves with a known center [39]. In contrast, the cubic spline parametrization employed in the proposed method here can be used to approximate more general interface curves, and other types of parametrization can be easily adopted for the proposed method.

Another feature of the proposed method distinguished from many methods for interface inverse problems in the literature is its capability to perform all the computations, including computing direct interface problems, shape functionals, and shape sensitivities optimally and stably on an interface-independent mesh which can be fixed once for all. Neither the accuracy or the conditioning of the stiffness matrix for solving the direct interface problem by IFE methods will deteriorate while the interface evolves in the background fixed mesh according to an optimization procedure. Because of this property together with the "hidden" regularization mechanism mentioned above, we believe the proposed algorithm is advantageous for solving geometric inverse problems.

5 Conclusions And Further Issues

In a summary, we have developed an IFE-based parameterized shape optimization algorithm to solve geometric inverse problems for linear elasticity systems. The proposed work has several advantages. Firstly, all the computations in the optimization can be carried on a fixed mesh independent of the interface change which circumvents the burden of mesh generation or movement in contrast to the related traditional approaches in the literature. Secondly, explicit formulas are derived to accurately compute the shape sensitivities on a fixed mesh, and in this procedure almost all the computations are carried out on interface elements only, not on the whole mesh. Therefore, the proposed algorithm is both accurate and efficient for updating the interface during iterations. Thirdly, as discussed already, the parametrization technique of the interface equips the shape optimization algorithm with a self-regularization mechanism to reduce the adverse effects from the ill-posedness of the original inverse problem. The numerical results demonstrate the efficiency and accuracy of the proposed algorithm as well as its stability with respect to measurement noise in the data for the inverse problems.

On the other hand, the proposed algorithm poses many questions for future researches. In particular, in the current implementation, we need to start the optimization from a numerical curve which is topologically related to the exact interface, namely a relatively good initial guess capturing some a-priori topological information is required. Therefore it is interesting to develop an algorithm to recover such a satisfactory initial guess, for example, we believe direct methods [22, 45] have such a potential. In addition, the extension of the proposed algorithm to three dimensions is also desirable. Although IFE methods have been developed for three dimensional interface problems [34, 51], we believe the application to three dimensional geometric inverse problems requires more sophisticated interface parametrization techniques.

In addition, we believe that the proposed method can be employed to deal with more complicated inverse problems. For example, the IFE-based shape optimization algorithm proposed in this article for elasticity systems and the one in [37] for scalar elliptic equations may be combined to solve reconstruction problems in thermoelasticity [8] governed by different physics. In addition to the geometric inverse problems, we believe that the proposed algorithm also has the potential to be extended to solve optimal structure designing problems related to mechanics such as the optimization of elastic compliance or target displacement [4, 5]. We note that the optimization process of these problems can inevitably involve topological changes, and interface capturing methods such as level-set methods might be more suitable. Due to the more complicated geometry or topology, the computations in general can benefit from a reliable numerical solver for the underling PDEs that can use a fixed mesh independent of the boundary/interface. This is where we believe the IFE-based shape optimization methods can make contributions.

Appendix A Formulas of Shape Derivatives

In this appendix, we present a group of derivatives for quantities in the shape derivative formula (45) for IFE shape functions. First of all, for $L(X)$, we have

$$\frac{\partial L}{\partial P} = \frac{(X-P)^t \bar{\mathbf{t}}\bar{\mathbf{n}}^t}{\|P-Q\|} + \bar{\mathbf{n}}, \quad \frac{\partial L}{\partial Q} = -\frac{(X-Q)^t \bar{\mathbf{t}}\bar{\mathbf{n}}^t}{\|P-Q\|}, \tag{65}$$

where $\bar{\mathbf{n}}$ and $\bar{\mathbf{t}}$ are the normal and tangential vectors given in (10) for the line l connecting the points P and Q. For the coefficient c_0 given in (12), and $E = P$ or Q, we have

$$\frac{\partial \mathbf{c}_0}{\partial E} = \left(\frac{\partial K^{-1}}{\partial E}\right) \hat{\sigma}(\boldsymbol{\psi}_T^{int,+})(F_0)\bar{\mathbf{n}} + K^{-1}\left(\hat{\sigma}(\boldsymbol{\psi}_T^{int,+})(F_0)\frac{\partial \bar{\mathbf{n}}}{\partial E} + \sum_{s=1,2}\hat{\sigma}(\boldsymbol{\psi}_T^{int,+})\left(\frac{\partial F_0}{\partial E}\right)\bar{\mathbf{n}}\right), \tag{66}$$

