

Isogeometric analysis for solving discontinuous two-phase engineering problems with precise and explicit interface representation

Emad Shakur

*Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences,
Altenbergerstr. 69, 4040 Linz, Austria*

Keywords: Isogeometric analysis, Untrimming, Unstructured spline-based meshes, Precise interface representation, Two-phase engineering problems.

Abstract

This paper presents a new computational method for solving two-phase engineering problems. The novelty of this approach lies in its ability to perform the analysis over the exact two-phase geometry. The geometry is defined by a spline-based representation and constructed based on the level-set method. The level-set-based geometry is mapped into a spline-based geometry using an untrimming technique that is newly developed in this paper. The spline-based representation is defined over unstructured meshes enabling precise representations of geometrically complex domains. An analysis-suitable mechanical model that accurately replicates the geometry is constructed and the governing equations are solved using the isogeometric analysis method. The interface between phases is explicitly represented by cubic B-spline curves, allowing for the evaluation of the solution field and various physical measures over its exact geometry. Additionally, algorithms that allow for changing the continuity of the solution field between the phases while precisely preserving the discontinuous two-phase domain are developed. Consistent analytical sensitivity analysis is formulated for generating the geometrical and mechanical models to allow for solving engineering problems with moving boundaries following the gradient-based approach. The proposed approach is tested for solving Poisson's equation, linear elasticity, topology optimization, and two-phase flow problems. The results expose the capability of the proposed approach to solve two-phase engineering problems while accurately representing the interface.

1. Introduction

Various engineering problems involve multiple material phases to describe their physical behavior. Examples include elasticity problems with multiple material properties, contact between multiple objects, and two-phase flow problems. In some cases, a static interface exists between the material phases, such as in steady-state fluid-structure interaction and static linear elasticity. Conversely, the shape and topology of the interface can be dynamic variables as in topology

Email address: emad.shakur@ricam.oeaw.ac.at (Emad Shakur)

optimization and dynamic fluid mechanics. Multiphase engineering problems are typically sensitive to the quality of representing the interface between the different phases. An inaccurate representation of the interface can lead to undesired inaccuracies in the solution of the physical problem.

While the geometrical models for representing the multiphase domains are typically precisely defined by using phase-field [1, 2] or level-set-based methods [3, 4], the models that are used to numerically solve the physical problem – the mechanical models – are constructed using various approximation techniques. The key difference between these techniques lies in dealing with the interface representation in the mechanical model. With the advancement of the finite element analysis approach, various methods have been developed aiming to increase the quality of representing the interface in the mechanical model. These methods could be distinguished as diffusive interface methods and sharp interface methods.

In diffusive interface methods, the exact interface is smeared to constitute a transition region of non-zero width that separates the domains of the different phases. The transition region is typically smooth and considered an intermediate phase that averages the physical properties of the adjacent phases. In two-phase flow problems, the intermediate phase is occasionally interpreted as an artificial mixing between the two fluids [5], whereas, in other multiphase engineering problems, it is considered as an approximation of the actual physical behavior. Following this class of methods, interactions between two phases, such as surface tension in two-phase flow problems, are usually formulated over the smeared interface region and not along the actual interface geometry. Diffusive interface methods have been used in various multiphase engineering problems such as topology optimization [6, 7], two-phase flow problems [8] and crack-propagation [9]. Despite the popularity and computational efficiency of diffusive interface methods in solving various engineering problems, the interfaces according to these methods are not well defined, making it challenging to evaluate physical and geometrical measures in proximity to the interface regions.

In contrast, sharp interface methods aim for a crisp separation between the different material phases in the mechanical model. Sharp interface methods include the major class of immersed boundary techniques. A well-known immersed boundary techniques are the extended finite element method (X-FEM) [10, 11] and the CutFEM methods [12, 13, 14, 15]. In these methods, elements along the crossing interface are decomposed into sub-domains representing the different material phases. The decomposition is performed by enriching the finite element shape functions in the vicinity of the interface using projection functions. X-FEM and CutFEM have been used to solve various multiphase engineering problems including two-phase flow [16, 17], contact [18], and multiphase linear elasticity problems [19, 20, 21]. In other immersed boundary techniques, the finite elements mesh can be constantly modified to approximate the exact geometrical interface, such as in arbitrary Lagrangian-Eulerian methods [22].

A main difference between the various methods for constructing mechanical models in multiphase engineering problems is their ability to approximate the actual geometry of the interface. The quality of the approximation impacts the accuracy of evaluating physical measures near

interfaces, such as interactions between the phases, and consequently, it might influence the accuracy of the analysis results. In general, inconsistencies between the mechanical model and the exact geometry emanate from two main reasons. The first reason is that the interfaces in the geometrical models are typically implicitly defined, making them hard to replicate in the mechanical model. The second reason is the difficulty of the traditionally adopted finite element method to precisely describe free-formed curved interfaces. In principle, the primary objective of the current paper is to solve this problem by developing a methodology characterized by a unified representation of the geometrical and mechanical models to eliminate inconsistencies between them. Additionally, to develop a representation that has an explicit and precise description of the interfaces between the phases to allow for more accurate evaluations of physical and geometrical measures in the proximity of interface regions.

