Journal of Scientific Computing (2019) 79:148–175
https://doi.org/10.1007/s10915-018-0847-y

A Fixed Mesh Method with Immersed Finite Elements for Solving Interface Inverse Problems

Ruchi Guo[1] · **Tao Lin**[1] · **Yanping Lin**[2]

Received: 11 July 2018 / Revised: 21 September 2018 / Accepted: 3 October 2018 /
Published online: 13 October 2018

Abstract
We present a new fixed mesh method for solving a class of interface inverse problems for the typical elliptic interface problems. These interface inverse problems are formulated as shape optimization problems. By an immersed finite element (IFE) method, both the governing partial differential equations and the objective functional for an interface inverse problem are discretized optimally regardless of the location of the interface in a chosen mesh, and the shape optimization for recovering the interface is reduced to a constrained optimization problem. The formula for the gradient of the objective function in this constrained optimization is derived and this formula can be implemented efficiently in the IFE framework. As demonstrated by three representative applications, the proposed IFE method can be employed to solve a spectrum of interface inverse problems efficiently.

Keywords Inverse problems · Interface problems · Shape optimization · Discontinuous coefficients · Immersed finite element methods

1 Introduction

In this article, we present a numerical method for solving a class of interface inverse problems with a fixed mesh by an immersed finite element (IFE) method. Without loss of generality, let Ω be a domain separated by an interface Γ into two subdomains Ω^- and Ω^+ each occupied by a different material represented by a piecewise constant function β discontinuous across Γ. We consider a group of K forward interface boundary value problems posed on the domain Ω for the typical second order elliptic equation:

This research was partially supported by Polyu G-UA7V and HKSAR B-Q40W.

✉ Tao Lin
tlin@vt.edu

Ruchi Guo
ruchi91@vt.edu

Yanping Lin
yanping.lin@polyu.edu.hk

[1] Department of Mathematics, Virginia Tech, Blacksburg, VA 24061, USA

[2] Department of Applied Mathematics, Hong Kong Polytechnic University, Kowloon, Hong Kong, China

$$\begin{aligned} &-\nabla\cdot(\beta\nabla u^k) = f^k, \ \text{ in } \Omega^-\cup\Omega^+,\\ &u^k = g_D^k, \ \text{ on } \partial\Omega_D^k\subseteq\partial\Omega, \ \ \tfrac{\partial u^k}{\partial \mathbf{n}} = g_N^k, \ \text{ on } \partial\Omega_N^k\subseteq\partial\Omega, \end{aligned} \quad \text{for } k=1,2,\ldots,K, \tag{1.1}$$

where $\overline{\partial\Omega_N^k}\cup\overline{\partial\Omega_D^k}=\partial\Omega$ and $\mathbf{n}$ is the outward normal vector of $\partial\Omega$, together with the jump conditions on the interface Γ:

$$\begin{aligned} &[u^k]|_\Gamma := u^{k,+}|_\Gamma - u^{k,-}|_\Gamma = 0,\\ &[\beta\nabla u^k\cdot\mathbf{n}]|_\Gamma := \left(\beta^+\nabla u^{k,+}-\beta^-\nabla u^{k,-}\right)|_\Gamma\cdot\mathbf{n}=0, \ \ \mathbf{n} \text{ is the normal of } \Gamma, \end{aligned} \quad k=1,2,\ldots,K, \tag{1.2}$$

$$\text{in which } \ u^{k,s}=u^k|_{\Omega^s}, \ \beta(X)=\beta^s \ \text{ for } \ X\in\Omega^s, \ s=-,+. \tag{1.3}$$

An important inverse problem related to the typical second order elliptic equation is to identify the coefficient β where one needs to either identify the physical properties of materials, i.e., the values (the *parameter estimation* problem) and/or detect the location and shape of inclusions/interfaces (the *inverse geometric* problem) using the data measured for $u^k,\ 1\le k\le K$ on a subset of the domain or on a subset of the boundary $\partial\Omega$ [22,37,42]. This type of inverse problems arises from many applications in engineering and sciences, such as the electrical impedance tomography (EIT) [14,39] and groundwater or oil reservoir simulation [25,77]. In the former case, u^ks and β represent the electrical potential and the conductivity, respectively, whereas in the latter case u^ks and β are the piezometric head and transmissivity, respectively. Similar inverse problems related to other partial differential equations can be found in [38,44] for medical imaging problems, [48,71] for elastography, and references therein. It is well known that these inverse problems are usually ill-posed especially when the available data is rather limited. Numerical methods based on the output-least-squares formulation are commonly used to handle these types of inverse problems, see [17,20,22,37,41] and references therein.

In many engineering and science applications, the values of material properties or parameters are known or chosen such as the elastic properties of tissue and bone in medical problems [58,66] and the electrical properties in EIT problems [3,10] to mention just a couple of applications. Thus, the focus of this article is to develop an efficient numerical method based on a fixed mesh for the inverse geometric problem related to the forward interface problem described by (1.1) and (1.2) in which we assume that the material values $\beta^s=\beta|_{\Omega^s}, s=-,+$ are known and we need to use the given measurements about $u^k,\ 1\le k\le K$ to recover the location and geometry of the material interface Γ.

The shape optimization method [34,64] is widely used for inverse geometric problems by which we seek for the interface Γ^* from an optimization problem:

$$\Gamma^* = \text{argmin}\ \mathcal{J}(\Gamma), \tag{1.4}$$

where

$$\mathcal{J}(\Gamma) = \int_{\Omega_0} J(u^1(\Gamma), u^2(\Gamma),\ldots,u^K(\Gamma); X, \Gamma)dX, \tag{1.5}$$

and $u^k(\Gamma)$s are the solutions to the forward interface problems (1.1) and (1.2), but $\Omega_0\subseteq\Omega$ and $J(u^1(\Gamma), u^2(\Gamma),\ldots,u^K(\Gamma); X, \Gamma)$ are application dependent. A few specific formulations of J are given in Sect. 4 for a chosen group of representative applications. We note that the shape optimization approach has been applied to numerous applications, see for examples [10,13,40,49,59].

The movement of the structure, boundary or interface is a critical issue in a shape optimization process challenging a solver chosen for the related forward problems. Traditional

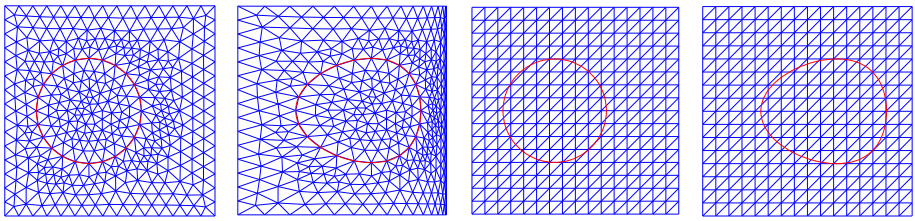

Fig. 1 The body-fitting and interface independent mesh. **a** The initial body fitting mesh. **b** The body fitting mesh after movement. **c** The interface independent mesh. **d** The interface independent mesh (Color figure online)

finite element methods can be used to obtain accurate solutions to the forward interface problems provided that they use a body fitting (or interface conforming) mesh [4,16]; otherwise, their performance may not be satisfactory [7,19]. The shape optimization methods based on body fitting mesh are referred as the Lagrangian approach [21] which, however, has a few drawbacks. The first concerns the mesh updating process from one iteration to the next in the optimization. As the geometry changes, to guarantee the accuracy, the mesh used by the chosen solver for the forward problem needs to be updated to fit the new shape of the boundary or interface [11,73], which not only consumes time but also generates unsatisfactory meshes in many situations, see the illustrations in Fig. 1 where the two plots on the left demonstrate an inappropriate mesh movement strategy leading to a mesh with less desirable qualities, especially near the right boundary.

The sensitivity analysis in a shape optimization is about the derivatives of an objective function with respect to the design variables, i.e., the parameters describing the geometry of the domain [11,34], and this leads to the gradient of the objective function which is a necessary ingredient in common numerical optimization algorithms such as descent direction methods and trust region methods [23,62]. A velocity field defined as the derivatives of node coordinates with respect to the design variables [21,68] is usually employed in the sensitivity analysis. Several approaches for computing the velocity in a Lagrangian framework are summarized in [21], which either require computations to be carried out over the whole domain or need some special numerical methods for generating the velocity field approximately.

Alternatively, the Eulerian approaches based on fixed meshes have been widely used in the shape optimization algorithms [45,46], which allow the material interface/boundary to cut elements as illustrated in the two plots on the right in Fig. 1. For example, the extended finite element methods (XFEM) based on a fixed mesh are used to solve optimal design problems [57,60,72,79] and the inverse geometric problems related to crack detection [61,67,75]; the immersed interface methods (IIM) based on a Cartesian mesh is employed to solve a cavity (rigid inclusion) detection problem in [42]. Various techniques for improving the accuracy of the evaluation of either the stiffness matrix or sensitivity on those boundary/interface elements in Eulerian methods have been discussed in [5,24,43].

The goal of this article is to develop a fixed mesh method based on the partially penalized immersed finite element (PPIFE) method [54] for solving the inverse geometric problems/optimal design problems described by (1.1)–(1.5). A key motivation is that IFE methods can solve interface forward problems on interface independent meshes optimally with respect to the degree of the involved polynomial spaces, and, as explained in Sect. 3, this naturally leads to optimal discretizations of cost functionals in the shape optimization for many common interface inverse problems. In an IFE method, the interface shape/location and the jump

conditions across the interface are intrinsically utilized in the local IFE shape functions on interface elements, while the standard local finite element spaces are used on non-interface elements. We refer readers to IFE spaces constructed with linear polynomials [50,51], with bilinear polynomials [35,53], with rotated-Q_1 polynomials [32,80] and high-order polynomials [2]. Applications of IFE methods to other types of equations or jump conditions can be found in [1,8,18,36,52,55,78].

The proposed IFE method for the inverse geometric problems has additional advantages. Because it is based on an interface independent fixed mesh in the shape optimization process, the issues caused by the mesh regeneration/movement, the mesh distortion caused by large geometry changes, as well as some practical and theoretical issues for the construction of the velocity field [21] are circumvented, see the plots (c) and (d) in Fig. 1 for an illustration. When the numerical interface curve Γ is expressed as a parametric curve whose control points are the design variables for the shape optimization, the fixed mesh used in the proposed IFE method allows us to develop velocity fields and shape derivatives of IFE shape functions that are advantageous for efficient implementation because they both vanish outside the interface elements whose total number is of the order $O(h^{-1})$ on a shape regular mesh compared to the total number of elements in the order of $O(h^{-2})$. In addition, the explicit formulas for the velocity fields and shape derivatives in the proposed IFE method can be implemented precisely within the IFE framework without any further approximation procedures, and they are essential ingredients for the formula derived in this article to efficiently compute the gradient of the objective function with respect to the design variables. We highlight that almost all the computations in the sensitivity calculation in the proposed IFE method are concentrated only on the interface elements. These benefits together with the fact that no need to remesh again and again in the shape optimization process suggest a strong potential of the proposed IFE method.

This article is organized as follows. The next section recalls the linear IFE space and the related PPIFE scheme for the interface forward problems. Section 3 presents the shape optimization algorithm based on the IFE discretization on a fixed mesh of Ω for the inverse geometric problem described by (1.1)–(1.5) and the computation procedure for its sensitivity. In Sect. 4, we demonstrate the strength and versatility of the proposed IFE method by applying it to three representative interface inverse problems. Numerical examples presented in this section also serve as hints/suggestions about how to implement the proposed IFE method efficiently.

2 An IFE Method for the Interface Forward Problems

In this section we recall the linear IFE method [31,50,51] for the discretizatoin of the interface forward problem described by (1.1) and (1.2) with an interface independent mesh. The following notations will be used throughout this article. We let $\Gamma(t, \boldsymbol{\alpha}), t \in [0, 1]$ be a parametrization of the interface Γ with design variables as entries in the vector $\boldsymbol{\alpha} = (\alpha_j)_{j\in\mathcal{D}}$ where $\mathcal{D}$ is the index set of the chosen design variables. For example, when $\Gamma(t, \boldsymbol{\alpha})$ is a cubic spline, $\boldsymbol{\alpha}$ is the vector of all the coordinates of control points [27]. Let $\mathcal{T}_h$ be an interface independent triangular mesh of the domain Ω. An element $T \in \mathcal{T}_h$ will be called an interface element if its interior intersects the interface $\Gamma(t, \boldsymbol{\alpha})$; otherwise we call it a non-interface element. Let $\mathcal{T}_h^{int}$ ($\mathcal{E}_h^{int}$) and $\mathcal{T}_h^{n}$ ($\mathcal{E}_h^{n}$) be the sets of interface and non-interface elements (edges), respectively. Denote the set of the interior interface edges by $\mathring{\mathcal{E}}_h^{int}$. And let

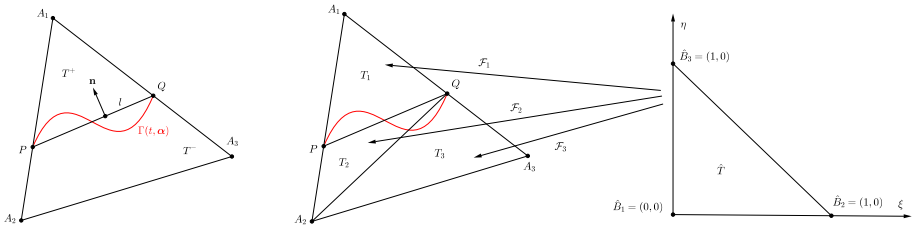

Fig. 2 An interface element and its partitions (Color figure online)

$\mathcal{N}_h = \{X_1, X_2, \ldots, X_{|\mathcal{N}_h|}\}$ and $\mathring{\mathcal{N}}_h$ be the sets of all the nodes in the mesh and the interior nodes, respectively.

For each element $T = \triangle A_1A_2A_3 \in \mathcal{T}_h$, we let $\mathcal{I} = \{1, 2, 3\}$ and let $\psi^{non}_{p,T}$, $p = 1, 2, 3$ be the standard linear shape functions on T [12] such that $\psi^{non}_{p,T}(A_q) = \delta_{pq}$, $p, q \in \mathcal{I}$. The local IFE space on each $T \in \mathcal{T}^n_h$ is

$$S_h(T) = \text{Span}\{\psi^{non}_{p,T},\ p \in \mathcal{I}\} = \mathbb{P}_1. \tag{2.1}$$

On an interface element $T = \triangle A_1A_2A_3 \in \mathcal{T}^{int}_h$, we let $P = (x_P, y_P)^T$ and $Q = (x_Q, y_Q)^T$ be the two interface-mesh intersection points, and let l be the line connecting P and Q. The normal vector for the line l is $\bar{\mathbf{n}} = \frac{1}{||P-Q||}(y_P - y_Q, -(x_P - x_Q))^T$ and the equation for the line l is $L(X) = 0$ with $L(X) = \bar{\mathbf{n}} \cdot (X - P)$. The line l cuts the element into two sub-elements $\overline{T}^+$ and $\overline{T}^-$, see the sketch on the left in Fig. 2, and we use them to introduce another two index sets $\mathcal{I}^- = \{p : A_p \in \overline{T}^-\}$ and $\mathcal{I}^+ = \{p : A_p \in \overline{T}^+\}$.