$$\text{with } \frac{\partial K^{-1}}{\partial E} = 2\frac{\partial R}{\partial E}\begin{bmatrix} (\lambda^- + 2\mu^-)^{-1} & 0 \\ 0 & (\mu^-)^{-1} \end{bmatrix} R^t, \quad \frac{\partial R}{\partial E} = \left[\frac{\partial \bar{\mathbf{n}}}{\partial E}, \frac{\partial \bar{\mathbf{t}}}{\partial E}\right], \tag{67}$$

$$\text{and } \frac{\partial \bar{\mathbf{n}}}{\partial P} = \frac{\bar{\mathbf{t}}\bar{\mathbf{n}}^t}{\|P-Q\|}, \quad \frac{\partial \bar{\mathbf{n}}}{\partial Q} = -\frac{\bar{\mathbf{t}}\bar{\mathbf{n}}^t}{\|P-Q\|}, \quad \frac{\partial \bar{\mathbf{t}}}{\partial P} = -\frac{\bar{\mathbf{n}}\bar{\mathbf{n}}^t}{\|P-Q\|}, \quad \frac{\partial \bar{\mathbf{t}}}{\partial Q} = \frac{\bar{\mathbf{n}}\bar{\mathbf{n}}^t}{\|P-Q\|}, \tag{68}$$

At last, for the coefficient vector $\mathbf{c}_i$ given in (13) and $E = P$ or Q, we have

$$\begin{aligned}
\frac{\partial \mathbf{c}_i}{\partial \alpha_j} =& -\frac{\partial}{\partial E}\left(L(A_i)R^t\Xi^{-1}R\right)\sum_{k\in\mathcal{I}}\left(v_k^1\hat{\sigma}(\boldsymbol{\psi}_{k,T}^{non})(F_0)\bar{\mathbf{n}}+v_k^2\hat{\sigma}(\boldsymbol{\psi}_{k+4,T}^{non})(F_0)\bar{\mathbf{n}}\right)\\
&+L(A_i)R^t\Xi^{-1}R\sum_{k\in\mathcal{I}}\left(v_k^1\hat{\sigma}(\boldsymbol{\psi}_{k,T}^{non})(F_0)\frac{\partial\bar{\mathbf{n}}}{\partial E}+v_k^2\hat{\sigma}(\boldsymbol{\psi}_{k+4,T}^{non})(F_0)\frac{\partial\bar{\mathbf{n}}}{\partial E}\right) \qquad (69)\\
&+L(A_i)R^t\Xi^{-1}R\sum_{k\in\mathcal{I}}\left(v_k^1\hat{\sigma}(\boldsymbol{\psi}_{k,T}^{non})\left(\frac{\partial F_0}{\partial E}\right)\bar{\mathbf{n}}+v_k^2\hat{\sigma}(\boldsymbol{\psi}_{k+4,T}^{non})\left(\frac{\partial F_0}{\partial E}\right)\bar{\mathbf{n}}\right),
\end{aligned}$$

$$\text{with}\quad \frac{\partial}{\partial E}(L(A_i)R^t\Xi^{-1}R)=\frac{\partial L(A_i)}{\partial E}R^t\Xi^{-1}R+L(A_i)\left(\left(\frac{\partial R^t}{\partial E}\right)\Xi^{-1}R-R^t\Xi^{-1}\left(\frac{\partial\Xi}{\partial E}\right)\Xi^{-1}R+R^t\Xi^{-1}(D_{\alpha_j}R)\right),$$

$$\text{and}\quad \frac{\partial\Xi}{\partial E}=\begin{bmatrix}((\lambda^+-\lambda^-)+2(\mu^+-\mu^-))\ \partial g_n(F_0)/\partial E & (\lambda^+-\lambda^-)\ \partial g_t(F_0)/\partial E\\ (\mu^+-\mu^-)\ \partial g_t(F_0)/\partial E & (\mu^+-\mu^-)\ \partial g_n(F_0)/\partial E\end{bmatrix}, \qquad (70)$$

$$\frac{\partial g_n(F_0)}{\partial E}=\sum_{i\in\mathcal{I}^-}\left(\frac{\partial L(A_i)}{\partial E}\right)\nabla\psi_{i,T}^{non}(F_0)\bar{\mathbf{n}}+L(A_i)\left(\nabla\psi_{i,T}^{non}\left(\frac{\partial F_0}{\partial E}\right)\right)\cdot\bar{\mathbf{n}}+L(A_i)\nabla\psi_{i,T}^{non}(F_0)\frac{\partial\bar{\mathbf{n}}}{\partial E},$$