The need to eliminate inconsistencies between the geometrical and mechanical models is not new, while several approaches have been developed aiming for a unified representation for both models [23, 24] leading to the development of isogeometric analysis (IGA) [25, 26]. The basic idea of IGA is that it adopts a spline-based basis to represent the mechanical model aiming to match the spline-based representation that is commonly used to represent geometrical models in Computer-Aided Design systems (CAD). This approach enables reaching a unified and explicit representation for the geometrical and mechanical models, rendering it an attractive choice for addressing multiphase engineering problems that require accurate interface representation.

One of the main challenges in solving two-phase engineering problems is the inclusion of the trimmed patches at the interface regions in the mechanical model. Numerous studies have been conducted addressing this challenge using IGA, aiming to increase the accuracy of the analysis results. Kim et al. [27] introduced a method that deals with trimmed patches in B-spline surfaces. Their method involves replacing the trimmed patches with triangular patches, each of which incorporates curved edges that accurately replicate given B-spline trimming curves. Various methods have been developed with the aim of improving the accuracy of physical results near trimmed patches by developing specific quadrature rules [28, 29, 30]. Wei et al. [31] introduces an approach that combines immersed boundary techniques with conformal discretization near the interface aiming for more accurate physical responses. Lieu and Lee [32] solved multi-material topology optimization problems constructed over a B-spline domain, employing k-refinement techniques to enhance mesh resolution near the interface. Hosseini et al. [33] adopted IGA to solve two-phase flow problems where the geometry of the fluids is governed by the Cahn–Hilliard phase-field approach. In Yan et al. [34] high-order nonuniform rational B-splines (NURBS) were used to model multiphase flow problems with geometry based on the level-set method.

IGA has been integrated with enrichment techniques to the basis functions to make physical results near interfaces more accurate. This led to the development of the Extended Isogeometric Analysis (X-IGA). Noël et al. [35] introduced a new generalized Heaviside enrichment strategy that addresses numerical instabilities that characterize the X-IGA approach to solving problems over domains that accommodate an arbitrary number of materials. This paper’s concept has been further enhanced in Schmidt et al. [36] by utilizing the refinability of spline-based representations,

resulting in an X-IGA procedure that is based on high-order hierarchical B-spline domains. It is worth noting that high-order hierarchical B-spline representation was previously used by Noël et al. [37] for geometric design. Additionally, X-IGA has been successfully adopted to solve fracture propagation [38, 39] and topology optimization problems [40, 41].

Most of the conducted research that uses IGA for solving multiphase engineering problems adopts implicit representation for the geometrical design and structured meshes with NURBS to represent the mechanical model. This obstructs the achievement of a unified representation for both the geometrical and mechanical models. While NURBS offer high polynomial degrees and continuity compared to finite element methods, their structured configuration limits their ability to accurately represent geometrically complex domains. This requires using projection procedures or other approximation techniques in order to capture the physical behavior near the interface between the material phases. Other research addresses this issue by using local refinement techniques to increase mesh resolution near the interface. This indicates that the full potential of IGA for integration over exact domains of complex geometries has yet to be realized. To take a step toward achieving this goal, the approach in the current paper adopts an explicit spline-based representation of the geometry, enabling a precise replication of the mechanical model. Nowadays, the topic of performing IGA over exact geometrically complex domains is on active research with a focus on domains that are defined over unstructured meshes [42, 43, 44].

A recent paper by the author introduces a new approach that uses IGA on unstructured meshes to solve single-phase topology optimization problems [45]. The main advantage of this approach is the adoption of untrimming techniques to generate unstructured meshes that can represent complex geometries in a precise and explicit manner. Additionally, the approach utilizes the ability of IGA to perform integrations over the exact domain of the geometry. This approach was used to solve stress-constrained topology optimization problems in [46]. The work in [46] emphasizes the importance of precisely representing the boundaries of the mechanical model in engineering problems where the physical behavior near the boundaries is crucial. However, this approach is developed mainly to solve single-phase linear elasticity problems and topology optimization and it is incapable of representing multiphase domains.

Inspired by the work in [45], in this paper, a new method for solving two-phase engineering problems is developed. The novelty of this method with respect to current research on solving two-phase engineering problems is that it allows for a precise and explicit spline-based representation for both the geometrical and the mechanical models. The main idea of this method is that it considers the implicit level-set-based representation as an intermediate step rather than a final representation of the geometry. Instead, the final geometry is defined by a spline-based representation, constructed using a linear mapping procedure that is applied to the level-set-based representation. The resulting spline-based representation possesses several analysis advantages and desired geometrical features:

1. Crisp phase separation: The proposed method utilizes a new untrimming technique that allows for a crisp separation between the phases in both the geometrical and mechanical models.