According to [31], with $\mu = \beta^+/\beta^- - 1$, the linear IFE function constructed according to the interface jump condition (1.2) and nodal values $\mathbf{v} = (v_1, v_2, v_3)$ has the following formula

$$\psi^{int}_T(X) = \begin{cases} \psi^{int,-}_T(X) = \psi^{int,+}_T(X) + c_0 L(X) & \text{if } X \in \overline{T}^-, \\ \psi^{int,+}_T(X) = \sum_{p\in\mathcal{I}^-} c_p \psi^{non}_{p,T}(X) + \sum_{p\in\mathcal{I}^+} v_p \psi^{non}_{p,T}(X) & \text{if } X \in \overline{T}^+, \end{cases} \tag{2.2}$$

$$\text{with} \quad c_0 = \mu \left(\sum_{p\in\mathcal{I}^-} c_p \nabla\psi^{non}_{p,T} \cdot \bar{\mathbf{n}} + \sum_{p\in\mathcal{I}^+} v_p \nabla\psi^{non}_{p,T} \cdot \bar{\mathbf{n}} \right),$$

$$\mathbf{c} := (c_p)_{p\in\mathcal{I}^-} = \mathbf{b} - \mu \frac{(\boldsymbol{\gamma}^T\mathbf{b})\boldsymbol{\delta}}{1 + \mu\boldsymbol{\gamma}^T\boldsymbol{\delta}}, \tag{2.3}$$

$$\text{in which} \quad \boldsymbol{\gamma} = \left(\nabla\psi^{non}_{p,T} \cdot \bar{\mathbf{n}}\right)_{p\in\mathcal{I}^-}, \quad \boldsymbol{\delta} = \left(L(A_p)\right)_{p\in\mathcal{I}^-},$$

$$\mathbf{b} = \left(v_p - \mu L(A_p) \sum_{q\in\mathcal{I}^+} \nabla\psi^{non}_{q,T} \cdot \bar{\mathbf{n}}\, v_q \right)_{p\in\mathcal{I}^-}. \tag{2.4}$$

Then, using $\mathbf{v} = \mathbf{e}_p$, $p \in \mathcal{I}$, the standard basis vectors of $\mathbb{R}^3$ in (2.2)–(2.4), we obtain the IFE shape functions $\psi^{int}_{p,T}(X)$, $p \in \mathcal{I}$ satisfying $\psi^{int}_{p,T}(A_q) = \delta_{pq}$ for $p, q \in \mathcal{I}$. The local IFE space on each $T \in \mathcal{T}^{int}_h$ is defined as

$$S_h(T) = \text{Span}\{\psi^{int}_{p,T} \ :\ p \in \mathcal{I}\}. \tag{2.5}$$

Local IFE spaces defined on all elements of $\mathcal{T}_h$ are then used to define the global IFE space as follows

$$S_h(\Omega) = \big\{ v \in L^2(\Omega) : v|_T \in S_h(T);\, v|_{T_1}(A) = v|_{T_2}(A), \\ \forall A \in \mathcal{N}_h, \forall\, T_1, T_2 \in \mathcal{T}_h \text{ with } A \in T_1 \cap T_2 \big\}. \tag{2.6}$$

With this IFE space and its associated space $S_h^0(\Omega) = \{v \in S_h(\Omega) \,:\, v(X) = 0,\ \forall X \in \mathcal{N}_h \cap \partial\Omega_D\}$, the interface forward problem (1.1) and (1.2) can be disretized by the symmetric PPIFE (SPPIFE) method [54] as follows: find $u_h^k \in S_h(\Omega)$, $k = 1, 2, \ldots, K$ such that

$$a_h(u_h^k, v_h) = L_f^k(v_h),\ \ \forall v_h \in S_h^0(\Omega),\ \ u_h^k(X) = g_D^k(X),\ \ \forall X \in \mathcal{N}_h \cap \partial\Omega_D^k, \tag{2.7}$$

where the bilinear form $a_h(\cdot,\cdot)$ and linear functional $L_f^k(\cdot)$ are given by

$$\begin{aligned} a_h(u_h, v_h) = &\sum_{T\in\mathcal{T}_h} \int_T \beta \nabla u_h \cdot \nabla v_h dX - \sum_{e\in\mathcal{E}_h^{int}\setminus\partial\Omega_N^k} \int_e \{\beta\nabla u_h\}_e \cdot [v_h]_e ds \\ &- \sum_{e\in\mathcal{E}_h^{int}\setminus\partial\Omega_N^k} \int_e \{\beta\nabla v_h\}_e \cdot [u_h]_e ds \\ &+ \sum_{e\in\mathcal{E}_h^{int}} \frac{\sigma_e^0}{|e|} \int_e [u_h]_e \cdot [v_h]_e ds,\ \ \forall u_h, v_h \in S_h(\Omega), \end{aligned} \tag{2.8}$$

$$\begin{aligned} L_f^k(v_h) = &\int_\Omega f^k v_h dX + \int_{\partial\Omega_N^k} g_N^k v_h ds - \sum_{e\in\mathcal{E}_h^{int}\cap\partial\Omega_D^k} \int_e \beta g_D^k \nabla v_h \cdot \mathbf{n}_e ds \\ &+ \sum_{e\in\mathcal{E}_h^{int}\cap\partial\Omega_D^k} \frac{\sigma_e^0}{|e|} \int_e g_D^k v_h ds,\ \ \forall v_h \in S_h(\Omega). \end{aligned} \tag{2.9}$$

In the bilinear form $a_h(\cdot,\cdot)$, the operators $[\cdot]_e$ and $\{\cdot\}_e$ on each interior interface edge $e \in \mathring{\mathcal{E}}_h^{int}$ shared by elements T_1 and T_2 are such that $[v]_e = (v|_{T_1}\mathbf{n}_e^1 + v|_{T_2}\mathbf{n}_e^2)$, and $\{\beta\nabla v\}_e = \frac{1}{2}(\beta\nabla v|_{T_1} + \beta\nabla v|_{T_2})$, $\forall v \in S_h(\Omega)$, where the normal vector $\mathbf{n}_e^1 = -\mathbf{n}_e^2$ is from T^1 to T^2. For $e \in \mathcal{E}_h^{int} \cap \partial\Omega$, we define the operators $[\cdot]_e$ and $\{\cdot\}_e$ as $[v]_e = v|_T\mathbf{n}_e$, $\{\beta\nabla v\}_e = \beta\nabla v|_T$, $\forall v \in S_h(\Omega)$, where T is the element that contains e and $\mathbf{n}_e$ is the outward normal vector to $\partial\Omega$. In our applications, we choose $\sigma_e^0 = 10\max\{\beta^-, \beta^+\}$. It has been shown [54] that the PPIFE solutions u_h^k from (2.7) approximate the true solutions u^k, $1 \leqslant k \leqslant K$, with an optimal accuracy in both the L^2 and H^1 norms with respect to the involved polynomials regardless of the interface location and shape, i.e.,

$$\|u_h^k - u^k\|_{L^2(\Omega)} + h|u_h^k - u^k|_{H^1(\Omega)} \leqslant Ch^2\|u^k\|_{H^2(\Omega)}. \tag{2.10}$$

In order to describe how this IFE discretization is used to solve interface inverse problems, we put the SPPIFE method described by (2.7)–(2.9) in the matrix form. We assume that $S_h(\Omega) = \text{Span}\{\phi_i(X) \mid X_i \in \mathcal{N}_h\}$ in which $\phi_i(X)$ is the global IFE basis function associated with the node $X_i \in \mathcal{N}_h$. When the k-th ($1 \leq k \leq K$) interface forward problem has a mixed boundary condition, we let $\mathcal{N}_h^M = \{X_i \mid X_i \in \mathring{\mathcal{N}}_h \cup \partial\Omega_N\}$ such that we can denote the SPPIFE solution $u_h^k(X) \in S_h(\Omega)$ determined by (2.7)–(2.9) as follows:

$$u_h^k(X) = \sum_{i=1}^{|\mathcal{N}_h^M|} u_i^k \phi_i(X) + \sum_{i=|\mathcal{N}_h^M|+1}^{|\mathcal{N}_h|} g_D^k(X_i)\phi_i(X), \tag{2.11}$$

where, without loss of generality, we have assumed that nodes in $\mathcal{N}_h^M$ are ordered first. Here and from now on, the superscript M means that the boundary condition in the k-th interface forward problem is of a mixed type. The stiffness matrix $\tilde{\mathbf{A}} = (a_{i,j})_{i,j=1}^{|\mathcal{N}_h|}$ associated with the bilinear form defined in (2.8) can be assembled from the following local matrices on elements and edges of $\mathcal{T}_h$:

$$\mathbf{K}_T = \left(\int_T \beta \nabla \psi_{p,T} \cdot \nabla \psi_{q,T} dX \right)_{p,q \in \mathcal{I}}, \quad \forall T \in \mathcal{T}_h, \tag{2.12a}$$

$$\mathbf{E}_e^{r_1 r_2} = \left(\int_e \beta \nabla \psi_{p,T^{r_1}} \cdot (\psi_{q,T^{r_2}} \mathbf{n}_e^{r_2}) ds \right)_{p,q \in \mathcal{I}}, \quad \forall e \in \mathcal{E}_h^{int}, \tag{2.12b}$$

$$\mathbf{G}_e^{r_1 r_2} = \left(\frac{\sigma_e^0}{|e|} \int_e (\psi_{p,T^{r_1}} \mathbf{n}_e^{r_1}) \cdot (\psi_{q,T^{r_2}} \mathbf{n}_e^{r_2}) ds \right)_{p,q \in \mathcal{I}}, \quad \forall e \in \mathcal{E}_h^{int}, \tag{2.12c}$$

where the index $r_1, r_2 = 1, 2$ and the edge $e \in \mathring{\mathcal{E}}_h^{int}$ is shared by the two neighbor elements T^1 and T^2. But in the case $e \in \mathcal{E}_h^{int} \cap \partial\Omega$, we let $r_1 = r_2 = 0$, $\mathbf{n}_e^0 = \mathbf{n}_e$ is the outward normal vector and $T^0 = T$ is the element that contains e. Let $\tilde{\mathbf{A}}_b^{M,k} = (a_{b,i}^k)_{i=1}^{|\mathcal{N}_h|} = \tilde{\mathbf{A}} \left[\mathbf{0} \ \mathbf{g}_D^k \right]^T$, where $\mathbf{0}$ is the $|\mathcal{N}_h^M|$-dimensional zero vector and $\mathbf{g}_D^k = [g_D^k(X_{|\mathcal{N}_h^M|+1}), \ldots, g_D^k(X_{|\mathcal{N}_h|})]^T$. Similarly, the load vector $\tilde{\mathbf{F}}^k = (f_i^k)_{i=1}^{|\mathcal{N}_h|}$ associated with the linear form defined in (2.9) can be assembled from the following vectors:

$$\mathbf{F}_T^k = \left(\int_T f^k \psi_{p,T} dX \right)_{p \in \mathcal{I}}, \quad \forall T \in \mathcal{T}_h, \tag{2.13a}$$

$$\mathbf{B}_e^k = \left(\int_e \beta g_D^k \nabla \psi_{p,T} \cdot \mathbf{n}_e ds \right)_{p \in \mathcal{I}}, \quad \mathbf{C}_e^k = \frac{\sigma_e^0}{|e|} \left(\int_e \beta g_D^k \psi_{p,T} ds \right)_{p \in \mathcal{I}}, \quad \forall e \in \mathcal{E}_h^{int} \cap \partial\Omega_D, \tag{2.13b}$$

$$\mathbf{N}_e^k = \left(\int_e g_N^k \psi_{p,T} ds \right)_{p \in \mathcal{I}}, \quad \forall e \in \mathcal{E}_h^{int} \cap \partial\Omega_N. \tag{2.13c}$$

Letting $\mathbf{u}_h^{M,k} = [u_1^k, u_2^k, \ldots, u_{|\mathcal{N}_h^M|}^k]^T$, we can see that the unknown vector $\mathbf{u}_h^{M,k}$ of the SPPIFE solution $u_h^k(X)$ described by (2.11) is determined by the following linear system:

$$\mathbf{A}^{M,k} \mathbf{u}_h^{M,k} = \mathbf{F}^{M,k}, \tag{2.14}$$

where $\mathbf{A}^{M,k} = (a_{i,j})_{i,j=1}^{|\mathcal{N}_h^M|}$, $\mathbf{F}^{M,k} = (f_i^k)_{i=1}^{|\mathcal{N}_h^M|} - (a_{b,i}^k)_{i=1}^{|\mathcal{N}_h^M|}$.

When the k-th ($1 \le k \le K$) interface forward problem has a Neumann boundary condition such that $\partial\Omega_N^k = \partial\Omega$, we know that $|\mathcal{N}_h^M| = |\mathcal{N}_h|$, $u_h^k(X)$ given in (2.11) does not have the second term and the related load vector $\tilde{\mathbf{F}}^k = (f_i^k)_{i=1}^{|\mathcal{N}_h|}$ is assembled by the local vectors only in (2.13a) and (2.13c). Since the solution to the interface problem is not unique, as a common practice, the normalization condition $\int_\Omega u^k dX = u_0^k$ is imposed such that the SPPIFE solution $u_h^k(X)$ described by (2.11) is determined by the following linear system:

$$\mathbf{A}^{N,k} \mathbf{u}_h^{N,k} = \mathbf{F}^{N,k}, \quad \text{with } \mathbf{A}^{N,k} = \begin{bmatrix} \tilde{\mathbf{A}} & \mathbf{R} \\ \mathbf{R}^T & 0 \end{bmatrix}, \quad \begin{cases} \mathbf{u}_h^{N,k} = [u_1^k, u_2^k, \ldots, u_{|\mathcal{N}_h|}^k, \lambda]^T, \\ \mathbf{F}^{N,k} = [f_1^k, f_2^k, \ldots, f_{|\mathcal{N}_h|}^k, u_0^k]^T, \end{cases} \tag{2.15}$$

where the superscript N refers to the pure Neuman boundary condition, λ is the Lagrange multiplier, and $\mathbf{R}$ is the vector assembled with the following local vector constructed on each element:

$$\mathbf{R}_T = \left(\int_T \psi_{p,T} dX \right)_{p \in \mathcal{I}}, \quad \forall T \in \mathcal{T}_h. \tag{2.16}$$

In summary, according to (2.14) and (2.15), the SPPIFE discretization for the K interface forward problems described in (1.1) and (1.2) can be written in the following unified matrix form:

$$\begin{aligned} &\mathbf{A}^k \mathbf{u}_h^k = \mathbf{F}^k, \\ &\mathbf{u}_h^k = \begin{cases} \mathbf{u}_h^{M,k} \\ \mathbf{u}_h^{N,k} \end{cases} \quad \mathbf{A}^k = \begin{cases} \mathbf{A}^{M,k} \\ \mathbf{A}^{N,k} \end{cases} \quad \mathbf{F}^k = \begin{cases} \mathbf{F}^{M,k} & \text{for a mixed boundary condition,} \\ \mathbf{F}^{N,k} & \text{for a Neumann boundary condition.} \end{cases} \end{aligned} \tag{2.17}$$

We note that the matrices $\mathbf{A}^k$s in (2.17) are symmetric positive definite, and their size and algebraic structure remain the same as the interface $\Gamma(t, \boldsymbol{\alpha}), t \in [0, 1]$ evolves in a fixed mesh when the design variable $\boldsymbol{\alpha}$ varies.