$$\frac{\partial g_t(F_0)}{\partial E}=\sum_{i\in\mathcal{I}^-}\left(\frac{\partial L(A_i)}{\partial E}\right)\nabla\psi_{i,T}^{non}(F_0)\bar{\mathbf{t}}+L(A_i)\left(\nabla\psi_{i,T}^{non}\left(\frac{\partial F_0}{\partial E}\right)\right)\cdot\bar{\mathbf{t}}+L(A_i)\nabla\psi_{i,T}^{non}(F_0)\frac{\partial\bar{\mathbf{t}}}{\partial E},$$

in which $\frac{\partial L(A_i)}{\partial E}$ is given in (65) with $X=A_i$, $\frac{\partial\bar{\mathbf{t}}}{\partial E}$, $\frac{\partial\bar{\mathbf{t}}}{\partial E}$ are given in (68), and $g_n(F_0)$ and $g_t(F_0)$ are given in (16).

Appendix B Material Derivatives of Local IFE Matrices

This appendix provides formula for the material derivatives of the local IFE matrices given in (25a). The key technique used here is the standard formula given in Lemma 3.3 of [40] relating material derivatives, shape derivatives and velocity fields for shape functionals defined by integration. Without loss of generality, we consider the local IFE matrices associated with an interface element $T\in\mathcal{T}_h^i$ and interface edge $e=A_1P\cup PA_2\in\mathcal{E}_h^i$ with the interface-mesh intersection point P and $A_1P\subset\Omega^-$, $A_2P\subset\Omega^+$. Then, we have the following formulas for the material derivatives of local IFE matrices:

$$\begin{aligned}
D_{\alpha_j}\mathbf{K}_T^1=&\left(\int_T 2\mu\epsilon(\frac{\partial\boldsymbol{\psi}_{p,T}}{\partial\alpha_j}):\epsilon(\boldsymbol{\psi}_{q,T})dX\right)_{p,q}+\left(\int_T 2\mu\epsilon(\frac{\partial\boldsymbol{\psi}_{p,T}}{\partial\alpha_j}):\epsilon(\boldsymbol{\psi}_{q,T})dX\right)_{p,q}^t\\
&+\left(\int_T 2\mu\nabla(\epsilon(\boldsymbol{\psi}_{p,T}):\epsilon(\boldsymbol{\psi}_{q,T}))\cdot\mathbf{V}_T^j dX\right)_{p,q}\\
&+\left(\sum_{i=1}^4\int_{T_i}2\mu\epsilon(\boldsymbol{\psi}_{p,T}):\epsilon(\boldsymbol{\psi}_{q,T})\ dX\ tr\left((D_{\alpha_j}\mathbf{J}_i)\mathbf{J}_i^{-1}\right)\right)_{p,q}, \qquad (71a)\\
D_{\alpha_j}\mathbf{K}_T^2=&\left(\int_T\lambda(\nabla\cdot\frac{\partial\boldsymbol{\psi}_{p,T}}{\partial\alpha_j})(\nabla\cdot\boldsymbol{\psi}_{q,T})dX\right)_{p,q}+\left(\int_T\lambda(\nabla\cdot\frac{\partial\boldsymbol{\psi}_{p,T}}{\partial\alpha_j})(\nabla\cdot\boldsymbol{\psi}_{q,T})dX\right)_{p,q}^t\\
&+\left(\int_T\lambda\nabla((\nabla\cdot\boldsymbol{\psi}_{p,T})(\nabla\cdot\boldsymbol{\psi}_{q,T}))\cdot\mathbf{V}_T^j dX\right)_{p,q}\\
&+\left(\sum_{i=1}^4\int_{T_i}\lambda(\nabla\cdot\boldsymbol{\psi}_{p,T})(\nabla\cdot\boldsymbol{\psi}_{q,T})\ dX\ tr\left((D_{\alpha_j}\mathbf{J}_i)\mathbf{J}_i^{-1}\right)\right)_{p,q}, \qquad (71b)\\
D_{\alpha_j}\mathbf{E}_e^{r_1r_2}=&\left(\int_e\sigma(\frac{\partial\boldsymbol{\psi}_{p,T^{r_1}}}{\partial\alpha_j}):(\boldsymbol{\psi}_{q,T^{r_2}}\otimes\mathbf{n}_e^{r_2})ds\right)_{p,q}+\left(\int_e\sigma(\boldsymbol{\psi}_{p,T^{r_1}}):(\frac{\partial\boldsymbol{\psi}_{q,T^{r_2}}}{\partial\alpha_j}\otimes\mathbf{n}_e^{r_2})ds\right)_{p,q}\\
&+\left(\sigma(\boldsymbol{\psi}_{p,T^{r_1}}^+):(\boldsymbol{\psi}_{q,T^{r_2}}^+\otimes\mathbf{n}_e^{r_2})|_P-\sigma(\boldsymbol{\psi}_{p,T^{r_1}}^-):(\boldsymbol{\psi}_{q,T^{r_2}}^-\otimes\mathbf{n}_e^{r_2})|_P\right)_{p,q}\frac{D_{\alpha_j}P\cdot(A_2-A_1)}{\|A_2-A_1\|}, \qquad (71c)
\end{aligned}$$