2. Accurate interface representation: The interface between the phases is accurately represented as a set of cubic B-spline curves in both the geometrical and the mechanical models. This precise representation enables the evaluation of various physical and geometrical measures along the exact interface, including curvature, tangent vector, and values of the solution field, while also ensuring compatibility with CAD systems.
3. Conforming mesh: This approach provides a conforming mesh between different phases, eliminating the requirement for special treatments to achieve a tight bond between phases.
4. Precise geometry replication by the mechanical model: The mechanical model according to this approach precisely replicates the physical domain of the geometry, eliminating the need for approximation procedures between the two models.
5. Controlled continuity: The approach provides control over the continuity between different phases in the solution field without affecting the interface's geometry. This feature is particularly useful when solving multi-physics engineering problems where multiple partial differential equations (PDEs) with varying continuity requirements within the same two-phase domain are solved.
6. Analytical sensitivity analysis: All stages of this method are differentiable, allowing for the development of a consistent and analytical sensitivity analysis for the entire solving procedure. This facilitates the integration of the proposed approach into engineering problems involving moving interfaces.

Additionally, it is worth noting that this approach relies on the level-set method for the generation of the two-phase domain, allowing it to efficiently handle topological changes that may occur in engineering problems with moving interfaces. To the best of the author's knowledge, these advantages have not yet been integrated into a single procedure in the literature for solving two-phase engineering problems.

The remainder of this paper is organized as follows: Section 2 presents the procedure for generating the geometrical and mechanical models. Algorithms for changing the continuity of the solution field between the phases are given in Section 3. The sensitivity analysis for generating the geometrical and mechanical models is developed in Section 4. The ability of the proposed approach to solve two-phase engineering problems is demonstrated in Section 5. A discussion and concluding remarks are given in Sections 6 and 7, respectively. Finally, Appendices A and B are attached to the paper to provide further insights into the proposed approach.

2. Generating the geometrical and mechanical models

This section outlines the proposed procedure for generating the geometrical and mechanical models for solving two-phase engineering problems. The procedure assumes a set of given trimming curves, ζ , that are used to construct the interfaces in the geometrical model. The trimming curves in the current paper are generated following the level-set method, assuming a bi-cubic

B-spline surface in $\mathbb{R}^3$ as the level-set function. The z-coordinates of the B-spline surface served as control parameters of the level-set function and are denoted by α . It should be noted that the curves in ζ do not directly define the interfaces in the geometrical model. Instead, they are mapped into cubic B-spline curves that define the interfaces. This mapping is performed using untrimming techniques, resulting in a spline-based model defined over an unstructured mesh.

The procedure for generating the geometrical and mechanical models will be demonstrated using the example in Figures 1, 2, and 5. Figure 1 displays the level-set function that is used for the illustrative example. The zero contour line of the level-set function is considered as the trimming curves ζ . The level-set function is defined by a B-spline surface constructed over a mesh of 11×11 control points and open uniform knots vectors.

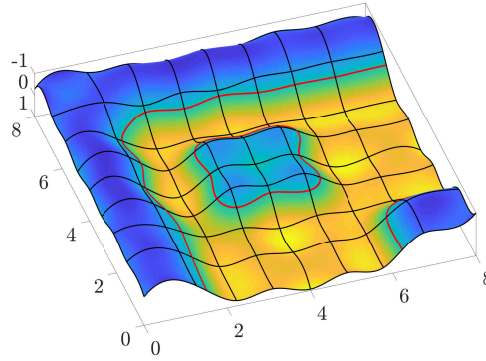


Figure 1: The level-set function that is used to demonstrate the procedure for generating the geometrical and mechanical models. The zero contour level of the level-set function is marked with red lines.

2.1. Generating the geometrical model

To generate the geometrical model using the given trimming curves, ζ , the following steps are performed:

- Step 1.** Define a bi-cubic B-spline surface, S^0 , that covers the entire physical domain of the problem. S^0 is considered as an initial geometry for the untrimming procedure. The knots vector and the planar locations of the control points of S^0 are chosen to be equal to those that define the level-set function. Figure 2a shows the initial B-spline surface, while Figure 2b shows its control polygon with black and red lines representing unit and zero knot intervals, respectively.
- Step 2.** Draw the curves in ζ on top of S^0 and its control polygon. Classify the trimmed faces in the control polygon according to the trimming manner, as shown in the first column of Table 1. The trimming manner for each patch is determined based on the values of the level-set function at its corner points. These trimming manners can be represented as cases 1, 2, and 3 in the table. Additionally, case 4 is introduced to handle zero-measure faces in the control polygons and by that to ensure proper Bézier end conditions for the trimmed patches along the boundary of S^0 . Note that each case in the table can be rotated by 90 degrees to cover all the possible trimming cases. While cases 1, 2,