3 An IFE Method for the Interface Inverse Problem

We now discuss the discretization of the inverse geometric problem (1.4) subject to the governing equations (1.1) and (1.2) by the SPPIFE method on a fixed mesh. By the discussion above, we note that the local matrices (2.12a)–(2.12b), local vectors (2.13a)–(2.13c), and (2.16) are all influenced by the shape variation. Hence, we write the matrix $\mathbf{A}^k$ and vector $\mathbf{F}^k$ in the SPPIFE equation (2.17) as $\mathbf{A}^k = \mathbf{A}^k(\boldsymbol{\alpha})$, $\mathbf{F}^k = \mathbf{F}^k(\boldsymbol{\alpha})$, $k = 1, 2, \ldots, K$, which further imply the solution $\mathbf{u}_h^k$ to the IFE equation (2.17) depends on $\boldsymbol{\alpha}$ and so we will denote it as $\mathbf{u}_h^k(\boldsymbol{\alpha})$ from now on. Also, in a shape optimization problem, the spacial variable X is considered as a function of $\boldsymbol{\alpha}$. Therefore, the IFE solution u_h^k to the k-th ($1 \le k \le K$) governing interface forward problem in the form of (2.11) depends on $\boldsymbol{\alpha}$ through the IFE solution vector $\mathbf{u}_h^k(\boldsymbol{\alpha})$, the spacial variable $X(\boldsymbol{\alpha})$, and the IFE basis functions, then we can denote it as follows

$$\begin{aligned} u_h^k &= u_h^k(\boldsymbol{\alpha}) = u_h^k(\mathbf{u}_h^k(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\ &= \sum_{i=1}^{|\mathcal{N}_h^M|} u_i^k(\boldsymbol{\alpha}) \phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) + \sum_{i=|\mathcal{N}_h^M|+1}^{|\mathcal{N}_h|} g_D^k(X_i) \phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \end{aligned} \tag{3.1}$$

where the second variable in $\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ emphasizes the fact that $\boldsymbol{\alpha}$ also effects the IFE solution u_h^k through the coefficients c_p, c_0 and the function $L(X)$ in the IFE shape functions by the formulas (2.3)–(2.4).

The IFE solutions $u_h^k(X) \approx u^k(X)$, $1 \le k \le K$, spacial-design-variable mapping $X = X(\boldsymbol{\alpha})$, and the interface parameterization $\Gamma(\cdot, \boldsymbol{\alpha})$ together naturally suggest the following discretization of the integrand in the objective functional defined by (1.5):

$$J(u^1(\boldsymbol{\alpha}), u^2(\boldsymbol{\alpha}), \ldots, u^K(\boldsymbol{\alpha}); X, \Gamma(\cdot, \boldsymbol{\alpha})) \approx J(u_h^1(\boldsymbol{\alpha}), u_h^2(\boldsymbol{\alpha}), \ldots, u_h^K(\boldsymbol{\alpha}); X(\boldsymbol{\alpha}), \Gamma(\cdot, \boldsymbol{\alpha})). \tag{3.2}$$

Following the explanations similar to those in the previous paragraph, the design variable $\boldsymbol{\alpha}$ can influence the approximated integrand $J(u_h^1(\boldsymbol{\alpha}), u_h^2(\boldsymbol{\alpha}), \ldots, u_h^K(\boldsymbol{\alpha}); X(\boldsymbol{\alpha}), \Gamma(\cdot, \boldsymbol{\alpha}))$ not only through $\mathbf{u}_h^k(\boldsymbol{\alpha})$, $1 \le k \le K$, $X(\boldsymbol{\alpha})$, but also through those coefficients c_p, c_0 and $L(X)$

of IFE shape functions (2.2) represented by the last variable $\boldsymbol{\alpha}$ in the following formula for the discretized integrand:

$$J_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), \ldots, \mathbf{u}_h^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) := J(u_h^1(\boldsymbol{\alpha}), u_h^2(\boldsymbol{\alpha}), \ldots, u_h^K(\boldsymbol{\alpha}); X(\boldsymbol{\alpha}), \Gamma(\cdot, \boldsymbol{\alpha})). \tag{3.3}$$

The discretized integrand defined in (3.3) can be then used to define a discretized objective function:

$$\mathcal{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), \ldots, \mathbf{u}_h^K(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \int_{\Omega_0} J_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), \ldots, \mathbf{u}_h^K(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) dX. \tag{3.4}$$

In many applications such as those to be presented in Sect. 4, the approximation optimality of the IFE solution $u_h^k(X)$, $1 \le k \le K$ implies that the discretized objective function $\mathcal{J}_h$ in this IFE method is also an optimal approximation to the exact objective functional given by (1.5) regardless of the interface location in the fixed mesh.

Therefore, we propose an IFE method on a fixed mesh of Ω for solving the inverse geometric problem formulated as a shape optimization with (1.1)–(1.5) by carrying out a constrained multi-variable optimization as follows: find the design variable $\boldsymbol{\alpha}^*$ such that

$$\begin{aligned} \boldsymbol{\alpha}^* = \text{argmin } \mathcal{J}_h(\boldsymbol{\alpha}), \quad & \mathcal{J}_h(\boldsymbol{\alpha}) = \mathcal{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), \ldots, \mathbf{u}_h^K(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\ \text{subject to} \quad & \mathbf{A}^k(\boldsymbol{\alpha})\mathbf{u}_h^k(\boldsymbol{\alpha}) - \mathbf{F}^k(\boldsymbol{\alpha}) = \mathbf{0}, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{3.5}$$

The proposed method follows the idea of discrete derivatives [74], i.e., we first discretize the whole system and then calculate the gradient of the discretized objective function with respect to the vector $\boldsymbol{\alpha}$ for optimization. This methodology has been used in many shape optimization applications, see [60,68,74] and reference therein.

3.1 A Velocity Field for Sensitive Computations

In shape optimization, the so called velocity field $\frac{\partial X}{\partial \boldsymbol{\alpha}}$ [68] is a key ingredient in the sensitivity analysis. In this subsection, we follow the ideas in [34,43,46,60] to generate a velocity field on a fixed mesh for the IFE-based shape optimization method. As suggested by these references, all the points located in non-interface elements can be considered as constant functions of the design variable $\boldsymbol{\alpha}$, i.e., the velocity field vanishes on all the non-interface elements. So we only need to discuss the velocity field on interface elements. In the following discussions, we use D_{α_j} to denote the total derivative operator with respect to the j-th design variable α_j, $j \in \mathcal{D}$, and $D_{\boldsymbol{\alpha}}$ is the corresponding gradient operator, and use $\frac{\partial}{\partial \alpha_j}$ and $\frac{\partial}{\partial \boldsymbol{\alpha}}$ to denote the standard partial differential operators and the gradient operator with respect to α_j and $\boldsymbol{\alpha}$.

Consider a typical interface element $T = \triangle A_1A_2A_3 \in \mathcal{T}_h^{int}$, without loss of generality, we assume that the parameterized interface $\Gamma(t, \boldsymbol{\alpha}) = (x(t, \boldsymbol{\alpha}), y(t, \boldsymbol{\alpha}))$, $t \in [0, 1]$ intersects with the boundary of T at $P(\boldsymbol{\alpha}) \in \overline{A_1A_2}$ and $Q(\boldsymbol{\alpha}) \in \overline{A_1A_3}$, see the first sketch in Fig. 2, but neither P nor Q coincides with vertices of T. All the results derived from now on are readily extended to the case in which one of the interface-mesh intersection points P and Q is a vertex of T. Following the idea used in [34,43,60], we partition T into three sub-elements as follows: $T_1 = \triangle A_1PQ$, $T_2 = \triangle A_2QP$, $T_3 = \triangle A_3QA_2$, and let $\hat{T} = \triangle \hat{B}_1\hat{B}_2\hat{B}_3$ be the usual reference element with vertices $\hat{B}_1 = (0, 0)^T$, $\hat{B}_2 = (1, 0)^T$, $\hat{B}_3 = (0, 1)^T$, see the 2nd and the 3rd sketches in Fig. 2. Then, the standard affine mappings from the reference element $\hat{T} = \triangle \hat{B}_1\hat{B}_2\hat{B}_3$ to T_m, $m = 1, 2, 3$ provide a relation between the points in T and the design variables $\boldsymbol{\alpha}$ as follows:

$$X(\boldsymbol{\alpha}) = \mathcal{F}_m(\boldsymbol{\alpha}, \xi, \eta) = \mathbf{J}_m(\boldsymbol{\alpha}) \begin{pmatrix} \xi \\ \eta \end{pmatrix} + A_m, \quad \text{for } \begin{pmatrix} \xi \\ \eta \end{pmatrix} \in \hat{T}, \quad m = 1, 2, 3, \tag{3.6}$$

where the matrix $\mathbf{J}_m(\boldsymbol{\alpha})$ is the Jacobian matrix of $\mathcal{F}_m$ such that $\mathbf{J}_1(\boldsymbol{\alpha}) = (P(\boldsymbol{\alpha}) - A_1, Q(\boldsymbol{\alpha}) - A_1)$, $\mathbf{J}_2(\boldsymbol{\alpha}) = (Q(\boldsymbol{\alpha}) - A_2, P(\boldsymbol{\alpha}) - A_2)$, $\mathbf{J}_3(\boldsymbol{\alpha}) = (Q(\boldsymbol{\alpha}) - A_3, A_2 - A_3)$. Then, the proposed velocity $\mathbf{V}^j$ is constructed by differentiating the relation (3.6) with respect to the j-th design variable:

$$\mathbf{V}^j(X) = \begin{cases} \mathbf{V}_T^j(X) = \mathbf{0}, & \text{if } T \notin \mathcal{T}_h^{int}, \\ \mathbf{V}_T^j(X) = (D_{\alpha_j} \mathbf{J}_m(\boldsymbol{\alpha})) \mathbf{J}_m^{-1}(\boldsymbol{\alpha})(X(\boldsymbol{\alpha}) - A_m), & \text{if } T \in \mathcal{T}_h^{int} \text{ and } X \in T_m, m = 1, 2, 3. \end{cases} \tag{3.7}$$

In order to implement the formula (3.7), we further need the derivatives of the interface-mesh intersection points with respect to α_j, i.e., the velocity at these intersection points, which are given by the explicit formula in Lemma 3.1 below. Actually, these derivatives will be also used in deriving the shape derivatives of IFE shape functions presented in Sect. 3.2. Without loss of generality, we only discuss the intersection point $P = (x_P, y_P)$ and the discussion for Q is similar. Let $\hat{t}_P = \hat{t}_P(\boldsymbol{\alpha}) \in [0, 1]$ be the parameter corresponding to P and express P as $P = (x(\hat{t}_P(\boldsymbol{\alpha}), \boldsymbol{\alpha}), y(\hat{t}_P(\boldsymbol{\alpha}), \boldsymbol{\alpha}))$ in which the second variable $\boldsymbol{\alpha}$ in x and y indicates the dependence through the formula of parameterized curve $\Gamma(t, \boldsymbol{\alpha})$, for example, the coefficients in a cubic spline.

Lemma 3.1 *Assume $\Gamma(t, \boldsymbol{\alpha})$ is not tangent to A_1A_2 at P. Then the function $P = P(\hat{t}_P(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ is differentiable and its velocity defined as the total derivatives $D_{\alpha_j} P$ with respect to α_j, $j \in \mathcal{D}$ are determined by the following linear system:*

$$\begin{aligned} & M_P(\hat{t}_P)\ D_{\alpha_j} P = b_{P,j}(\hat{t}_P), \quad \forall\ j \in \mathcal{D}, \\ & \textit{with} \quad M_P(\hat{t}_P) = \begin{bmatrix} y_2 - y_1 & -(x_2 - x_1) \\ \frac{\partial y}{\partial t}|_{t=\hat{t}_P} & -\frac{\partial x}{\partial t}|_{t=\hat{t}_P} \end{bmatrix} \\ & \textit{and} \ \ b_{P,j}(\hat{t}_P) = \begin{bmatrix} 0 \\ \frac{\partial y}{\partial t}|_{t=\hat{t}_P} \frac{\partial x}{\partial \alpha_j}|_{t=\hat{t}_P} - \frac{\partial x}{\partial t}|_{t=\hat{t}_P} \frac{\partial y}{\partial \alpha_j}|_{t=\hat{t}_P} \end{bmatrix}. \end{aligned} \tag{3.8}$$

Proof See the "Appendix A.1". □

The formula given in Lemma 3.1 to compute $D_{\alpha_j} P$ relies on the fact that P is between the vertices A_1 and A_2; hence, this formula can be easily modified for computing $D_{\alpha_j} Q$.

We note that $\partial x/\partial t$ and $\partial y/\partial t$ needed in formulas (3.8) are readily obtainable for the parameterized curve $\Gamma(t, \boldsymbol{\alpha})$. But $\partial x/\partial \alpha_j$ and $\partial y/\partial \alpha_j$ depend on the specific parameterization of Γ. For example, let $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_x, \boldsymbol{\alpha}_y)$ be such that $\boldsymbol{\alpha}_x$ and $\boldsymbol{\alpha}_y$ are the vectors of the x and y coordinates of control points, and let $\Gamma(t, \boldsymbol{\alpha}) = (x(t, \mathbf{c}_x(\boldsymbol{\alpha}_x)), y(t, \mathbf{c}_y(\boldsymbol{\alpha}_y)))$, $t \in [0, 1]$ be the corresponding parametric cubic spline whose coefficient vectors $\mathbf{c}_x(\boldsymbol{\alpha}_x)$ and $\mathbf{c}_y(\boldsymbol{\alpha}_y)$ are determined by two linear systems $\mathbf{W}_x \mathbf{c}_x = \mathbf{r}_x(\boldsymbol{\alpha}_x)$ and $\mathbf{W}_y \mathbf{c}_y = \mathbf{r}_y(\boldsymbol{\alpha}_y)$ [27], respectively. We note that matrices $\mathbf{W}_s, s = x, y$ are independent of $\boldsymbol{\alpha}$ and the vectors $\mathbf{r}_s(\boldsymbol{\alpha}_s), s = x, y$ are simple polynomials of $\boldsymbol{\alpha}_s, s = x, y$. Hence we can solve for $\frac{\partial \mathbf{c}_x}{\partial \alpha_j}$ from $\mathbf{W}_x \frac{\partial \mathbf{c}_x}{\partial \alpha_j} = D_{\alpha_j} \mathbf{r}_x(\boldsymbol{\alpha}_x)$, with $1 \le j \le dim(\boldsymbol{\alpha}_x)$ and solve for $\frac{\partial \mathbf{c}_y}{\partial \alpha_j}$ from $\mathbf{W}_y \frac{\partial \mathbf{c}_y}{\partial \alpha_j} = D_{\alpha_j} \mathbf{r}_y(\boldsymbol{\alpha}_y)$ with $dim(\boldsymbol{\alpha}_x) + 1 \le j \le dim(\boldsymbol{\alpha}_x) + dim(\boldsymbol{\alpha}_y)$. Then, we obtain $\frac{\partial x}{\partial \alpha_j}|_{t=\hat{t}_P} = x(\hat{t}_P, \frac{\partial \mathbf{c}_x}{\partial \alpha_j})$ and $\frac{\partial y}{\partial \alpha_j}|_{t=\hat{t}_P} = y(\hat{t}_P, \frac{\partial \mathbf{c}_y}{\partial \alpha_j})$.

We now end this subsection by presenting some properties of the proposed velocity field.