$$D_{\alpha_j}\mathbf{G}_e^{r_1r_2} = \frac{\rho}{|e|}\left(\int_e (\frac{\partial \boldsymbol{\psi}_{p,T^{r_1}}}{\partial \alpha_j})\cdot(\boldsymbol{\psi}_{q,T^{r_2}})ds\right)_{p,q} + \frac{\rho}{|e|}\left(\int_e (\frac{\partial \boldsymbol{\psi}_{p,T^{r_1}}}{\partial \alpha_j})\cdot(\boldsymbol{\psi}_{q,T^{r_2}})ds\right)^t_{p,q}, \tag{71d}$$

where $j \in \mathcal{D}$, the subscripts $p, q = 1, \cdots, 8$ are for the entries in the corresponding matrices or vectors, $\frac{\partial \boldsymbol{\psi}_{p,T}}{\partial \alpha_j}$ are the shape derivatives of the vector IFE shape functions given by the formula (45), and the velocity field $\mathbf{V}_T^j$ is defined in (42).

Appendix C Material Derivatives of Local IFE Vectors

This appendix provides formula for the material derivatives of the local IFE vectors given in (26a) and (30). As before, without loss of generality, we consider the local IFE vectors associated with an interface element $T \in \mathcal{T}_h^i$ and interface edge $e = A_1P \cup PA_2 \in \mathcal{E}_h^i$ with the interface-mesh intersection point P and $A_1P \subset \Omega^-$, $A_2P \subset \Omega^+$. In addition, we assume that the boundary terms $\mathbf{g}_D^k$, $\mathbf{g}_N^k$ and the force terms $\mathbf{F}^k$ are fixed and independent of the evolution of the interface. Then, we have the following formulas for the material derivatives of local IFE vectors:

$$D_{\alpha_j}\mathbf{F}_T^k = \left(\int_T \mathbf{f}^k \cdot \frac{\partial \boldsymbol{\psi}_{p,T}}{\partial \alpha_j} dX\right)_p + \left(\int_T \nabla(\mathbf{f}^k \cdot \boldsymbol{\psi}_{p,T})\cdot \mathbf{V}_T^j dX\right)_p + \left(\sum_{i=1}^4 \int_{T_i} \mathbf{f}^k \cdot \boldsymbol{\psi}_{p,T}\ dX\ tr\big((D_{\alpha_j}\mathbf{J}_i)\mathbf{J}_i^{-1}\big)\right)_p, \tag{72a}$$

$$D_{\partial_j}\mathbf{B}_e^k = \left(\int_e \mathbf{g}_D^k \cdot \sigma(\frac{\partial \boldsymbol{\psi}_{p,T}}{\partial \alpha_j})\mathbf{n}_e ds\right)_p + \left(\mathbf{g}_D^k \cdot \sigma(\boldsymbol{\psi}_{p,T}^+)\mathbf{n}_e|_P - \mathbf{g}_D^k \cdot \sigma(\boldsymbol{\psi}_{p,T}^-)\mathbf{n}_e|_P\right)_p \frac{D_{\alpha_j}P\cdot(A_2-A_1)}{\|A_2-A_1\|}, \tag{72b}$$