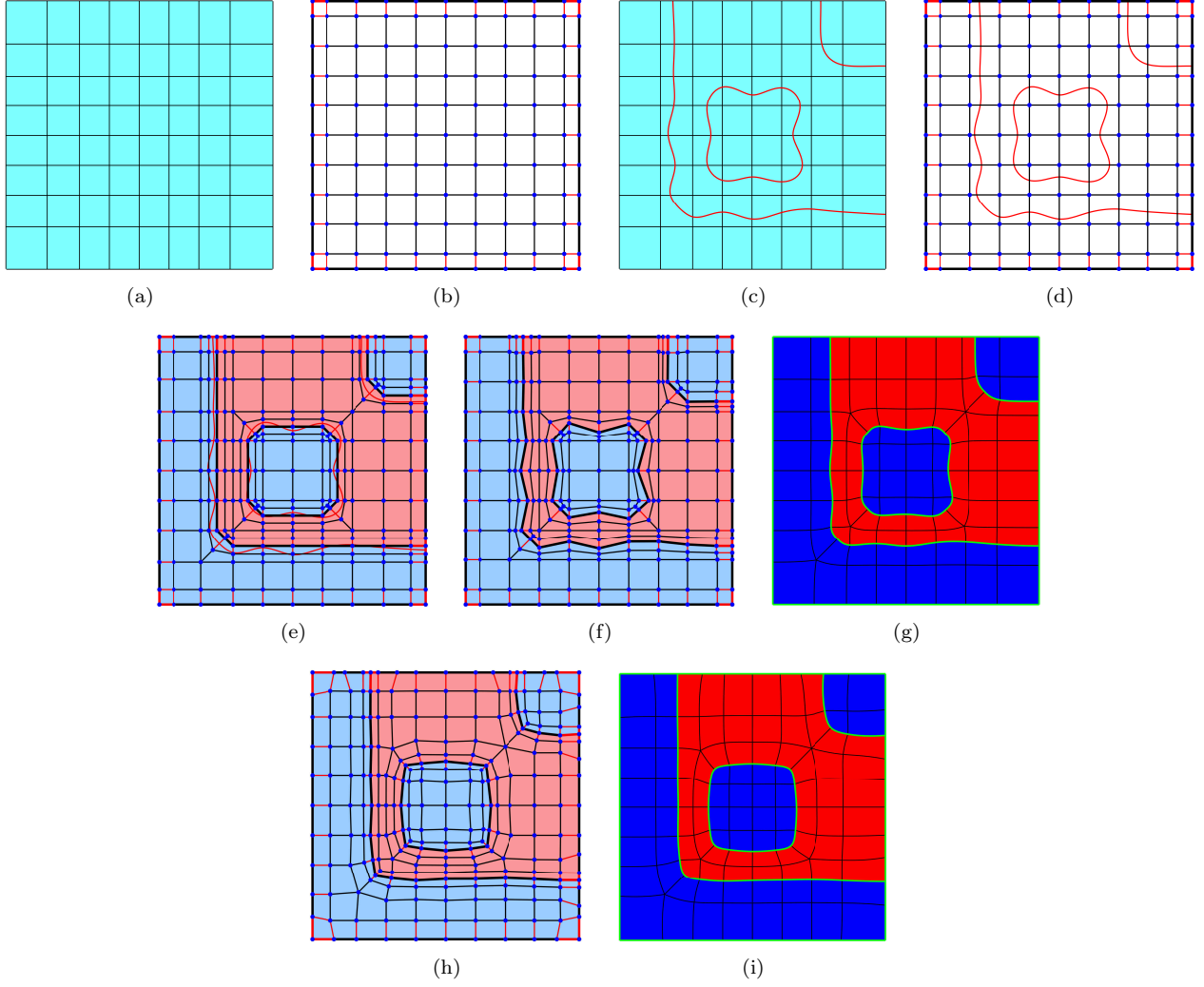


Figure 2: The steps involved in generating the geometrical model: (a) A bi-cubic B-spline surface S^0 is initially created. (b) The control polygon that corresponds to S^0 . (c) The curves in ζ drawn on top of S^0 . (d) The curves in ζ drawn on top of the control polygon in (b). (e) The unstructured control polygon after modifying the trimmed faces according to the trimming manner presented in Table 1. (f) The control polygon after computing the locations of the control points $\mathbf{P}^G$. (g) The physical domain corresponds to the control polygon in (f). (h) The control polygon after performing Laplacian smoothing. (i) The physical domain corresponds to the control polygon in (h).

and 4 can be directly determined by the values of the level-set function at the corners of the Bézier patches, case 3 requires additional knowledge of the contour path inside the domain. For the same level-set function values at the corners associated with case 3, two possible different configurations exist: the first is the one presented in the third row of the second column in the table, and the second is its rotation by 90 degrees. The correct configuration is determined by analyzing the contour path within the domain using square marching algorithms. For further details on this topic, readers are referred to the works in [47, 48]. Figures 2c and 2d demonstrate the curves in ζ drawn on top of S^0 and its control polygon, respectively.

Step 3. Replace all the trimmed faces according to their corresponding trimming manner as illustrated in the second column in Table 1. The blue dots in the same column represent the control points of the control polygon after the untrimming procedure. The filled dots