Theorem 3.1 *For any $j \in \mathcal{D}$, the velocity $\mathbf{V}^j(X)$ defined in (3.7) has the properties:*

P1: *on each interface element $T = \triangle A_1A_2A_3 \in \mathcal{T}_h^{int}$, there hold*

$$\mathbf{V}_T^j|_{A_mP} = \frac{\|X - A_m\|}{\|P - A_m\|} D_{\alpha_j} P, \quad m = 1, 2$$

$$\mathbf{V}_T^j|_{A_mQ} = \frac{\|X - A_m\|}{\|Q - A_m\|} D_{\alpha_j} Q, \quad m = 1, 2, 3, \tag{3.9a}$$

$$\mathbf{V}_T^j|_{PQ} = \frac{\|X - Q\|}{\|P - Q\|} D_{\alpha_j} P + \frac{\|X - P\|}{\|P - Q\|} D_{\alpha_j} Q, \qquad \mathbf{V}_T^j|_{A_2A_3} = \mathbf{0}, \tag{3.9b}$$

$$\operatorname{div}(\mathbf{V}_{T_m}^j) = \operatorname{tr}\left((D_{\alpha_j}\mathbf{J}_m)\mathbf{J}_m^{-1}\right), \quad m = 1, 2, 3; \tag{3.9c}$$

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

P3: *when restricted on each interface edge e, $\mathbf{V}^j(X)$ is in the same direction as the edge e.*

Proof All of these properties can be verified by direct calculations. □

3.2 Shape Derivatives of IFE Shape Functions

In the proposed IFE method described by (3.5), the IFE basis functions ϕ_i, $1 \le i \le |\mathcal{N}_h|$ on the chosen fixed interface independent mesh are directly employed in the objective function $\mathcal{J}_h$ according to (3.1)–(3.3). By their construction described in (2.2)–(2.4), the IFE basis functions change when the interface $\Gamma(t, \boldsymbol{\alpha})$, $t \in [0, 1]$ moves because of the variations in the design variable $\boldsymbol{\alpha}$. Hence, the gradient of the objective function $\mathcal{J}_h$ in this IFE method inevitably involves the derivatives of the IFE basis functions with respect to $\boldsymbol{\alpha}$. By definition, each IFE basis function is a piecewise polynomial that is a linear combination of the IFE shape functions on each element according to (2.5) or (2.1) depending on whether the element is an interface element or not. Consequently, the derivative of an IFE basis function ϕ_i with respect to $\boldsymbol{\alpha}$ is zero on each non-interface element where all the shape functions are independent of $\boldsymbol{\alpha}$, and our focus in this subsection will be the derivative of IFE shape functions with respect to $\boldsymbol{\alpha}$ on interface elements. We note that [60,79] presented similar approaches to calculate the shape derivative for special finite element shape functions.

Consider a typical interface element $T = \triangle A_1A_2A_3$ configured as in Fig. 2. By (2.2) and the discussions at the beginning of this section, we express an IFE shape function $\psi_T^{int}(X)$ on T as $\psi_T^{int}(X) = \psi_T^{int}(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ to emphasize that the design variable $\boldsymbol{\alpha}$ influences the value of ψ_T^{int} not only through the spatial variable X which is a function of $\boldsymbol{\alpha}$ according to (3.6), but also directly through its coefficients c_0, $\mathbf{c}$ and the coefficients of $L(X)$. However, the rate of change for an IFE shape function ψ_T^{int} with respect to α_j, $j \in \mathcal{D}$ through $X(\boldsymbol{\alpha})$ is readily known by the simple chain rule for differentiation because the velocity field $\frac{\partial X}{\partial \alpha_j}$ is already discussed in Sect. 3.1. Therefore, we only need to discuss the rate of change for an IFE shape function ψ_T^{int} with respect to α_j, $j \in \mathcal{D}$ not through $X(\boldsymbol{\alpha})$, and this rate of change is referred as a shape derivative in the shape optimization literature [34].

First, by their formulas given in Sect. 2, both $L(X)$ and $\bar{\mathbf{n}}$ depend on the design variable $\boldsymbol{\alpha}$ because of their dependence on the interface-mesh intersection points $P = (x_P, y_P)^T$ and $Q = (x_Q, y_Q)^T$ that are functions of $\boldsymbol{\alpha}$. By direct calculations, we have

$$\frac{\partial L}{\partial P} = -\frac{(X-P)^T\bar{\mathbf{t}}\bar{\mathbf{n}}^T}{\|P-Q\|} - \bar{\mathbf{n}}^T, \quad \frac{\partial L}{\partial Q} = \frac{(X-P)^T\bar{\mathbf{t}}\bar{\mathbf{n}}^T}{\|P-Q\|},$$
$$\frac{\partial \bar{\mathbf{n}}}{\partial P} = \frac{-\bar{\mathbf{t}}\bar{\mathbf{n}}^T}{\|P-Q\|}, \quad \frac{\partial \bar{\mathbf{n}}}{\partial Q} = \frac{\bar{\mathbf{t}}\bar{\mathbf{n}}^T}{\|P-Q\|}, \tag{3.10}$$

where $\frac{\partial L}{\partial P} = (\frac{\partial L}{\partial x_P}, \frac{\partial L}{\partial y_P})$, $\frac{\partial L}{\partial Q} = (\frac{\partial L}{\partial x_Q}, \frac{\partial L}{\partial y_Q})$ are 1-by-2 matrices, and $\bar{\mathbf{t}} = \frac{1}{\|P-Q\|}(x_P - x_Q, y_P - y_Q)^T$ is the tangential vector of l, $\frac{\partial \bar{\mathbf{n}}}{\partial P} = (\frac{\partial \bar{\mathbf{n}}}{\partial x_P}, \frac{\partial \bar{\mathbf{n}}}{\partial y_P})$, $\frac{\partial \bar{\mathbf{n}}}{\partial Q} = (\frac{\partial \bar{\mathbf{n}}}{\partial x_Q}, \frac{\partial \bar{\mathbf{n}}}{\partial y_Q})$ are 2-by-2 matrices. Then, by the chain rule, we can use (3.10) to calculate $\frac{\partial L(X,\boldsymbol{\alpha})}{\partial \alpha_j}$ and $\frac{\partial \bar{\mathbf{n}}}{\partial \alpha_j}$ as follows:

$$\frac{\partial L(X,\boldsymbol{\alpha})}{\partial \alpha_j} = \frac{\partial L}{\partial P} D_{\alpha_j} P + \frac{\partial L}{\partial Q} D_{\alpha_j} Q, \quad \frac{\partial \bar{\mathbf{n}}}{\partial \alpha_j} = \frac{\partial \bar{\mathbf{n}}}{\partial P} D_{\alpha_j} P + \frac{\partial \bar{\mathbf{n}}}{\partial Q} D_{\alpha_j} Q, \tag{3.11}$$

in which $D_{\alpha_j} P$ is given by formulas in (3.8) and similar formulas adapted for $D_{\alpha_j} Q$.

Then, by (2.3) and (2.4), we have

$$\frac{\partial c_0}{\partial \alpha_j} = \mu \left(\sum_{p\in\mathcal{I}^-} \left(\frac{\partial c_p}{\partial \alpha_j} \nabla \psi_{p,T}^{non} \cdot \bar{\mathbf{n}} + c_p \nabla \psi_{p,T}^{non} \cdot \frac{\partial \bar{\mathbf{n}}}{\partial \alpha_j} \right) + \sum_{p\in\mathcal{I}^+} v_p \nabla \psi_{p,T}^{non} \cdot \frac{\partial \bar{\mathbf{n}}}{\partial \alpha_j} \right), \tag{3.12a}$$

$$\frac{\partial \boldsymbol{\gamma}}{\partial \alpha_j} = \left(\nabla \psi_{p,T}^{non} \cdot \frac{\partial \bar{\mathbf{n}}}{\partial \alpha_j} \right)_{p\in\mathcal{I}^-}, \qquad \frac{\partial \boldsymbol{\delta}}{\partial \alpha_j} = \left(\frac{\partial L(A_p)}{\partial \alpha_j} \right)_{p\in\mathcal{I}^-}, \tag{3.12b}$$

$$\frac{\partial \mathbf{b}}{\partial \alpha_j} = \left(-\mu \frac{\partial L(A_p)}{\partial \alpha_j} \sum_{q\in\mathcal{I}^+} \nabla \psi_{q,T}^{non} \cdot \bar{\mathbf{n}}\, v_q - \mu L(A_p) \sum_{q\in\mathcal{I}^+} \nabla \psi_{q,T}^{non} \cdot \frac{\partial \bar{\mathbf{n}}}{\partial \alpha_j} v_q \right)_{p\in\mathcal{I}^-}. \tag{3.12c}$$

Furthermore, by (2.3) again, we can compute $\frac{\partial \mathbf{c}}{\partial \alpha_j}$, $j \in \mathcal{D}$ from (3.12b), (3.12c) as follows:

$$\frac{\partial \mathbf{c}}{\partial \alpha_j} = \frac{\partial \mathbf{b}}{\partial \alpha_j} - \mu \frac{\left[\left(\frac{\partial \boldsymbol{\gamma}}{\partial \alpha_j} \right)^T \mathbf{b}\boldsymbol{\delta} + \boldsymbol{\gamma}^T \frac{\partial \mathbf{b}}{\partial \alpha_j} \boldsymbol{\delta} + \boldsymbol{\gamma}^T \mathbf{b} \frac{\partial \boldsymbol{\delta}}{\partial \alpha_j} \right] (1 + \mu \boldsymbol{\gamma}^T \boldsymbol{\delta}) - \mu \boldsymbol{\gamma}^T \mathbf{b}\boldsymbol{\delta} \left[\left(\frac{\partial \boldsymbol{\gamma}}{\partial \alpha_j} \right)^T \boldsymbol{\delta} + \boldsymbol{\gamma}^T \frac{\partial \boldsymbol{\delta}}{\partial \alpha_j} \right]}{(1 + \mu \boldsymbol{\gamma}^T \boldsymbol{\delta})^2}. \tag{3.13}$$

Finally, we use (3.11), (3.12a), and (3.13) to obtain the formula for the shape derivatives of an IFE shape function defined by (2.2) by the following formula: for every $j \in \mathcal{D}$,

$$\frac{\partial \psi_T^{int}(X,\boldsymbol{\alpha})}{\partial \alpha_j} = \begin{cases} \dfrac{\partial \psi_T^{int,-}(X,\boldsymbol{\alpha})}{\partial \alpha_j} = \dfrac{\partial \psi_T^{int,+}(X,\boldsymbol{\alpha})}{\partial \alpha_j} + \dfrac{\partial c_0}{\partial \alpha_j} L(X,\boldsymbol{\alpha}) + c_0 \dfrac{L(X,\boldsymbol{\alpha})}{\partial \alpha_j} & \text{if } X \in \overline{T}^-, \\ \dfrac{\partial \psi_T^{int,+}(X,\boldsymbol{\alpha})}{\partial \alpha_j} = \sum_{p\in\mathcal{I}^-} \dfrac{\partial c_p}{\partial \alpha_j} \psi_{p,T}^{non}(X) & \text{if } X \in \overline{T}^+. \end{cases} \tag{3.14}$$

3.3 The Gradient of the Discretized Objective Function

The gradient of the objective function $\mathcal{J}_h$ is necessary for implementing the proposed IFE method with a chosen minimization algorithm based on a descent direction or trust region. We now put all the preparations in the previous subsections, including the velocity field and shape derivatives, together into the formula for the gradient of the objective function $\mathcal{J}_h$ that

can be executed efficiently within the IFE framework. In the following discussion, we use ∇ to denote the standard gradient operator with respect to X.

Now, by differentiating (3.4) with respect to $\boldsymbol{\alpha}$ through $\mathbf{u}_h^k(\boldsymbol{\alpha})$, $1 \leqslant k \leqslant K$, $X(\boldsymbol{\alpha})$ and $\boldsymbol{\alpha}$ itself, and then following the standard procedure, see Lemma 3.3 in [34] for example, we have the following formula for the material derivative associated to the j-th design variable α_j:

$$D_{\alpha_j}\mathcal{J}_h(\boldsymbol{\alpha}) = \sum_{k=1}^{K}\left(\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k}\cdot D_{\alpha_j}\mathbf{u}_h^k\right) + \int_{\Omega_0}\frac{\partial J_h}{\partial \alpha_j}dX + \int_{\Omega_0}\nabla J_h\cdot\mathbf{V}^j dX + \int_{\Omega_0} J_h \operatorname{div}\left(\mathbf{V}^j\right)dX \tag{3.15}$$

where we have used the fact that $\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k} = \int_{\Omega_0}\frac{\partial J_h}{\partial \mathbf{u}_h^k}dX$. The terms $\frac{\partial J_h}{\partial \alpha_j}$ are derivatives of J_h with respect to its last variable $\boldsymbol{\alpha}$ specified in the generic formula (3.3), not through other variables $\mathbf{u}_h^k(\boldsymbol{\alpha})$s and $X(\boldsymbol{\alpha})$ of J_h , and the computations for $\frac{\partial J_h}{\partial \alpha_j}$ essentially rely on the shape derivatives of IFE shape functions given in (3.14), a typical example to further explain this is given by (4.4) in the next section. Furthermore, as demonstrated by examples presented in the next section, in this fundamental formula for $D_{\alpha_j}\mathcal{J}_h(\boldsymbol{\alpha})$, the terms $\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k}$, ∇J_h, $\frac{\partial J_h}{\partial \alpha_j}$, and J_h are problem dependent, but they are easy to calculate for many applications by explicit formulas. Also, we note that $\mathbf{V}^j$ is given in (3.7) and $\operatorname{div}\left(\mathbf{V}^j\right)$ is given in (3.9c); hence, we proceed to derive the formula for $\left(\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k}\right)\cdot D_{\alpha_j}\mathbf{u}_h$, $j \in \mathcal{D}$ which can be directly used in (3.15).

For $D_{\alpha_j}\mathbf{u}_h^k$, $1 \leq k \leq K$, $j \in \mathcal{D}$, by differentiating the IFE system in (2.17) with respect to α_j, we have the linear system for $D_{\alpha_j}\mathbf{u}_h^k$: $\mathbf{A}^k(\boldsymbol{\alpha})\, D_{\alpha_j}\mathbf{u}_h^k = D_{\alpha_j}\mathbf{F}^k(\boldsymbol{\alpha}) - D_{\alpha_j}\mathbf{A}^k(\boldsymbol{\alpha})\,\mathbf{u}_h^k(\boldsymbol{\alpha})$, $1 \leq k \leq K$. Then, by the standard process in the discretized adjoint method [29], we can compute $\left(\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k}\right)\cdot D_{\alpha_j}\mathbf{u}_h^k$ efficiently (especially when $|\mathcal{D}|$ is large) by solving for $\mathbf{Y^k}$ from $\left(\mathbf{A}^k\right)^T\mathbf{Y}^k = \frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k}$, and then compute

$$\left(\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k}\right)\cdot D_{\alpha_j}\mathbf{u}_h^k = \mathbf{Y}^k\cdot\left(D_{\alpha_j}\mathbf{F}^k(\boldsymbol{\alpha}) - D_{\alpha_j}\mathbf{A}^k(\boldsymbol{\alpha})\,\mathbf{u}_h^k(\boldsymbol{\alpha})\right),\quad 1 \leq k \leq K,\quad 1 \leq j \leq |\mathcal{D}|, \tag{3.16}$$

where $\mathbf{A}^k$ is the matrix for the k-th IFE equation described in (2.17). We further note that the formula (3.16) involves the material derivatives of the global stiffness matrix and load vector with respect to α_j, $j \in \mathcal{D}$, i.e., $D_{\alpha_j}\mathbf{A}^k(\boldsymbol{\alpha})$ and $D_{\alpha_j}\mathbf{F}^k(\boldsymbol{\alpha})$ which can be assembled with the material derivatives of the local matrices for $\mathbf{A}^k$ and the material derivatives of local vectors for $\mathbf{F}^k$ according to the same standard assemblage procedure as that for assembling matrix $\mathbf{A}^k$ and vector $\mathbf{F}^k$. However, the assemblage for $D_{\alpha_j}\mathbf{A}^k(\boldsymbol{\alpha})$ and $D_{\alpha_j}\mathbf{F}^k(\boldsymbol{\alpha})$ only needs to be performed over the interface elements/edges since the material derivatives of the local matrices and local vectors all vanish on the non-interface elements/edges. The derivation of these material derivatives of the local matrices and local vectors follows from Lemma 3.3 of [34] in the direction of the velocity field developed in Sect. 3.1 and the shape derivatives of the IFE shape functions given by the formula (3.14) together with the results in Theorem 3.1. We present the related formulas for these derivatives in the "Appendix A.2".