$$D_{\partial_j}\mathbf{C}_e^k = \frac{\rho}{|e|}\left(\int_e \mathbf{g}_D^k \cdot \frac{\partial \boldsymbol{\psi}_{p,T}}{\partial \alpha_j} ds\right)_p + \frac{\rho}{|e|}\left(\mathbf{g}_D^k \cdot \boldsymbol{\psi}_{p,T}^+|_P - \mathbf{g}_D^k \cdot \boldsymbol{\psi}_{p,T}^-|_P\right)_p \frac{D_{\alpha_j}P\cdot(A_2-A_1)}{\|A_2-A_1\|}, \tag{72c}$$

$$D_{\partial_j}\mathbf{N}_e^k = \left(\int_e \mathbf{g}_N^k \cdot \frac{\partial \boldsymbol{\psi}_{p,T}}{\partial \alpha_j} ds\right)_p + \left(\mathbf{g}_N^k \cdot \boldsymbol{\psi}_{p,T}^+|_P - \mathbf{g}_N^k \cdot \boldsymbol{\psi}_{p,T}^-|_P\right)_p \frac{D_{\alpha_j}P\cdot(A_2-A_1)}{\|A_2-A_1\|}, \tag{72d}$$

(72e)

$$D_{\alpha_j}\mathbf{R}_T^l = \left(\int_T \frac{\partial \psi_{p,T}^l}{\partial \alpha_j} dX\right)_p + \left(\int_T \nabla \psi_{p,T}^l \cdot \mathbf{V}_T^j dX\right)_p + \left(\sum_{i=1}^4 \int_{T_i} \psi_{p,T}^l\ dX\ tr\big((D_{\alpha_j}\mathbf{J}_i)\mathbf{J}_i^{-1}\big)\right)_p, \quad l = 1, 2, \tag{72f}$$

$$D_{\alpha_j}\mathbf{R}_T^3 = \left(\int_T \frac{\partial rot(\boldsymbol{\psi}_{p,T})}{\partial \alpha_j} dX\right)_p + \left(\int_T \nabla(rot(\boldsymbol{\psi}_{p,T}^l))\cdot \mathbf{V}_T^j dX\right)_p + \left(\sum_{i=1}^4 \int_{T_i} rot(\boldsymbol{\psi}_{p,T}^l)\ dX\ tr\big((D_{\alpha_j}\mathbf{J}_i)\mathbf{J}_i^{-1}\big)\right)_p, \tag{72g}$$

where $j \in \mathcal{D}$, the subscripts $p = 1, \cdots, 8$ are for the entries the in corresponding vectors, $\frac{\partial \boldsymbol{\psi}_{p,T}}{\partial \alpha_j}$ are the shape derivatives of the vector IFE shape functions given by the formula (45), and the velocity fields $\mathbf{V}_T^j$ are defined in (42).

References

[1] J. P. Agnelli, A. De Cezaro, A. Leitão, and M. Marques Alves. On the identification of piecewise constant coefficients in optical diffusion tomography by level set. *ESAIM: Control, Optimisation and Calculus of Variations*, 23(2):663–683, 2017.

[2] G. Alessandrini, A. Morassi, and E. Rosset. Detecting an inclusion in an elastic body by boundary measurements. *SIAM Review*, 46(3):477–498, 2004.

[3] Giovanni Alessandrini, Antonio Bilotta, Giovanni Formica, Antonino Morassi, Edi Rosset, and Emilio Turco. Numerical size estimates of inclusions in elastic bodies. *Inverse Problems*, 21(1):133–151, nov 2004.

[4] Grégoire Allaire, Frédéric de Gournay, François Jouve, and Anca-Maria Toader. Structural optimization using topological and shape sensitivity via a level set method. *Control and cybernetics*, 2005.

[5] Grégoire Allaire, François Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.*, 194(1):363–393, 2004.

[6] Carlos J. S. Alves and Tuong Ha-Duong. Inverse scattering for elastic plane cracks. *Inverse Problems*, 15:91, 1999.

[7] Hend B. Ameur, Martin Burger, and Benjamin Hackl. Level set methods for geometric inverse problems in linear elasticity. *Inverse Problems*, 20(3):673–696, 2004.

[8] Hend B. Ameur, Martin Burger, and Benjamin Hackl. Cavity identification in linear elasticity and thermoelasticity. *Mathematical Methods in the Applied Sciences*, 30(6), 2007.

[9] Habib Ammari. *Mathematical methods in elasticity imaging*. Princeton University Press, Oxford;Princeton;, 2015.

[10] Douglas N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.

[11] Ivo Babuška and John E. Osborn. Can a finite element method perform arbitrarily badly? *Math. Comp.*, 69(230):443–462, 2000.