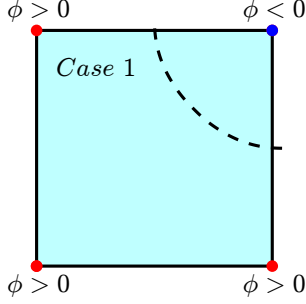
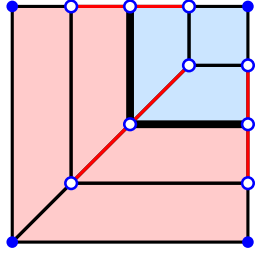
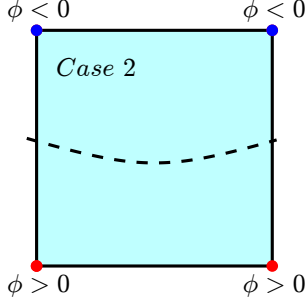
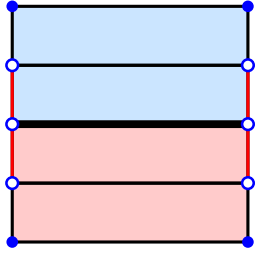
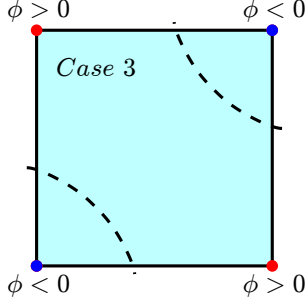
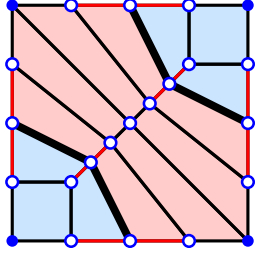
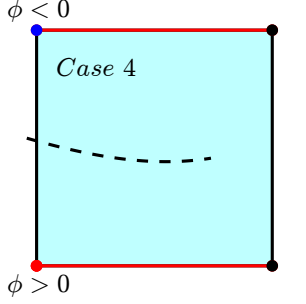
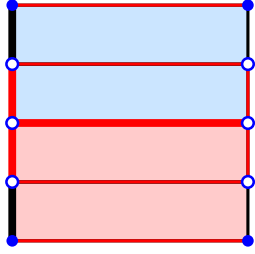
Trimming manner	Replacement
 Case 1	
 Case 2	
 Case 3	
 Case 4	

Table 1: Modifying the trimmed faces in the control polygon of S^0 according to the trimming manner. First column: trimmed faces with the possible contours configurations drawn in dashed lines. Second column: the corresponding modifications that should be applied to the trimmed faces.

represent control points already existing in S^0 , whereas the hollow dots represent newly added control points. The positions of the new control points are computed according to the subsequent step. The thick lines indicate the boundary edges of the different phases. The newly added edges that intersect boundary edges are assigned with zero knot intervals while all the other edges are assigned with unit knot intervals. In the table, black and red lines mark unit and zero knot intervals, respectively. The control polygon after this step is shown in Figure 2e. Note that the mesh after this step becomes unstructured due to the existence of extraordinary points, which are interior vertices in the control polygon that are not T-junctions and the number of their connected edges

is not 4.

Step 4. Compute the control points of the untrimmed geometry with respect to the control parameters α . It is important to note that the computation is not unique and can be chosen freely by the user. However, it is recommended to choose a smooth and differentiable computation method to facilitate the evaluation of sensitivities when dealing with problems with moving interfaces. We denote the control points of the mesh after this step by $\mathbf{P}^G$. These control points are a function of the control parameters of ζ , i.e. $\mathbf{P}^G(\alpha)$. A suitable computation procedure for positioning the newly added control points is provided in Appendix A. This Appendix refers to the 4 cases presented in Table 1. The relation between $\mathbf{P}^G$ and α in the provided computation is linear and can be represented by:

$$\mathbf{P}^G = \mathcal{A}\alpha, \quad (1)$$

where $\mathcal{A}$ is the correlation matrix. This computation procedure served to compute $\mathbf{P}^G$ for the illustrative example in this section and the examples in Section 5. The resulting control polygon after this step is displayed in Figure 2f, while the corresponding physical domain is shown in Figure 2g.

Step 5. Perform Laplacian smoothing to the control points in the domain. This step is highly beneficial for avoiding interface layers with very small-sized patches, as they could affect the conditionality of the linear system. Additionally, implementing this step leads to designs with reduced interface curvature. This is particularly advantageous in applications sensitive to high curvature values along the interface, such as problems related to stress concentrations. While this step successfully addresses these issues in the examples in the current paper, further in-depth studies on this matter are encouraged.

The modified location of a certain control point $\mathbf{P}_i^G$ after one Laplacian smoothing iteration is obtained by:

$$\mathbf{P}_i^{G\ell} = \frac{\sum_{j=1}^{N_{lp}} w_j \mathbf{P}_j^G}{\sum_{j=1}^{N_{lp}} w_j} \quad (2)$$

where $\mathbf{P}_i^{G\ell}$ is the modified location of $\mathbf{P}_i^G$; $\mathbf{P}_j^G$ represents a control point that is connected to $\mathbf{P}_i^G$; N_{lp} is the number of control points that are connected to $\mathbf{P}_i^G$; and w_j is the Laplacian smoothing weight that is associated with control point $\mathbf{P}_j^G$. All the weights are set to 1 at this step. Corner control points that have two zero-measure edges are excluded from the Laplacian smoothing procedure and their locations are not modified. The resulting control polygon and its corresponding physical domain after a single Laplacian smoothing iteration are displayed in Figures 2h and 2i, respectively. The modified control points $\mathbf{P}^{G\ell}$ are linearly correlated with the control points before

Laplacian smoothing, $\mathbf{P}^G$. This correlation can be expressed as:

$$\mathbf{P}^{G\ell} = \mathcal{L}^G \mathbf{P}^G \quad (3)$$

where $\mathcal{L}^G$ is the correlation matrix.