For the implementation of the proposed IFE-based shape optimization method, we summarize the discretization of forward/inverse problems and the sensitivity computation discussed above into the following algorithm.

Algorithm The IFE Shape Optimization Algorithm

1: Generate a fixed mesh and choose an initial design variable $\boldsymbol{\alpha}$.
2: Loop until convergence.
3: Prepare data:
   a: use the design variables to generate the parametric curve $\Gamma(\boldsymbol{\alpha})$ as the numerical interface;
   b: find the interface-mesh intersection points, interface edges and interface elements.
4: Prepare matrices and vectors for the IFE systems and compute the objective function:
   a: use (2.12) and (2.13) and the IFE spaces given in (2.1) and (2.5) to assemble matrices and vectors $\mathbf{A}^k, \mathbf{F}^k, 1 \le k \le K$ for the IFE systems (2.17);
   b: compute the PPIFE solutions $\mathbf{u}^k, 1 \le k \le K$ by (2.17) and the objective function $\mathcal{J}_h(\boldsymbol{\alpha})$ in (3.5).
5: Compute the shape sensitivities:
   a: form the material derivatives of local matrices and vectors according to Appendix A.2, and use them to assemble the global matrices $D_{\alpha_j}\mathbf{A}^k(\boldsymbol{\alpha})$ and vectors $D_{\alpha_j}\mathbf{F}^k(\boldsymbol{\alpha})$;
   b: compute $\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k} \cdot D_{\alpha_j}\mathbf{u}_h^k$ for $k = 1, \ldots, K$, according to (3.16);
   c: compute the terms $\int_{\Omega_0} \frac{\partial J_h}{\partial \alpha_j} dX$, $\int_{\Omega_0} \nabla J_h \cdot \mathbf{V}^j dX$ and $\int_{\Omega_0} J_h \mathrm{div}(\mathbf{V}^j) dX$ according to the given shape functional;
   d: compute the material derivatives of $\mathcal{J}_h(\boldsymbol{\alpha})$ according to (3.15).
6: Use $\mathcal{J}_h(\boldsymbol{\alpha})$ and its material derivatives to update the design variable $\boldsymbol{\alpha}$ by a chosen gradient-based optimization algorithm.
7: End loop

In this proposed IFE Shape Optimization Algorithm, we note that the mesh can be fixed during the optimization process, and the only mesh information needed to be updated are those interface-mesh intersection points and interface elements/edges. Consequently, the global matrices $\mathbf{A}^k$ and vectors $\mathbf{F}^k$ in step 4 remain the same size and algebraic structure on this fixed mesh, which is beneficial for implementation. Also, they do not need to be completely re-assembled in each iteration, because only those global basis functions whose supports overlap with the interface elements/edges in two consecutive iterations are changed. As a result, their assemblage can be done very efficiently by just updating those entries corresponding to the global basis functions whose supports overlap with the interface elements/edges in the previous and the current iteration.

In step 5 above (computing the shape sensitivities), we emphasize that the velocity fields and the shape derivatives of IFE shape functions are only needed on interface elements, which can be implemented according to the analytical formulas (3.7) and (3.14). These two quantities vanishing over all the non-interface elements make the whole procedure of shape sensitivity computation remarkably efficient. Firstly, the integration for the terms $\int_{\Omega_0} \nabla J_h \cdot \mathbf{V}^j dX$ and $\int_{\Omega_0} J_h \mathrm{div}(\mathbf{V}^j) dX$ in the material derivative of the objective functional (3.15) only needs to be done on those interface elements intersecting with Ω_0 because the involved integrands all vanish on the non-interface elements. Secondly, assembling the matrices $D_{\alpha_j}\mathbf{F}^k(\boldsymbol{\alpha})$ and $D_{\alpha_j}\mathbf{A}^k(\boldsymbol{\alpha})$, i.e., the material derivatives of global matrices $\mathbf{A}^k$ and $\mathbf{F}^k$, is also a very efficient process since it is only performed over the interface elements/edges by the explicit formulas given in "Appendix A.2". In summary, the computations of shape sensitivity in this algorithm only need to be carried out over interface elements whose number is in the order of $O(h^{-1})$ versus the number of all elements in the order of $O(h^{-2})$ in the mesh. In contrast, preparing $D_{\alpha_j}\mathbf{F}^k(\boldsymbol{\alpha})$ and $D_{\alpha_j}\mathbf{A}^k(\boldsymbol{\alpha})$ is usually expensive within the Lagrange

framework where a global velocity field requires to carry out the assemblages over all elements in a mesh [21], and $D_{\alpha_j}\mathbf{F}^k(\boldsymbol{\alpha})$ and $D_{\alpha_j}\mathbf{A}^k(\boldsymbol{\alpha})$ are usually prepared approximately in methods in the Eullerian framework, see the related discussions in [24,76,79].

Finally, we note that the proposed IFE shape optimization algorithm is highly parallelizable because computing the velocity fields $\mathbf{V}^j$ (3.7), shape derivatives of IFE shape functions $\frac{\partial \psi_T^{int}}{\partial \alpha_j}$ (3.14) and the material derivatives of stiffness matrices and vectors $D_{\alpha_j}\mathbf{A}^k(\boldsymbol{\alpha})$ and $D_{\alpha_j}\mathbf{F}^k(\boldsymbol{\alpha})$, i.e., the material derivatives of objective functions (3.15) with respect to each individual design variable α_j, are independent with each other. Hence these computations can be done very efficiently with an easy implementation on modern parallel computers.

Therefore, we believe these properties together with the optimal accuracy of PPIFE solutions (2.10) and the resulted optimal accuracy of discretized objective functions, regardless of the interface location, make the proposed IFE shape optimization algorithm advantageous compared with those in the literature.

4 Some Applications

In this section, we demonstrate how the general IFE shape optimization method proposed in the previous section can use a fixed mesh to solve a wide spectrum of interface inverse problems posed in the format of (1.1)–(1.5) by applying this method to, but not limited to, three representative interface inverse/design problems: (1). an output-least-squares problem [15,17,30]; (2). a Dirichlet–Neumann problem [10,39,70]; and (3). a heat dissipation minimization problem [26,49,81]. The first problem uses the interior data available on the whole or a portion of Ω to reconstruct/design the interface, the second one recovers the interface from the data only available on $\partial\Omega$, and the last one is an application for an optimal design of heat conduction fields. These examples also serve as hints/suggestions how the proposed IFE method can be implemented efficiently.

All numerical examples to be presented are posed on the domain $\Omega = (-1, 1) \times (-1, 1)$ with a fixed Cartesian mesh that is formed by cutting Ω into congruent rectangles and then cutting each rectangle into two triangles along its diagonal line. The parameterized interface $\Gamma(\boldsymbol{\alpha})$ in the shape optimization is expressed by a cubic spline. This choice of parametrization for the interface Γ is based on the accuracy, versatility, and popularity of the cubic spline, and we emphasize that the fixed mesh method developed here can be readily extended to other parameterizations.

4.1 An Output-Least-Squares Problem

In this example, we consider an interface inverse/design problem associated to the interface forward problem described by (1.1) and (1.2) with $K = 1$, in which we assume an observation data (or a target function in optimal design application) $\bar{u}$ for the solution u^1 to the forward problem described by (1.1)–(1.3) with a pure Dirichlet boundary condition g_D^1 on $\partial\Omega$ is available on a sub-domain $\Omega_0 \subseteq \Omega$, and we need to recover/design the location and the shape of the interface from $\bar{u}$ by solving an output-least-squares problem [17,42], i.e., by optimizing the following shape functional

$$\mathcal{J}(\Gamma) = \int_{\Omega_0} (u^1 - \bar{u})^2 dX. \tag{4.1}$$

This problem arises from oil/underwater reservoirs [25,77] and optimal designing of cooling elements in battery systems [65]. A related time dependent problem is discussed in [33]. Applying the IFE method proposed in (3.5) to the inverse problem formulated in (4.1) suggests to seek the design variable $\boldsymbol{\alpha}^*$ by solving the following constrained optimization problem:

$$\begin{aligned} \boldsymbol{\alpha}^* = \text{argmin } \mathcal{J}_h(\boldsymbol{\alpha}), \quad \mathcal{J}_h(\boldsymbol{\alpha}) = \int_{\Omega_0} J_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) dX, \\ \text{subject to} \quad \mathbf{A}^1(\boldsymbol{\alpha})\mathbf{u}_h^1(\boldsymbol{\alpha}) - \mathbf{F}^1(\boldsymbol{\alpha}) = \mathbf{0}, \end{aligned} \tag{4.2}$$

where, expressing the IFE solution $u_h^1(\boldsymbol{\alpha}) = u_h^1(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ in the format given in (3.1), we have

$$\begin{aligned} J_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \big(\tilde{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha})\big)^2, \\ \text{with } \tilde{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \sum_{i=1}^{|\mathring{\mathcal{N}}_h|} u_i^1(\boldsymbol{\alpha})\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) + \sum_{i=|\mathring{\mathcal{N}}_h|+1}^{|\mathcal{N}_h|} g_D^1(X_i)\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) - \bar{u}. \end{aligned} \tag{4.3}$$

It is easy to see that the discrete objective function $\mathcal{J}_h(\boldsymbol{\alpha})$ has the optimal second order accuracy to approximate the continuous objective functional $\mathcal{J}(\Gamma(\alpha))$ regardless of the interface location and shape on a fixed mesh. According to (4.3), the evaluation of $J_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ is straightforward and it is obvious that

$$\begin{aligned} \nabla J_h = 2\tilde{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \left(\sum_{i=1}^{|\mathring{\mathcal{N}}_h|} u_i^1(\boldsymbol{\alpha}) \nabla\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \right. \\ \left. + \sum_{i=|\mathring{\mathcal{N}}_h|+1}^{|\mathcal{N}_h|} g_D^1(X_i)\nabla\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) - \nabla\bar{u} \right), \\ \frac{\partial J_h}{\partial \alpha_j} = 2\tilde{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \left(\sum_{i=1}^{|\mathring{\mathcal{N}}_h|} u_i^1(\boldsymbol{\alpha}) \frac{\partial\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})}{\partial\alpha_j} + \sum_{i=|\mathring{\mathcal{N}}_h|+1}^{|\mathcal{N}_h|} g_D^1(X_i) \frac{\partial\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})}{\partial\alpha_j} \right), \end{aligned} \tag{4.4}$$

where $\bar{u}$ is assumed to be optimization independent, $\nabla\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ is the standard gradient with respect to X, and $\frac{\partial\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha})}{\partial\alpha_j}$, $j \in \mathcal{D}$ are the shape derivatives of the global IFE basis functions which are zero on all the non-interface elements and computed by (3.14) on interface elements. Furthermore, a direct calculation leads to

$$\mathcal{J}_h = \begin{pmatrix} \mathbf{u}_h^1 \\ \mathbf{g}_D^1 \end{pmatrix}^T \mathbf{M} \begin{pmatrix} \mathbf{u}_h^1 \\ \mathbf{g}_D^1 \end{pmatrix} - 2 \begin{pmatrix} \mathbf{u}_h^1 \\ \mathbf{g}_D^1 \end{pmatrix}^T \bar{\mathbf{u}} + \int_{\Omega_0} \bar{u}^2 dX, \quad \frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^1} = 2\mathbf{M}_0 \begin{pmatrix} \mathbf{u}_h^1 \\ \mathbf{g}_D^1 \end{pmatrix} - 2\bar{\mathbf{u}}_0, \tag{4.5}$$

$$\text{where } \mathbf{M} = \left(\int_{\Omega_0} \phi_i \phi_j dX \right)_{i=1,j=1}^{|\mathcal{N}_h|,|\mathcal{N}_h|} \in \mathbb{R}^{|\mathcal{N}_h|\times|\mathcal{N}_h|}, \quad \bar{\mathbf{u}} = \left(\int_{\Omega_0} \bar{u}\phi_i dX \right)_{i=1}^{|\mathcal{N}_h|} \in \mathbb{R}^{|\mathcal{N}_h|\times 1}, \tag{4.6}$$

and $\mathbf{M}_0, \bar{\mathbf{u}}_0$ are formed by the first $|\mathring{\mathcal{N}}_h|$ rows of $\mathbf{M}$ and $\bar{\mathbf{u}}$, respectively. Formulas above confirm the observation that the computations for $\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k}$, ∇J_h, $\frac{\partial J_h}{\partial \alpha_j}$ and J_h itself are problem dependent

Table 1 Configuration for the Output-Least-Squares Problem

Cases	β	Interface S and initial guess	Data $\bar{u}$
Case 1	$\beta^- = 1$ $\beta^+ = 20$	$S = (x^2+y^2)^2(1+0.8\sin(6\arctan(y/x))) - 0.1$ $S_0 = (x+0.6)^2 + (y+0.2)^2 - (\pi/9)^2$	$\bar{u} = S/\beta^s$ in Ω^s $s = \pm$
Case 2	$\beta^1 = 1$ $\beta^2 = 10$ $\beta^3 = 100$	$S = 4\sin(\pi x)\cos(\pi y + \pi/2) - 2$ $S_0^1 = 64x^2 + 144(y+0.5)^2 - \pi^2$ $S_0^2 = 64x^2 + 144(y-0.5)^2 - \pi^2$	$\bar{u} = S/\beta^i$ in Ω^i $i = 1, 2, 3$
Case 3	$\beta^- = 1$ $\beta^+ = 10$ or 1000	$S = r - 1$, where $r = (16x^2 + 64(y-0.4)^2)/\pi^2$ $S_0 = (x-0.4)^2 + (y-0.2)^2 - (\pi/6.28)^2$	$\bar{u} = \frac{1024}{\pi^4\beta^s}(r^{\frac{5}{2}} - 1)$ $+ \frac{1024}{\pi^4\beta^-}$ in Ω_0^s, $s = \pm$.

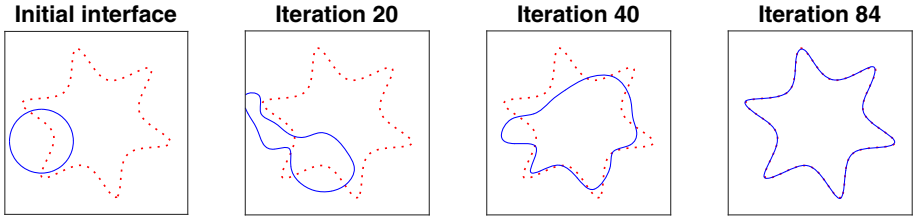

Fig. 3 Optimization process for case 1 (Color figure online)

but they are usually straightforward to calculate within the IFE framework. These preparations can then be utilized in the proposed IFE Shape Optimization Algorithm presented in Sect. 3.3.