[12] Robert Banasiak and Manuchehr Soleimani. Shape based reconstruction of experimental data in 3d electrical capacitance tomography. *NDT and E International*, 43(3):241–249, 2010.

[13] Martin P. Bendsøe. *Optimization of structural topology, shape, and material*. Springer-Verlag, Berlin, 1995.

[14] Elena Beretta, Stefano Micheletti, Simona Perotto, and Matteo Santacesaria. Reconstruction of a piecewise constant conductivity on a polygonal partition via shape optimization in EIT. *Journal of Computational Physics*, 353:264 – 280, 2018.

[15] Marc Bonnet and Andrei Constantinescu. Inverse problems in elasticity. *Inverse Problems*, 21(2):R1–R50, 2005.

[16] Susanne C. Brenner and L. Ridgway Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts Appl. Math.* Springer, New York, third edition, 2008.

[17] Martin Burger. Levenberg-marquardt level set methods for inverse obstacle problems. *Inverse Problems*, 20(1):259–282, 2004.

[18] Martin Burger and Stanley J. Osher. A survey on level set methods for inverse problems and optimal design. *European J. Appl. Math.*, 16(2):263–301, 2005.

[19] Tony F. Chan and Xue-Cheng Tai. Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients. *Journal of Computational Physics*, 193(1):40–66, 2004.

[20] Kuang-Hua Chang and Kyung K. Choi. A geometry-based parameterization method for shape design of elastic solids. *Mechanics of Structures and Machines*, 20(2):215–252, 1992.

[21] Kyung K. Choi and Kuang-Hua Chang. A study of design velocity field computation for shape optimal design. *Finite Elem. Anal. Des.*, 15(4):317–341, 1994.

[22] Yat Tin Chow, Kazufumi Ito, and Jun Zou. A direct sampling method for electrical impedance tomography. *Inverse Problems*, 30(9):095003, aug 2014.

[23] Eric T. Chung, Tony F. Chan, and Xue-Cheng Tai. Electrical impedance tomography using level set representation and total variational regularization. *Journal of Computational Physics*, 205(1):357–372, 2005.

[24] A. Constantinescu. On the identification of elastic moduli in plates. In M. Tanaka and G.S. Dulikravich, editors, *Inverse Problems in Engineering Mechanics*, pages 205 – 214. Elsevier Science Ltd, Oxford, 1998.

[25] Andrei Constantinescu. On the identification of elastic moduli from displacement-force boundary measurements. *Inverse Problems in Engineering*, 1(4):293–313, 1995.

[26] Marc Dambrine and Djalil Kateb. Conformal mapping and inverse conductivity problem with one measurement. *ESAIM: Control, Optimisation and Calculus of Variations*, 13(1):163–177, 2007.

[27] J.E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, volume 16 of *Classics Appl. Math.* SIAM, 1996.

[28] Jim Douglas, Jr. and Todd Dupont. Interior penalty procedures for elliptic and parabolic Galerkin methods. In *Computing methods in applied sciences (Second Internat. Sympos., Versailles, 1975)*, pages 207–216. Lecture Notes in Phys., Vol. 58. Springer, Berlin, 1976.

[29] E. E. Drakonaki, G. M. Allen, and D. J. Wilson. Ultrasound elastography for musculoskeletal applications. *The British journal of radiology*, 85(1019):1435–1445, 2012.

[30] Giuseppe Geymonat and Stéphane Pagano. Identification of mechanical properties by displacement field measurement: A variational approach. *Meccanica*, 38(5):535–545, 2003.

[31] Michael B. Giles and Niles A. Pierce. An introduction to the adjoint approach to design. *Flow Turbul. Combust.*, 65(3):393–415, 2000.

[32] Ruchi Guo and Tao Lin. A group of immersed finite element spaces for elliptic interface problems. *IMA J.Numer. Anal.*, 39:482–511, 2019.

[33] Ruchi Guo and Tao Lin. A higher degree immersed finite element method based on a cauchy extension. *SIAM J. Numer. Anal.*, 57(4):1545–1573, 2019.

[34] Ruchi Guo and Tao Lin. A trilinear immersed finite element method for solving elliptic interface problems. *arXiv:1905.10012*, 2019.

[35] Ruchi Guo, Tao Lin, and Yanping Lin. Approximation capabilities of immersed finite element spaces for elasticity interface problems. *Numerical Methods for Partial Differential Equations*, 35:1243–1268, 2019.