The physical domains corresponding to the unstructured meshes in the present work are constructed using the Bézier extraction procedure described in [49]. We note that the proposed approach distinguishes itself from other level-set-based methods in that the actual geometry of the two-phase domain is defined by the geometrical model presented in Figure 2g (or Figure 2i if Step 5 is applied), rather than the level-set-based geometry whose boundaries correspond to the zero contour level of the level-set function. The level-set-based geometry following the presented approach serves only as a preliminary stage in constructing the spline-based geometry. Meanwhile, the primary objective of adopting the level-set method is to govern the topological changes that may occur in the two-phase domain when solving engineering problems with moving interfaces. Thus, it is worth clarifying that the geometries depicted in Figures 2g and 2i are neither an approximation of the level-set-based topology nor a precise replication of it, rather, they are the outcome of the mapping procedure introduced in this section. The main aim of this mapping procedure is to generate geometrical models with topologies that follow the ones of the level-set models not to replicate or approximate them. It should be noted, that the suggested mapping procedure is differentiable with derivatives presented in Section 4. These derivatives are taken into account when solving the examples in Section 5.

The continuity of the spline-based model is derived from the adopted Bézier extraction procedure. According to the Bézier extraction procedure outlined in [49], the continuity between Bézier patches that belong to the same phase and are at least two patches away from an extraordinary point is C^2 , while the continuity between patches in the proximity of an extraordinary point can be either C^0 or C^1 . Increased continuity between patches in the proximity of extraordinary points can be achieved by applying smoothing algorithms, such as in [43]. Moreover, the continuity between the different phases is determined by the values of the knot intervals that separate them. According to the knot intervals in Table 1, the different phases are separated with two layers of zero knot intervals. These values result in a continuity of C^0 between the different phases, i.e., in the direction perpendicular to the interface. Nevertheless, the continuity between the phases can be adjusted to C^{-1} , C^1 or C^2 according to the requirements of the physical problem as will be discussed in Section 3.

An advantage of the proposed approach is the ability to represent interfaces between phases as smooth cubic B-spline curves, enabling precise evaluation of physical measures at exact points along the interfaces. Examples of these measures include stresses, pressure, and velocities, which may vary depending on the engineering problem that is solved. The procedure for extracting the interfaces from the unstructured control polygon is elaborated in Appendix B. The interfaces in the geometrical model are highlighted in green lines in Figures 2g and 2i.

The geometrical model generated so far is not necessarily analysis suitable. As discussed in [49] and [43], extraordinary points should be at least two patches away from each other to

avoid linear dependencies between the basis functions. This condition is not guaranteed in the geometrical model following the proposed approach. For this purpose, a mechanical model that ensures the analysis suitability of the mesh is constructed. It should be emphasized that the shape and topology of the B-spline curves that represent the interfaces are precisely preserved in the mechanical model.

2.2. Generating the mechanical model

An analysis suitable mechanical model is generated by applying refinement and Laplacian smoothing procedures to the geometrical model. These procedures result in a mechanical model that is constructed over an unstructured mesh which includes T-junctions. The basis functions corresponding to the T-junctions' vertices are evaluated following the theory of T-splines [50], while Bézier extraction to the patches near T-junctions is performed following the Bézier extraction algorithm in [51]. To generate the mechanical model, apply the following steps:

- Step 1.** Perform h-refinement to certain faces in the control polygon to ensure the analysis suitability of the mesh. The refinement is performed primarily to avoid linear dependencies between the basis functions. The refinement is performed as follows:
- All faces that touch an extraordinary point are subjected to 2 h-refinement levels. This ensures that the extraordinary points are separated from each other by at least 4 patches. We denote these faces by f^1 .
 - All zero-measure faces in the geometrical model are subjected to 2 h-refinement levels. This refinement is performed to ensure that all extraordinary points are at least 3 patches away from the interface. As a result, the reduced continuity lines in the vicinity of an extraordinary point are pushed away from the interface. We denote the refined faces in this step by f^2 .
 - All faces that share an edge with f^1 and f^2 are subjected to one h-refinement level. The aim of applying this refinement is to avoid significant differences in the sizes of neighboring patches in the physical domain.
 - The refinement steps so far lead to the existence of T-junctions in the domain. To ensure the analysis suitability of the mesh in the regions of T-junctions, additional edges and vertices are added to the mesh. These edges and vertices are added to avoid intersections between T-junctions extensions, as elaborated in [52].

To ensure the interfaces of the geometrical model are accurately replicated in the mechanical model, the newly added control points along interface edges in the mesh are positioned using the B-spline curve refinement algorithm outlined in [26]. All other newly added control points are equally spaced between the vertices of the faces. The control points of the mesh after this step are denoted by $\mathbf{P}^M$ and are linearly correlated with $\mathbf{P}^{G\ell}$. This correlation can be expressed by:

$$\mathbf{P}^M = T\mathbf{P}^{G\ell}, \quad (4)$$

where $\mathbf{T}$ is the correlation matrix between $\mathbf{P}^M$ and $\mathbf{P}^{G\ell}$.

Figures 3 and 4 illustrate the refinement procedure in the vicinity of extraordinary points and zero-measure faces, respectively. Furthermore, these figures depict the formation of T-junctions after the refinement procedure. The refined mesh after applying the refinement procedure to the mesh in Figure 2f is demonstrated in Figure 5a. A zoom-in to the interface region is demonstrated in Figure 5b. The face extensions of the T-junction according to [52] are represented by green lines in both figures.