We now present three specific cases for this interface inverse/design problem whose key data are described in Table 1. In this table, $S(x, y) = 0$ is the target curve Γ to be recovered that is plotted as a dotted curve (in red color) in the related figures. We use the BFGS optimization algorithm [62] in step 6 of the IFE Shape Optimization Algorithm presented in Sect. 3.3, for which, $S_0(x, y) = 0$ is the initial curve that is plotted as a solid curve (in blue color) in the first plot of related figures as all other presented approximate curves in shape optimization by the BFGS iterations.

Case 1 The data $\bar{u}(X)$ is given on the whole Ω, i.e., $\Omega_0 = \Omega$. The numerical curve is a parametric cubic spline with 20 control points, and the target curve has a star shape representing a certain complexity in the convexity. Some approximate curves generated in the BFGS iterations are presented in Fig. 3 from which we can see a quick evolution of the numerical curve towards to the target curve for this inverse/design problem even with a complicated geometry, and this suggests a benefit of the accurate gradient formula available for the proposed IFE method.

Case 2 We demonstrate how the proposed algorithm can handle an interface inverse/design problem whose target interface consists of multiple components. For this purpose and for simplicity, we consider the case in which an interface Γ is formed by the two simple curves. We denote the sub-domain inside the upper-left dotted curve (in red color) by Ω^1, the sub-domain inside the lower-right dotted curve (in red color) by Ω^2, and denote the sub-domain outside these two closed dotted curves by Ω^3, see Fig. 4, and the corresponding parameter is given by β^i, $i = 1, 2, 3$. Each numerical curve component is a parametric cubic spline with 15 control points and 30 control points in total. We notice that the numerical curve component started from S_0^1 converges to the exact curve component much faster than that

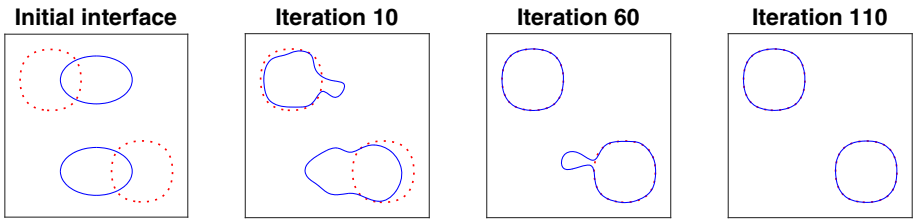

Fig. 4 Optimization process for case 2 (Color figure online)

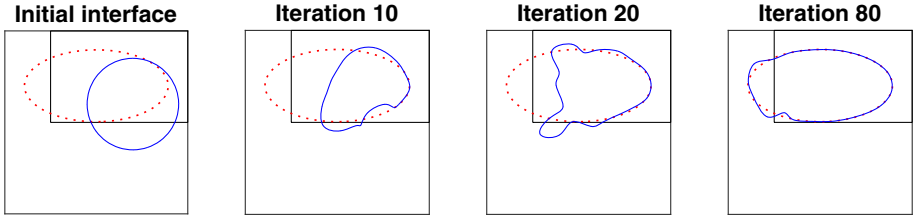

Fig. 5 Optimization process for case 3 with $\beta^- = 1$, $\beta^+ = 10$ (Color figure online)

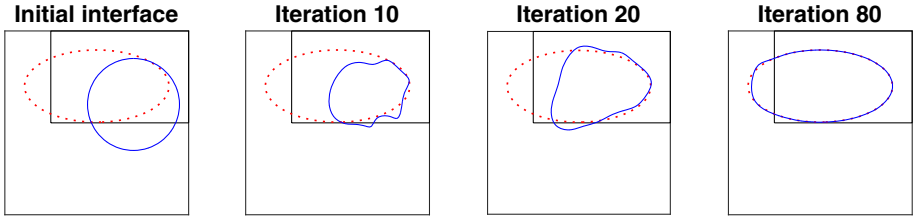

Fig. 6 Optimization process for case 3 with $\beta^- = 1$, $\beta^+ = 1000$ (Color figure online)

started from S_0^2. We believe the objective function is more sensitive to the design variables for the first numerical curve component than the second because the jump β^3/β^1 is much larger than β^3/β^2 in this example, and the gradient in the proposed IFE method is capable to capture this kind of subtle dependence of the objective function on the design variables.

Case 3 The data function $\bar{u}$ is given on a proper sub-domain $\Omega_0 = [-0.5, 1] \times [0, 1]$ in the upper-right of Ω illustrated in Fig. 5. In Fig. 5, we observe that the converged numerical curve is a much better approximation to the target interface curve Γ inside Ω_0 than outside, and we believe this is a reasonable consequence of the available data function $\bar{u}$ given only on Ω_0. In addition, we also test this example with a larger jump case, i.e., β^-=1, $\beta^+ = 1000$. The numerical results are presented in Fig. 6 which also shows a satisfactory reconstruction. We think these two examples suggest again that the gradient in the proposed IFE method can capture the nature of the interface inverse problem in accordance with the available data.

4.2 The Dirichlet–Neumann Problem with a Single Measurement

In this group of numerical examples, we apply the proposed IFE method to the popular but challenging Dirichlet–Neumann inverse problem in which we try to recover the interface Γ from a single pair of Dirichlet–Neumann data, i.e., g_D and g_N, provided on the boundary

for an interface forward problem of the elliptic equation described by (1.1)–(1.3). This type of inverse problems have a wide range of applications in electronic impedance tomography (EIT) [10,39,56] where one wishes to detect a material interface by injecting the voltage potential g_D on $\partial\Omega$ and measuring the current density g_N on (or a portion of) $\partial\Omega$. When the charge source $f = 0$, it is referred as the Calderón's inverse conductivity problem [14] which is well-known ill-conditioned since only the data on the boundary $\partial\Omega$ is available for the reconstruction of Γ.

We formulate this inverse problem as a shape optimization problem with a Kohn-Vogelius type functional [10,47,63]:

$$\mathcal{J}(\Gamma) = \int_\Omega (u^1 - u^2)^2 dX, \tag{4.7}$$

where u^1 and u^2 are the solutions of the following interface forward problems:

$$\begin{cases} -\nabla\cdot(\beta\nabla u^1) = f, \\ [u^1]|_\Gamma = 0, \\ [\beta\nabla u\cdot\mathbf{n}]|_\Gamma = 0, \\ u^1 = g_D^1 = g_D, \quad \text{on } \partial\Omega, \end{cases} \qquad \begin{cases} -\nabla\cdot(\beta\nabla u^2) = f, \\ [u^2]|_\Gamma = 0, \\ [\beta\nabla u^2\cdot\mathbf{n}]|_\Gamma = 0, \\ u^2 = g_D^2 = g_D, \quad \text{on } \partial\Omega_D, \ \frac{\partial u^2}{\partial\mathbf{n}} = g_N^2, \quad \text{on } \partial\Omega_N, \end{cases}$$

and $\int_\Omega u^2 dx = u_0$ needs to be imposed when $\partial\Omega_N = \partial\Omega$. Again, we solve this interface inverse problem by seeking the design variable $\boldsymbol{\alpha}^*$ from the following constrained optimization problem:

$$\begin{aligned} \boldsymbol{\alpha}^* = \text{argmin } \mathcal{J}_h(\boldsymbol{\alpha}), \quad \mathcal{J}_h(\boldsymbol{\alpha}) &= \int_{\Omega_0} J_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) dX, \\ \text{subject to} \quad \mathbf{A}^k(\boldsymbol{\alpha})\mathbf{u}_h^k(\alpha) - \mathbf{F}^k(\boldsymbol{\alpha}) &= \mathbf{0}, \quad k = 1, 2, \end{aligned} \tag{4.8}$$

where $J_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \left(\tilde{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha})\right)^2$ with

$$\begin{aligned} \tilde{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = &\sum_{i=1}^{|\mathring{\mathcal{N}}_h|} u_i^1 \phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) + \sum_{i=|\mathring{\mathcal{N}}_h|+1}^{|\mathcal{N}_h|} g_D^1(X_i)\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\ &- \sum_{i=1}^{|\mathcal{N}_h^M|} u_i^2 \phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) - \sum_{i=|\mathcal{N}_h^M|+1}^{|\mathcal{N}_h|} g_D^2(X_i)\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}). \end{aligned} \tag{4.9}$$

Again, we note that the discrete objective function $\mathcal{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ approximates the objective functional $\mathcal{J}(\Gamma(\alpha))$ with an optimal second order independent of interface location and shape, and this objective function can be evaluated quickly by matrix-vector operations as follows:

$$\mathcal{J}_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), \mathbf{u}_h^2(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \left[\begin{pmatrix}\mathbf{u}_h^1 \\ \mathbf{g}_D^1\end{pmatrix} - \begin{pmatrix}\mathbf{u}_h^2 \\ \mathbf{g}_D^2\end{pmatrix}\right]^T \mathbf{M} \left[\begin{pmatrix}\mathbf{u}_h^1 \\ \mathbf{g}_D^1\end{pmatrix} - \begin{pmatrix}\mathbf{u}_h^2 \\ \mathbf{g}_D^2\end{pmatrix}\right], \tag{4.10}$$

where $\mathbf{g}_D^1 = (g_D(X_i))_{i=|\mathring{\mathcal{N}}_h|+1}^{|\mathcal{N}_h|}$, $\mathbf{g}_D^2 = (g_D(X_i))_{i=|\mathcal{N}_h^M|+1}^{|\mathcal{N}|_h}$, and $\mathbf{M}$ is the global mass matrix on the whole domain Ω:

$$\mathbf{M} = \left(\int_\Omega \phi_i\phi_j dX\right)_{i=1,j=1}^{|\mathcal{N}_h|,|\mathcal{N}_h|}. \tag{4.11}$$

Also, formulas similar to those in (4.4) can be easily derived to compute $\frac{\partial J_h}{\partial X}$ and $\frac{\partial J_h}{\partial \alpha_j}$, $j \in \mathcal{D}$. Furthermore, by direct calculations, we have

$$\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^1} = 2\mathbf{M}_1 \left[\begin{pmatrix} \mathbf{u}_h^1 \\ \mathbf{g}_D^1 \end{pmatrix} - \begin{pmatrix} \mathbf{u}_h^2 \\ \mathbf{g}_D^2 \end{pmatrix} \right], \quad \text{and} \quad \frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^2} = 2\mathbf{M}_2 \left[\begin{pmatrix} \mathbf{u}_h^2 \\ \mathbf{g}_D^2 \end{pmatrix} - \begin{pmatrix} \mathbf{u}_h^1 \\ \mathbf{g}_D^1 \end{pmatrix} \right], \tag{4.12}$$

where $\mathbf{M}_1$ and $\mathbf{M}_2$ are the mass matrices formed by the first $|\mathring{\mathcal{N}}_h|$ and $|\mathcal{N}_h^M|$ rows of $\mathbf{M}$, respectively. These derivations together confirm again that formulas for J_h, ∇J_h, $\frac{\partial J_h}{\partial \alpha_j}$, $j \in \mathcal{D}$ as well as $\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^k}$, $k = 1, 2$ in general can be easily derived explicitly and efficiently implemented in the IFE framework. These preparations can then be used in the proposed IFE Shape Optimization Algorithm presented in Sect. 3.3.

For the Dirichlet–Neumann inverse problem, we report 3 experiments that are configured with the target curve $S(x, y) = 0$ and one of the exact solutions $u(X)$ given in Table 2 corresponding to the interface $S(x, y) = 0$ for the interface forward problem. We emphasize that $u(X)$ given in the third column of this table is used only to generate the Dirichlet and Neumann boundary data, i.e., g_D and g_N on $\partial\Omega$, for the related inverse problem. As before, the BFGS algorithm [62] is employed to carry out the shape optimization described by (4.8) according to the proposed IFE Shape Optimization Algorithm, for which, $S_0(x, y) = 0$ given in Table 2 is the initial curve that is plotted as a solid curve (in blue color) in the related figures as all other presented approximate curves in the BFGS iterations and the dotted curve (in red color) is the target curve to be recovered.

Case 1 The Neumann data is given on the whole $\partial\Omega$, and the numerical curve is a parametric cubic spline with 20 control points. These numerical results demonstrate that the proposed IFE method can handle a large shape change, as illustrated in Fig. 7 from the initial interface to the one generated by the third iteration. We note that such a large shape change often causes mesh distortion when body fitting mesh is used, but the proposed IFE method circumvents this issue by using a fixed interface independent mesh. The numerical curve quickly converges to the target curve after about 80 iterations, and this demonstrates again the benefit of the fact that the gradient in the proposed algorithm provides a good sensitivity with respect to the design variables of the numerical curve.

Case 2 We now consider a more difficult Dirichlet–Neumann interface inverse problem whose exact solution interface curve Γ has a non-conical and non-convex kidney-like shape, as illustrated in Fig. 8, and, to the best of our knowledge, there is no general theory to guarantee the uniqueness of the solution to this inverse problem with only one single pair of Dirichelt and Neumann data given on the whole $\partial\Omega$. The numerical curve is a parametric cubic spline with 20 control points. We note this example takes far more iterations to converge and we believe this is caused by the challenging nature of this interface inverse problem whose exact solution is non-convex; nevertheless, the proposed IFE method still produces an approximate solution quite satisfactory (Fig. 8).