[36] Ruchi Guo, Tao Lin, and Yanping Lin. Error estimates for a partially penalized immersed finite element method for elasticity interface problems. *ESAIM Math. Model. Numer. Anal. (to appear)*, 2019.

[37] Ruchi Guo, Tao Lin, and Yanping Lin. A fixed mesh method with immersed finite elements for solving interface inverse problems. *J. Sci. Comput.*, 79:148–175, 2019.

[38] Eduard Harabetian and Stanley Osher. Regularization of ill-posed problems via the level set approach. *SIAM Journal on Applied Mathematics*, 58(6):1689–1706, 1998.

[39] Helmut Harbrecht and Johannes Tausch. On the numerical solution of a shape optimization problem for the heat equation. *SIAM J. Sci. Comput.*, 35(1):A.104–A121, 2013.

[40] J. Haslinger and R. A. E. Mäkinen. *Introduction to shape optimization: theory, approximation, and computation.* SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2003.

[41] Maatoug Hassine and Imen Kallel. Kohn–vogelius formulation and topological sensitivity analysis based method for solving geometric inverse problems. *Arab Journal of Mathematical Sciences*, 24(1):43 – 62, 2018.

[42] Jan Hegemann, Alejandro Cantarero, Casey L. Richardson, and Joseph M. Teran. An explicit update scheme for inverse parameter and interface estimation of piecewise constant coefficients in linear elliptic pdes. *SIAM J. Sci. Comput.*, 35(2), 2013.

[43] R. Hiptmair and A. Paganini. Shape optimization by pursuing diffeomorphisms. *Computational Methods in Applied Mathematics*, 15(3):291–305, 2015.

[44] V. Isakov and J. Powell. On the inverse conductivity problem with one measurement. *Inverse Problems*, 6:311, 1990.

[45] Kazufumi Ito, Bangti Jin, and Jun Zou. A direct sampling method to an inverse medium scattering problem. *Inverse Problems*, 28(2):025003, 2012.

[46] Kazufumi Ito, Karl Kunisch, and Zhilin Li. Level-set function approach to an inverse interface problem. *Inverse Problems*, 17:1225, 2001.

[47] B. Jadamba, A.A. Khan, and F. Raciti. On the inverse problem of identifying lamé coefficients in linear elasticity. *Computers & Mathematics with Applications*, 56(2):431 – 443, 2008.

[48] Gang-Won Jang and Yoon Y. Kim. Sensitivity analysis for fixed-grid shape optimization by using oblique boundary curve approximation. *Int J Solids Struct.*, 42(11):3591–3609, 2005.

[49] Bangti Jin, Yifeng Xu, and Jun Zou. A convergent adaptive finite element method for electrical impedance tomography. *IMA J. Numer. Anal.*, 37(1):1520–1550, 2017.

[50] Jin, Bangti and Maass, Peter. An analysis of electrical impedance tomography with applications to tikhonov regularization. *ESAIM: COCV*, 18(4):1027–1048, 2012.

[51] R. Kafafy, T. Lin, Y. Lin, and J. Wang. Three-dimensional immersed finite element methods for electric field simulation in composite materials. *Internat. J. Numer. Methods Engrg.*, 64(7):940–972, 2005.

[52] Hyeonbae Kang and Jin K. Seo. Inverse conductivity problem with one measurement: Uniqueness of balls in $\mathbb{R}^3$. *SIAM Journal on Applied Mathematics*, 59(5):1533–1539, 1999.

[53] Nam H. Kim and Youngmin Chang. Eulerian shape design sensitivity analysis and optimization with a fixed grid. *Comput. Methods Appl. Mech. Engrg.*, 194(30):3291–3314, 2005.

[54] Robert V. Kohn and Michael Vogelius. Relaxation of a variational method for impedance computed tomography. *Commun. Pure Appl. Anal.*, 40(6):745–777, 1987.

[55] Hae S. Lee, Cheon J. Park, and Hyun W. Park. Identification of geometric shapes and material properties of inclusions in two-dimensional finite bodies by boundary parameterization. *Comput. Methods Appl. Mech. Engrg.*, 181(1):1–20, 2000.

[56] Tao Lin, Yanping Lin, and Xu Zhang. Partially penalized immersed finite element methods for elliptic interface problems. *SIAM J. Numer. Anal.*, 53(2):1121–1144, 2015.