Step 2. To further enhance the quality of the mesh, a Laplacian smoothing procedure is performed on all internal control points in the control polygon. The interface control points remain fixed in place to preserve the shape and topology of the interface from the previous step. This step is again linear and can be expressed as:

$$\mathbf{P}^{M\ell} = \mathcal{L}^M \mathbf{P}^M, \quad (5)$$

where $\mathbf{P}^{M\ell}$ represents the control points after Laplacian smoothing and $\mathcal{L}^M$ is the correlation matrix.

Figure 5c shows the control polygon after the Laplacian smoothing procedure. A zoom-in to the interface region is presented in Figure 5d. This refined mesh is considered to be the final control polygon that defines the mechanical model. The physical domain of the mechanical model is shown in Figure 5e, and a zoom-in to the interface region is shown in Figure 5f.

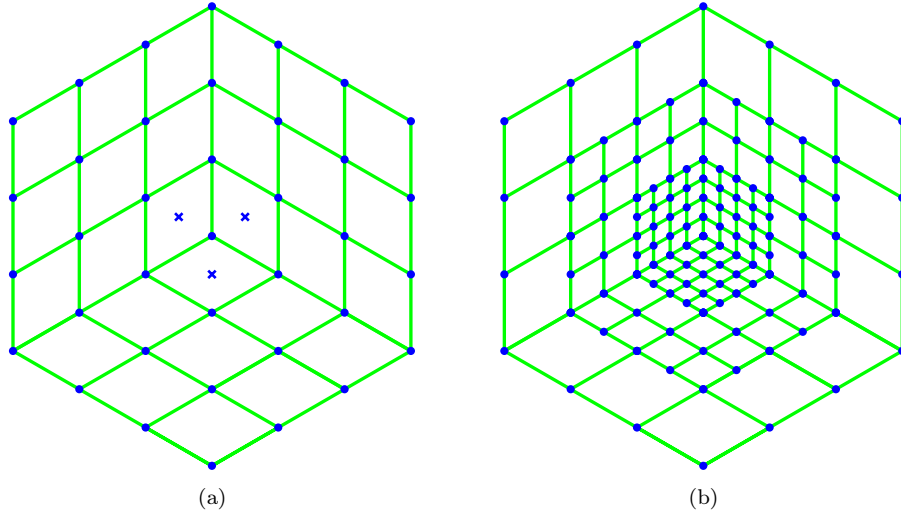


Figure 3: Refinement in the vicinity of extraordinary points. (a) The control mesh before the refinement. f^1 faces are marked with cross markers. (b) The refined mesh. f^1 faces are subjected to two h-refinement levels while neighboring faces are subjected to a single h-refinement level.

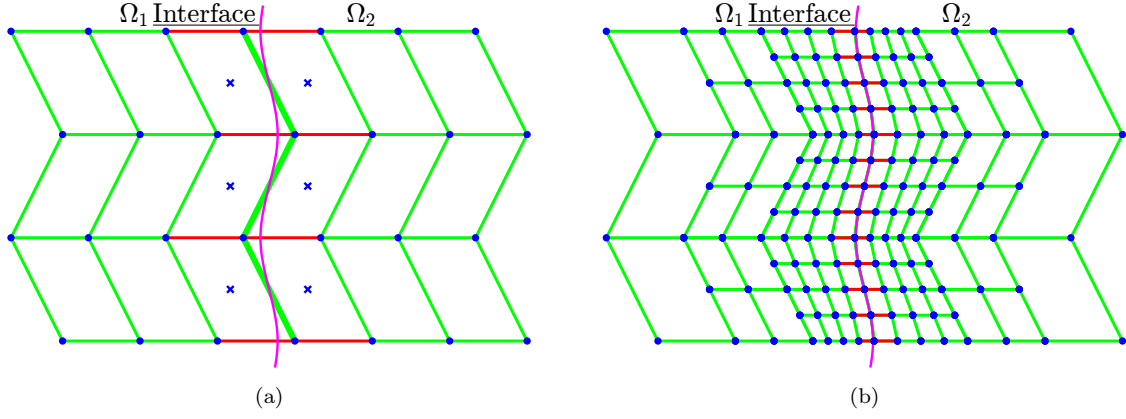
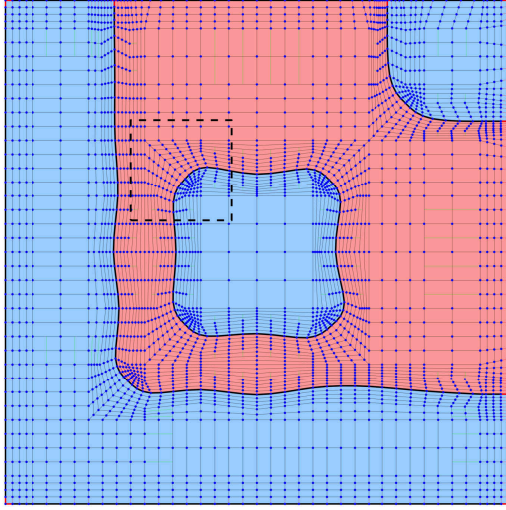
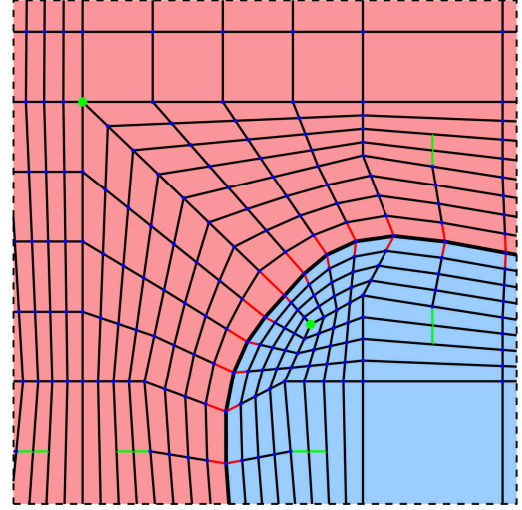