Case 3 In this case, the Neumann data is provided only on a proper subset of the boundary $\partial\Omega$. Specifically, the true interface Γ is the level set $S(x, y) = 0$ plotted as the dotted curve (in red color) in Fig. 9 that separates Ω into two sub-domains Ω^- and Ω^+ below and above Γ, and the Neumann data function g_N is given only on the lower and upper edge of the square domain Ω. The numerical interface is a 1-D cubic spline $y = y(t)$, $t \in [-1, 1]$ with 10 control points whose end points match the exact interface. The last plot in Fig. 9 shows

Table 2 Configuration for the Dirichlet–Neumann Problem

Cases	β	Interface S and initial guess	Exact u
Case 1	$\beta^- = 1$ $\beta^+ = 10$	$S = r - 1$, where $r = (100(x-0.4)^2 + 36(y+0.3)^2)/\pi^2$ $S_0 = (x-0.1)^2 + y^2 - (\pi/4)^2$	$u = \frac{3600}{\pi^4\beta^s}(r^{\frac{5}{2}} - 1)$ $+ \frac{3600}{\pi^4\beta^-}$ in Ω^s $s = \pm$
Case 2	$\beta^- = 1$ $\beta^+ = 2$	$S = (2((x+0.5)^2 + y^2) - x - 0.2)^2 - ((x+0.5)^2 + y^2) + 0.3$ $S_0 = 64(x-0.5)^2 + 16y^2 - \pi^2$	$u = S/\beta^s$ in Ω^s, $s = \pm$
Case 3	$\beta^- = 1$, $\beta^+ = 2$	$S = \sin(\pi x) + \frac{\pi}{1.5}y + 0.1$ $S_0 = y + 0.15/\pi$	$u = S/\beta^s$ in Ω^s, $s = \pm$

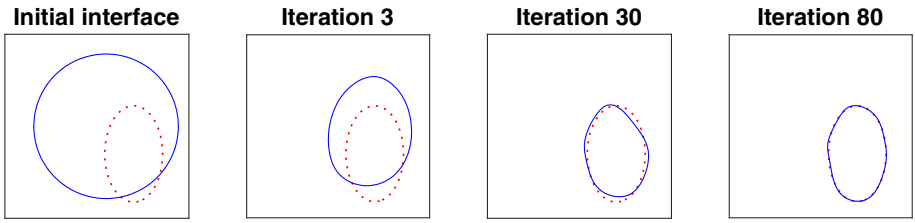

Fig. 7 Reconstruction process for case 1 (Color figure online)

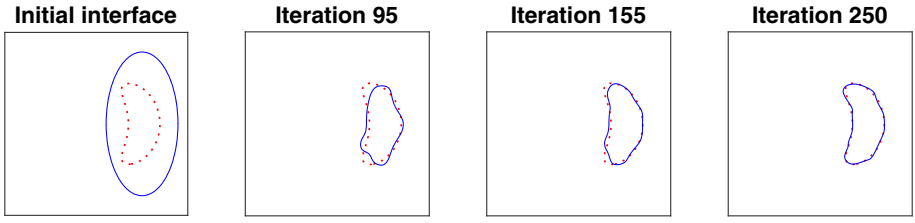

Fig. 8 Reconstruction process for case 2 (Color figure online)

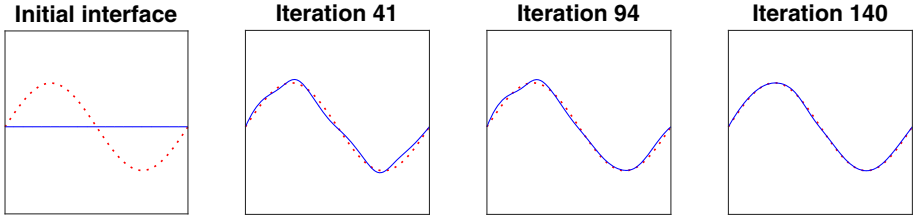

Fig. 9 Reconstruction process for case 3 (Color figure online)

that the numerical curve after 140 iterations matches the exact curve well. This demonstrates that the proposed IFE method can treat a Dirichlet–Naumann interface inverse problem that has a limited Neumann data measured on part of the boundary of Ω. We also test the case in which the Neumann data is on the left and right boundary of Ω instead of the lower and upper boundary, but the result is not as satisfactory as the one presented here.

4.3 The Heat Dissipation Problem

We now consider the application of the proposed IFE method to an optimal design problem for a heat system in which the goal is to minimize the overall heat dissipation by optimally distributing two materials in a domain [26,28,81]. This thermal design problem has wide applications such as cooling fins [6,69] and high-conductivity channel of electronic components [9].

In the steady heat conduction situation, this design problem is to find an optimal curve Γ^* separating two chosen materials that can minimize the following objective functional [26]:

$$\mathcal{J}(\Gamma) = \int_{\Omega} \nabla u^1 \cdot (\beta \nabla u^1) dX \quad \text{subject to} \quad |\Omega^-| \leqslant \theta |\Omega|, \tag{4.13}$$

where u^1 is the solution to the interface problem described by (1.1)–(1.2) with $K = 1$ with a Dirichlet boundary condition, Ω^- is the sub-domain filled with the high conductivity

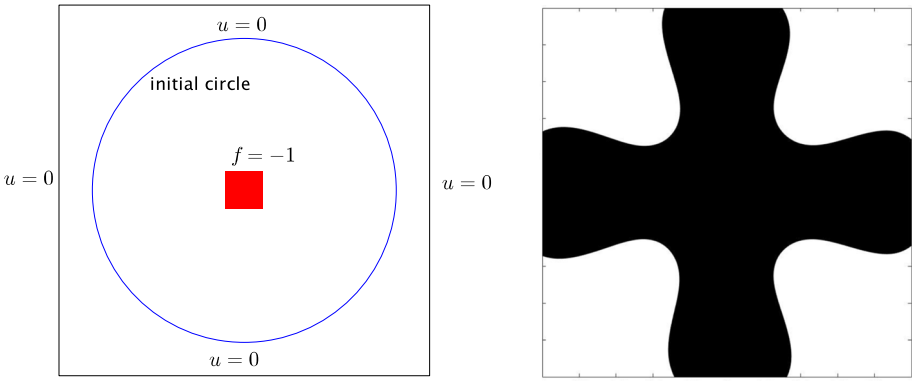

Fig. 10 The heat dissipation problem. **a** The heating conditions and initial guess. **b** The optimal design (Color figure online)

material, and $\theta \in (0, 1)$ is a prescribed design parameter. By the proposed IFE method (3.5), we seek a design variable $\boldsymbol{\alpha}^*$ by solving the following constrained optimization problem:

$$\boldsymbol{\alpha}^* = \text{argmin } \mathcal{J}_h(\boldsymbol{\alpha}), \quad \mathcal{J}_h(\boldsymbol{\alpha}) = \int_{\Omega_0} J_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) dX,$$
$$\text{subject to} \quad \mathbf{A}^1(\boldsymbol{\alpha})\mathbf{u}_h^1(\alpha) - \mathbf{F}^1(\boldsymbol{\alpha}) = \mathbf{0} \text{ and } |\Omega^-| \leqslant \theta|\Omega|, \tag{4.14}$$

where

$$J_h(\mathbf{u}_h^1(\boldsymbol{\alpha}), X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = \beta \left| \sum_{i=1}^{|\mathring{\mathcal{N}}_h|} u_i^1 \nabla\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) + \sum_{i=|\mathring{\mathcal{N}}_h|+1}^{|\mathcal{N}_h|} g_D(X_i) \nabla\phi_i(X(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \right|^2. \tag{4.15}$$

It is easy to show that the discrete objective function can approximate the true shape functional (4.13) with an optimal first order accuracy independent of the interface shape and location. Also, similar to (4.4)–(4.6) again, formulas can be easily derived for $\mathcal{J}_h(\boldsymbol{\alpha})$, $\frac{\partial J_h}{\partial \alpha_j}$, $j \in \mathcal{D}$ and $\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}_h^1}$ that can be implemented efficiently within the IFE framework. In particular, we have $\nabla J_h = \mathbf{0}$. These preparations can then be employed in the proposed IFE Shape Optimization Algorithm together with the SQP (sequential quadratic programming) method [62] to carry out the constrained optimization numerically.

We test the proposed IFE method on a specific design problem configured in the domain Ω that contains a design independent heat source $f = -1$ on a center square $[-0.1, 0.1] \times [-0.1, 0.1]$, and the boundary temperature is fixed to be $u = 0$ and $\theta = 0.5$, see the illustration in Fig. 10. The two materials separated by the curve Γ are such that $\beta^- = 1$ and $\beta^+ = 10^{-3}$. We start the SQP iteration from a circle $x^2 + y^2 = 0.82^2$ plotted as a solid curve (in blue color) in Fig. 10a, and the numerical curve in the optimization is a parametric cubic spline with 20 control points. After 28 iterations, the algorithm generates a design shown in Fig. 10b whose patten is very similar to the one reported in [26].

Appendix A Technical Results

A.1 Proof of Lemma 3.1

First, differentiating $x_P = x(\hat{t}_P(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ and $y_P = y(\hat{t}_P(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ with respect to α_j and letting that $\frac{\partial x}{\partial \hat{t}_P} := \frac{\partial x}{\partial t}|_{t=\hat{t}_P}$ and $\frac{\partial y}{\partial \hat{t}_P} := \frac{\partial y}{\partial t}|_{t=\hat{t}_P}$, we have $D_{\alpha_j} x_P = \frac{\partial x}{\partial \hat{t}_P}\frac{\partial \hat{t}_P}{\partial \alpha_j} + \frac{\partial x}{\partial \alpha_j}|_{t=\hat{t}_P}$, $D_{\alpha_j} y_P = \frac{\partial y}{\partial \hat{t}_P}\frac{\partial \hat{t}_P}{\partial \alpha_j} + \frac{\partial y}{\partial \alpha_j}|_{t=\hat{t}_P}$, which leads to

$$\frac{\partial y}{\partial \hat{t}_P} D_{\alpha_j} x_P - \frac{\partial x}{\partial \hat{t}_P} D_{\alpha_j} y_P = \frac{\partial y}{\partial \hat{t}_P}\frac{\partial x}{\partial \alpha_j}|_{t=\hat{t}_P} - \frac{\partial x}{\partial \hat{t}_P}\frac{\partial y}{\partial \alpha_j}|_{t=\hat{t}_P}. \tag{A.1}$$

On the other hand, since P is on the edge A_1A_2, we have the equation $(y_2 - y_1)x_P - (x_2 - x_1)y_P = x_2y_1 - x_1y_2$. Differentiating it with respect to α_j yields

$$(y_2 - y_1) D_{\alpha_j} x_P - (x_2 - x_1) D_{\alpha_j} y_P = 0. \tag{A.2}$$

Combining (A.2) and (A.1) yields the linear system for $D_{\alpha_j} P$ in (3.8). Let $\mathbf{n}_e$ be the normal vector to the edge A_1A_2. Then we have $\det(M_P(\hat{t}_P)) = \mathbf{n}_e \cdot \nabla\Gamma(\hat{t}_P(\boldsymbol{\alpha}), \boldsymbol{\alpha})$ which is non zero by the assumption that A_1A_2 is not tangent to $\Gamma(t, \boldsymbol{\alpha})$ at P.

A.2 Material Derivatives of Local Matrices and Vectors

For the simplicity, we assume that the boundary condition functions g_N^k, g_D^k and the force term f^k are fixed and independent with interface change, $1 \leqslant k \leqslant K$. Therefore, on each interface element $T \in \mathcal{T}_h^{int}$ and each interface edge $e \in \mathcal{E}_h^{int}$, we have:

$$\begin{aligned} D_{\alpha_j}\mathbf{K}_T = &\left(\int_T \beta \nabla \frac{\partial \psi_{p,T}}{\partial \alpha_j} \cdot \nabla \psi_{q,T} dX\right)_{p,q\in\mathcal{I}} + \left(\int_T \beta \nabla \frac{\partial \psi_{p,T}}{\partial \alpha_j} \cdot \nabla \psi_{q,T} dX\right)^T_{p,q\in\mathcal{I}} \\ &+ \left(\sum_{m=1}^{3} \int_{T_m} \beta \nabla \psi_{p,T} \cdot \nabla \psi_{q,T}\, dX\, tr\left((D_{\alpha_j}\mathbf{J}_m)\mathbf{J}_m^{-1}\right)\right)_{p,q\in\mathcal{I}}, \end{aligned} \tag{A.3a}$$

$$\begin{aligned} D_{\alpha_j}\mathbf{E}_e^{r_1r_2} = &\left(\int_e \beta \nabla \frac{\partial \psi_{p,T^{r_1}}}{\partial \alpha_j} \cdot (\psi_{q,T^{r_2}} \mathbf{n}_e^{r_2}) ds\right)_{p,q\in\mathcal{I}} \\ &+ \left(\int_e \beta \nabla \psi_{p,T^{r_1}} \cdot \left(\frac{\partial \psi_{q,T^{r_2}}}{\partial \alpha_j}\mathbf{n}_e^{r_2}\right) ds\right)_{p,q\in\mathcal{I}} \\ &+ \left(\beta^- \nabla \psi^-_{p,T^{r_1}} \cdot (\psi^-_{q,T^{r_2}} \mathbf{n}_e^{r_2})|_P - \beta^+ \nabla \psi^+_{p,T^{r_1}} \cdot (\psi^+_{q,T^{r_2}} \mathbf{n}_e^{r_2})|_P\right)_{p,q\in\mathcal{I}} \\ &\frac{D_{\alpha_j} P \cdot (A_2 - A_1)}{\|A_2 - A_1\|}, \end{aligned} \tag{A.3b}$$

$$\begin{aligned} D_{\alpha_j}\mathbf{G}_e^{r_1r_2} = &\frac{\sigma_e^0}{|e|}\left(\int_e \left(\frac{\partial \psi_{p,T^{r_1}}}{\partial \alpha_j}\mathbf{n}_e^{r_1}\right) \cdot (\psi_{q,T^{r_2}} \mathbf{n}_e^{r_2}) ds \right. \\ &\left. + \int_e (\psi_{p,T^{r_1}} \mathbf{n}_e^{r_1}) \cdot \left(\frac{\partial \psi_{q,T^{r_2}}}{\partial \alpha_j}\mathbf{n}_e^{r_2}\right) ds\right)_{p,q\in\mathcal{I}}, \end{aligned} \tag{A.3c}$$

$$D_{\alpha_j}\mathbf{R}_T = \left(\int_T \frac{\partial \psi_{p,T}}{\partial \alpha_j} dX + \int_T \nabla \psi_{p,T} \cdot \mathbf{V}^j dX\right)_{p\in\mathcal{I}}$$

$$+\left(\sum_{i=1}^{3}\int_{T_i}\psi_{p,T}\,dX\;tr\left((D_{\alpha_j}\mathbf{J}_i)\mathbf{J}_i^{-1}\right)\right)_{p\in\mathcal{I}}. \tag{A.3d}$$

and the material derivatives of vectors:

$$D_{\alpha_j}\mathbf{F}_T^k=\left(\int_T f^k\frac{\partial\psi_{p,T}}{\partial\alpha_j}dX\right)_{p\in\mathcal{I}}+\left(\int_T\nabla(f^k\psi_{p,T})\cdot\mathbf{V}_T^j dX\right)_{p\in\mathcal{I}}$$
$$+\left(\sum_{m=1}^{3}\int_{T_m}f^k\psi_{p,T}\,dX\;tr\left((D_{\alpha_j}\mathbf{J}_m)\mathbf{J}_m^{-1}\right)\right)_{p\in\mathcal{I}}, \tag{A.4a}$$

$$D_{\partial_j}\mathbf{B}_e^k=\left(\int_e\beta g_D^k\nabla\frac{\partial\psi_{p,T}}{\partial\alpha_j}\cdot\mathbf{n}_e ds\right)_{p\in\mathcal{I}}$$
$$+\left(\beta^- g_D^k\nabla\psi_{p,T}^-\cdot\mathbf{n}_e|_P-\beta^+ g_D^k\nabla\psi_{p,T}^+\cdot\mathbf{n}_e|_P\right)_{p\in\mathcal{I}}\frac{D_{\alpha_j}P\cdot(A_2-A_1)}{\|A_2-A_1\|}, \tag{A.4b}$$

$$D_{\partial_j}\mathbf{C}_e^k=\frac{\sigma_e^0}{|e|}\left(\int_e\beta g_D^k\frac{\partial\psi_{p,T}}{\partial\alpha_j}ds\right)_{p\in\mathcal{I}}+\frac{\sigma_e^0}{|e|}\left(\beta^- g_D^k\psi_{p,T}^-|_P-\beta^+ g_D^k\psi_{p,T}^+|_P\right)_{p\in\mathcal{I}}$$
$$\frac{D_{\alpha_j}P\cdot(A_2-A_1)}{\|A_2-A_1\|}, \tag{A.4c}$$

$$D_{\partial_j}\mathbf{N}_e^k=\left(\int_e g_N^k\frac{\partial\psi_{p,T}}{\partial\alpha_j}ds\right)_{p\in\mathcal{I}}+\left(g_N^k\psi_{p,T}^-|_P-g_N^k\psi_{p,T}^+|_P\right)_{p\in\mathcal{I}}\frac{D_{\alpha_j}P\cdot(A_2-A_1)}{\|A_2-A_1\|}. \tag{A.4d}$$