[57] Tao Lin, Dongwoo Sheen, and Xu Zhang. A locking-free immersed finite element method for planar elasticity interface problems. *J. Comput. Phys.*, 247:228–247, 2013.

[58] Tao Lin and Xu Zhang. Linear and bilinear immersed finite elements for planar elasticity interface problems. *J. Comput. Appl. Math.*, 236(18):4681–4699, 2012.

[59] Joyce R. McLaughlin, Ning Zhang, and Armando Manduca. Calculating tissue shear modulus and pressure by 2d log-elastographic methods. *Inverse Problems*, 26, 2010.

[60] Ahmad R. Najafi, Masoud Safdari, Daniel A. Tortorelli, and Philippe H. Geubelle. A gradient-based shape optimization scheme using an interface-enriched generalized fem. *Comput. Methods Appl. Mech. Engrg.*, 296:1–17, 2015.

[61] S. S. Nanthakumar, T. Lahmer, and T. Rabczuk. Detection of flaws in piezoelectric structures using extended fem. *International Journal for Numerical Methods in Engineering*, 96(6):373–389, 2013.

[62] J. Nocedal and S. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer, second edition, 2006.

[63] Antonio André Novotny, Alfredo Canelas, and Antoine Laurain. A non-iterative method for the inverse potential problem based on the topological derivative. In *Technical Report for Mini-Workshop: Geometries, Shapes and Topologies in PDE-based Applications, 57/2012, edited by Michael Hintermüller and Günter Leugering and Jan Sokołowski*, pages 3383–3387, Mathematisches Forschungsinstitut Oberwolfach, Oberwolfach, Germany, 2012.

[64] Jonathan Ophir, S. K. Alam, Brian S. Garra, Faouzi Kallel, Elisa E. Konofagou, Thomas Krouskop, Christopher R. B. Merritt, Raffaella Righetti, Remi Souchon, Seshadri Srinivasan, and Tomy Varghese. Elastography: Imaging the elastic properties of soft tissues with ultrasound. *Journal of Medical Ultrasonics*, 29(4):155–171, 2002.

[65] A. Paganini. *Approximate Shape Gradients for Interface Problems*, pages 217–227. Springer International Publishing, Cham, 2015.

[66] A. Paganini, F. Wechsung, and P. Farrell. Higher-order moving mesh methods for pde-constrained shape optimization. *SIAM Journal on Scientific Computing*, 40(4):A2356–A2382, 2018.

[67] Mauro Perego, Alessandro Veneziani, and Christian Vergara. A variational approach for estimating the compliance of the cardiovascular tissue: An inverse fluid-structure interaction problem. *SIAM J. Sci. Comput.*, 33(3):1181–1211, 2011.

[68] Daniel Rabinovich, Dan Givoli, and Shmuel Vigdergauz. Xfem-based crack detection scheme using a genetic algorithm. *International Journal for Numerical Methods in Engineering*, 71(9):1051–1080, 2007.

[69] J. J. Ródenas, F. J. Fuenmayor, and J. E. Tarancón. A numerical methodology to assess the quality of the design velocity field computation methods in shape sensitivity analysis. *Internat. J. Numer. Methods Engrg.*, 59(13):1725–1747, 2004.

[70] Fadil Santosa. A level-set approach for inverse problems involving obstacles. *ESAIM: Control, Optimisation and Calculus of Variations*, 1:17–33, 1996.

[71] D. S. Schnur and Nicholas Zabaras. An inverse method for determining elastic material properties and a material interface. *Internat. J. Numer. Methods Engrg.*, 33(10):2039–2057, 1992.

[72] Jan Sokolowski and Jean-Paul Zolésio. *Introduction to shape optimization : shape sensitivity analysis*, volume 16 of *Springer Series in Computational Mathematics*. Springer, 1992.

[73] M. Soleimani, W. R. B. Lionheart, and O. Dorn. Level set reconstruction of conductivity and permittivity from boundary electrical measurements using experimental data. *Inverse Problems in Science and Engineering*, 14(2):193–210, 2006.

[74] Gunther Uhlmann, Jenn-Nan Wang, and Chin-Tien Wu. Reconstruction of inclusions in an elastic body. *Journal de Mathématiques Pures et Appliquées*, 91(6):569 – 582, 2009.

[75] Haim Waisman, Eleni Chatzi, and Andrew W. Smyth. Detection and quantification of flaws in structures by the extended finite element method and genetic algorithms. *International Journal for Numerical Methods in Engineering*, 2009.