References

1. Adjerid, S., Chaabane, N., Lin, T.: An immersed discontinuous finite element method for stokes interface problems. Comput. Methods Appl. Mech. Eng. **293**, 170–190 (2015)
2. Adjerid, S., Guo, R., Lin, T.: High degree immersed finite element spaces by a least squares method. Int. J. Numer. Anal. Model. **14**(4–5), 604–626 (2016)
3. Alessandrini, G., Isakov, V., Powell, J.: Local uniqueness in the inverse conductivity problem with one measurement. Trans. Am. Math. Soc. **347**(8), 3031–3041 (1995)
4. Allaire, G., Dapogny, C., Frey, P.: Shape optimization with a level set based mesh evolution method. Comput. Methods Appl. Mech. Eng. **282**, 22–53 (2014)
5. Allaire, G., Jouve, F., Toader, A.-M.: Structural optimization using sensitivity analysis and a level-set method. J. Comput. Phys. **194**(1), 363–393 (2004)
6. Attetkov, A.V., Volkov, I.K., Tverskaya, E.S.: The optimum thickness of a cooled coated wall exposed to local pulseperiodic heating. J. Eng. Phys. Thermophys. **74**(6), 1467–1474 (2001)
7. Babuška, I., Osborn, J.E.: Can a Finite element method perform arbitrarily badly? Math. Comput. **69**(230), 443–462 (2000)
8. Bai, J., Cao, Y., He, X., Liu, H., Yang, X.: Modeling and an immersed finite element method for an interface wave equation. Comput. Math. Appl. **76**(7), 1625–1638 (2018)
9. Bejan, A.: Constructal-theory network of conducting paths for cooling a heat generating volume. Int. J. Heat Mass Transf. **40**(4), 799813–811816 (1997)
10. Belhachmi, Z., Meftahi, H.: Shape sensitivity analysis for an interface problem via minimax differentiability. Appl. Math. Comput. **219**(12), 6828 (2013)
11. Bendsøe, M.P.: Optimization of Structural Topology, Shape, and Material. Springer, Berlin (1995)
12. Brenner, S.C., Ridgway Scott, L.: The Mathematical Theory of Finite Element Methods, Volume 15 of Texts Applied Mathematics, 3rd edn. Springer, New York (2008)
13. Burger, M., Osher, S.J.: A survey on level set methods for inverse problems and optimal design. Eur. J. Appl. Math. **16**(2), 263–301 (2005)

14. Calderón, A.P.: On an inverse boundary value problem. Comput. Appl. Math. **25**(2–3), 133–138 (2006)
15. Cantarero, A., Goldstein, T.: A fast method for interface and parameter estimation in linear elliptic PDES with piecewise constant coefficients (2013). ftp://ftp.math.ucla.edu/pub/camreport/cam11-77.pdf
16. Carpentieri, G., Koren, B., van Tooren, M.J.L.: Adjoint-based aerodynamic shape optimization on unstructured meshes. J. Comput. Phys. **224**(1), 267–287 (2007)
17. Chan, T.F., Tai, X.-C.: Identification of discontinuous coefficients in elliptic problems using total variation regularization. SIAM J. Sci. Comput. **25**(3), 881–904 (2003)
18. Chen, Z., Wu, Z., Xiao, Y.: An adaptive immersed finite element method with arbitrary lagrangian-eulerian scheme for parabolic equations in time variable domains. Int. J. Numer. Anal. Mod. **12**(3), 567–591 (2015)
19. Chen, Z., Zou, J.: Finite element methods and their convergence for elliptic and parabolic interface problems. Numer. Math. **79**(2), 175–202 (1998)
20. Chen, Z., Zou, J.: An augmented lagrangian method for identifying discontinuous parameters in elliptic systems. SIAM J. Control Optim. **37**(3), 892–910 (1999)
21. Choi, K.K., Chang, K.-H.: A study of design velocity field computation for shape optimal design. Finite Elem. Anal. Des. **15**(4), 317–341 (1994)
22. Chow, S., Anderssen, R.S.: Determination of the transmissivity zonation using a linear functional strateg. Inverse Probl. **7**, 841 (1991)
23. Dennis, J.E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Volume 16 of Classics Applied Mathematics. SIAM, Philadelphia (1996)
24. Dunning, P.D., Kim, H.A., Mullineux, G.: Investigation and improvement of sensitivity computation using the area-fraction weighted fixed grid fem and structural optimization. Finite Elem. Anal. Des. **47**(8), 933–941 (2011)
25. Ewing, R.E.: Society for Industrial, and Applied Mathematics. The Mathematics of Reservoir Simulation, Volume 1. SIAM, Philadelphia (1983)
26. Gao, T., Zhang, W.H., Zhu, J.H., Xu, Y.J., Bassir, D.H.: Topology optimization of heat conduction problem involving design-dependent heat load effect. Finite Elem. Anal. Des. **44**(14), 805–813 (2008)
27. Gautschi, W.: Numerical Analysis, 2nd edn. Springer/Birkhäuser, New York (2012)
28. Gersborg-Hansen, A., Bendsøe, M.P., Sigmund, O.: Topology optimization of heat conduction problems using the finite volume method. Struct. Multidiscip. Optim. **31**(4), 251–259 (2006)
29. Giles, M.B., Pierce, N.A.: An introduction to the adjoint approach to design. Flow Turbul. Combust. **65**(3), 393–415 (2000)
30. Gockenbach, M.S., Khan, A.A.: An abstract framework for elliptic inverse problems: Part 2. An augmented Lagrangian approach. Math. Mech. Solids **14**(6):517–539 (2009;2008)
31. Guo, R., Lin, T.: A group of immersed finite element spaces for elliptic interface problems. IMA J. Numer. Anal. (2017). https://doi.org/10.1093/imanum/drx074
32. Guo, R., Lin, T., Zhang, X.: Nonconforming immersed finite element spaces for elliptic interface problems. Comput. Math. Appl. **75**(6), 2002–2016 (2018)
33. Harbrecht, H., Tausch, J.: On the numerical solution of a shape optimization problem for the heat equation. SIAM J. Sci. Comput. **35**(1), A.104–A121 (2013)
34. Haslinger, J., Mäkinen, R.A.E.: Introduction to Shape Optimization: Theory, Approximation, and Computation. SIAM, Society for Industrial and Applied Mathematics, Philadelphia (2003)
35. He, X., Lin, T., Lin, Y.: Approximation capability of a bilinear immersed finite element space. Numer. Methods Partial Differ. Equ. **24**(5), 1265–1300 (2008)
36. He, X., Lin, T., Lin, Y.: Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions. Int. J. Numer. Anal. Model. **8**(2), 284–301 (2011)
37. Hegemann, J., Cantarero, A., Richardson, C.L., Teran, J.M.: An explicit update scheme for inverse parameter and interface estimation of piecewise constant coefficients in linear elliptic PDES. SIAM J. Sci. Comput. **35**(2), A1098–A1119 (2013)
38. Hogea, C., Davatzikos, C., Biros, G.: An image-driven parameter estimation problem for a reaction-diffusion glioma growth model with mass effects. J. Math. Biol. **56**(6), 793–825 (2008)
39. Holder, D., Institute of Physics (Great Britain).: Electrical Impedance Tomography: Methods, History, and Applications. Institute of Physics Pub, Bristol (2005)
40. Huang, X., Xie, Y.M.: Evolutionary Topology Optimization of Continuum Structures: Methods and Applications. Wiley, Hoboken (2010)
41. Ito, K., Kunisch, K.: The augmented lagrangian method for parameter estimation in elliptic systems. SIAM J. Control Optim. **28**(1), 113–136 (1990)
42. Ito, K., Kunisch, K., Li, Z.: Level-set function approach to an inverse interface problem. Inverse Probl. **17**, 1225 (2001)
43. Jang, G.-W., Kim, Y.Y.: Sensitivity analysis for fixed-grid shape optimization by using oblique boundary curve approximation. Int. J. Solids Struct. **42**(11), 3591–3609 (2005)

44. Ji, L., McLaughlin, J.R., Renzi, D., Yoon, J.-R.: Interior elastodynamics inverse problems: shear wave speed reconstruction in transient elastography. Inverse Probl. **19**(6), S1–S29 (2003)
45. Kim, H., Querin, O.M., Steven, G.P., Xie, Y.M.: Improving efficiency of evolutionary structural optimization by implementing fixed grid mesh. Struct. Multidiscip. Optim. **24**(6), 441–448 (2002)
46. Kim, N.H., Chang, Y.: Eulerian shape design sensitivity analysis and optimization with a fixed grid. Comput. Methods Appl. Mech. Eng. **194**(30), 3291–3314 (2005)
47. Kohn, R.V., Vogelius, M.: Relaxation of a variational method for impedance computed tomography. Commun. Pure Appl. Anal. **40**(6), 745–777 (1987)
48. Lee, H.S., Park, C.J., Park, H.W.: Identification of geometric shapes and material properties of inclusions in two-dimensional finite bodies by boundary parameterization. Comput. Methods Appl. Mech. Eng. **181**(1), 1–20 (2000)
49. Li, Q., Steven, G.P., Xie, Y.M., Querin, O.M.: Evolutionary topology optimization for temperature reduction of heat conducting fields. Int. J. Heat Mass Transf. **47**(23), 5071–5083 (2004)
50. Li, Z., Lin, T., Lin, Y., Rogers, R.C.: An immersed finite element space and its approximation capability. Numer. Methods Partial Differ. Equ. **20**(3), 338–367 (2004)
51. Li, Z., Lin, T., Xiaohui, W.: New Cartesian grid methods for interface problems using the finite element formulation. Numer. Math. **96**(1), 61–98 (2003)
52. Lin, M., Lin, T., Zhang, H.: Error analysis of an immersed finite element method for euler-bernoulli beam interface problems. Int. J. Numer. Anal. Model. **14**, 822–841 (2017)
53. Lin, T., Lin, Y., Rogers, R., Lynne Ryan, M.: A rectangular immersed finite element space for interface problems. In: Scientific Computing and Applications (Kananaskis, AB, 2000), Volume 7 of Advanced Computer Theory Practice, pp. 107–114. Nova Science Publishers, Huntington (2001)
54. Lin, T., Lin, Y., Zhang, X.: Partially penalized immersed finite element methods for elliptic interface problems. SIAM J. Numer. Anal. **53**(2), 1121–1144 (2015)
55. Lin, T., Zhang, X.: Linear and bilinear immersed finite elements for planar elasticity interface problems. J. Comput. Appl. Math. **236**(18), 4681–4699 (2012)
56. Lionheart, W.R.B.: Boundary shape and electrical impedance tomography. Inverse Probl. **14**, 139 (1998)
57. Luo, Z., Wang, M.Y., Wang, S., Wei, P.: A level set-based parameterization method for structural shape and topology optimization. Int. J. Numer. Methods Eng. **76**(1), 1–26 (2008)
58. McLaughlin, J.R., Zhang, N., Manduca, A.: Calculating tissue shear modulus and pressure by 2D log-elastographic methods. Inverse Probl. **26**(8), 085007 (2010)
59. Mohammadi, B., Pironneau, O.: Shape optimization in fluid mechanics. Annu. Rev. Fluid Mech. **36**(1), 255–279 (2004)
60. Najafi, A.R., Safdari, M., Tortorelli, D.A., Geubelle, P.H.: A gradient-based shape optimization scheme using an interface-enriched generalized FEM. Comput. Methods Appl. Mech. Eng. **296**, 1–17 (2015)
61. Nanthakumar, S.S., Lahmer, T., Rabczuk, T.: Detection of flaws in piezoelectric structures using extended fem. Int. J. Numer. Methods Eng. **96**(6), 373–389 (2013)
62. Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Research, second edn. Springer, Berlin (2006)
63. Novotny, A.A., Canelas, A., Laurain, A.: A non-iterative method for the inverse potential problem based on the topological derivative. In: Hintermüller, M., Leugering, G., Sokołowski, J. (eds.) Technical Report for Mini-Workshop: Geometries, Shapes and Topologies in PDE-based Applications, 57/2012, pp. 3383–3387. Mathematisches Forschungsinstitut Oberwolfach, Oberwolfach (2012)
64. Novotny, A.A., Sokołowski, J.: Topological Derivatives in Shape Optimization. Springer, Heidelberg (2013)
65. Peng, X., Niakhai, K., Protas, B.: A method for geometry optimization in a simple model of two-dimensional heat transfer. SIAM J. Sci. Comput. **35**(5), B.1105–B1131 (2013)
66. Perego, M., Veneziani, A., Vergara, C.: 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)
67. Rabinovich, D., Givoli, D., Vigdergauz, S.: XFEM-based crack detection scheme using a genetic algorithm. Int. J. Numer. Methods Eng. **71**(9), 1051–1080 (2007)
68. Ródenas, J.J., Fuenmayor, F.J., Tarancón, J.E.: A numerical methodology to assess the quality of the design velocity field computation methods in shape sensitivity analysis. Int. J. Numer. Methods Eng. **59**(13), 1725–1747 (2004)
69. Sasikumar, M., Balaji, C.: Optimization of convective fin systems: a holistic approach. Heat Mass Transf. **39**(1), 57–68 (2002)
70. Sattinger, D.H., Tracy, C.A., Venakides, S.: Inverse Scattering and Applications, Volume 122. American Mathematical Society, Providence (1991)

71. Schnur, D.S., Zabaras, N.: An inverse method for determining elastic material properties and a material interface. Int. J. Numer. Methods Eng. **33**(10), 2039–2057 (1992)
72. Soghrati, S., Aragón, A.M., Armando Duarte, C., Geubelle, P.H.: An interface-enriched generalized fem for problems with discontinuous gradient fields. Int. J. Numer. Methods Eng. **89**(8), 991–1008 (2012)
73. Suzuki, K., Kikuchi, N.: A homogenization method for shape and topology optimization. Comput. Methods Appl. Mech. Eng. **93**(3), 291–318 (1991)
74. van Keulen, F., Haftka, R.T., Kim, N.H.: Review of options for structural design sensitivity analysis. Part 1: linear systems. Comput. Methods Appl. Mech. Eng. **194**(30), 3213–3243 (2005)
75. Waisman, H., Chatzi, E., Smyth, A.W.: Detection and quantification of flaws in structures by the extended finite element method and genetic algorithms. Int. J. Numer. Methods Eng. **82**, 303–328 (2010)
76. Wei, P., Wang, M.Y., Xing, X.: A study on X-FEM in continuum structural optimization using a level set model. Comput. Aided Des. **42**(8), 708–719 (2010)
77. Yeh, W.W.: Review of parameter identification procedures in groundwater hydrology: the inverse problem. Water Resour. Res. **22**(2), 95–108 (1986)
78. Zhang, H., Lin, T., Lin, Y.: Linear and quadratic immersed finite element methods for the multi-layer porous wall model for coronary drug-eluting stents. Int. J. Numer. Anal. Mod. **15**, 48–73 (2018)
79. Zhang, J., Zhang, W.H., Zhu, J.H., Xia, L.: Integrated layout design of multi-component systems using XFEM and analytical sensitivity analysis. Comput. Methods Appl. Mech. Eng. **245–246**, 75–89 (2012)
80. Zhang, X.: Nonconforming Immersed Finite Element Methods for Interface Problems. Ph.D. Thesis, Virginia Polytechnic Institute and State University (2013)
81. Zhang, Y., Liu, S., Qiao, H.: Design of the heat conduction structure based on the topology optimization. Developments in Heat Transfer, Chap. 26. IntechOpen (2011)