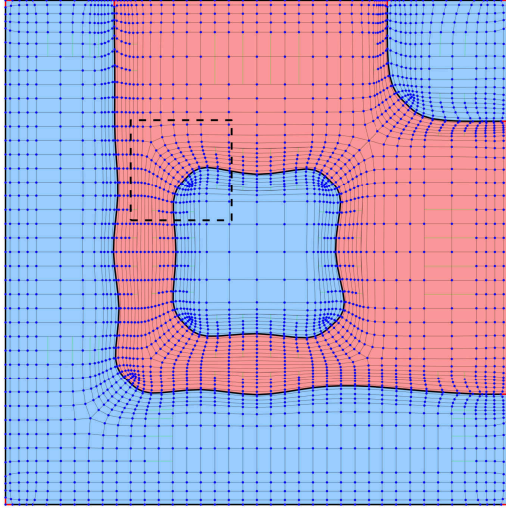
Figure 4: Refinement in the vicinity of zero-measure faces. (a) The control mesh before the refinement. f^2 faces are marked with cross markers. (b) The refined mesh. f^2 faces are subjected to two h-refinement levels while neighboring faces are subjected to a single h-refinement level. Notice that interface control points in the refined mesh are located to precisely replicate the interface geometry.



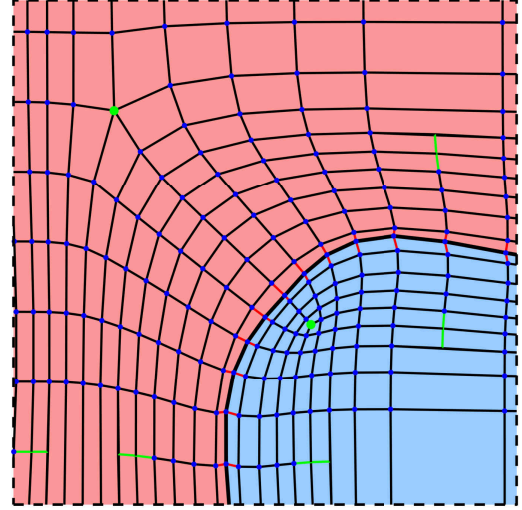
(a)



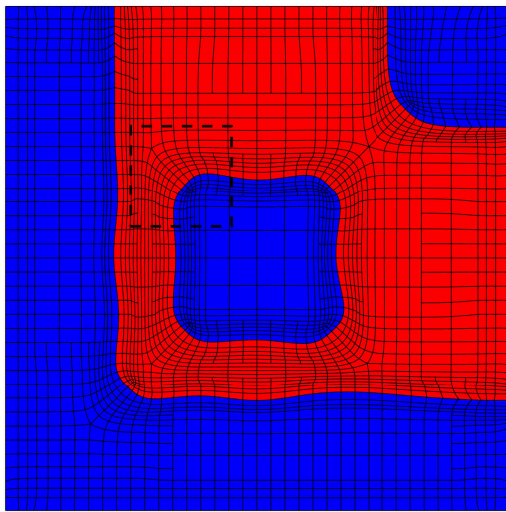
(b)



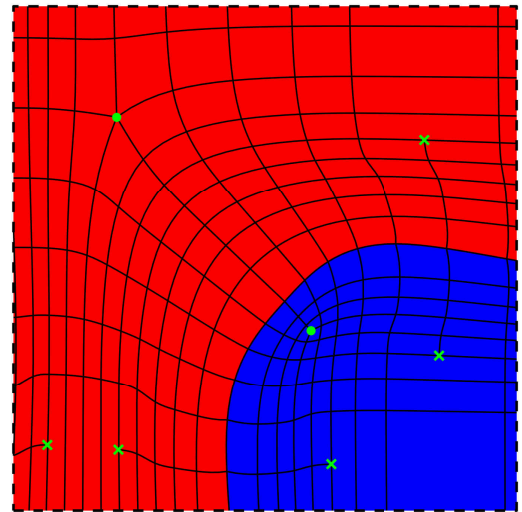
(c)



(d)



(e)



(f)

Figure 5: The procedure for generating the mechanical model. (a) The refined mesh according to Step 1 in Section 2.2. Unit and zero knot intervals are marked with black and red lines, respectively. The green lines represent the face extensions of the T-junctions in the mesh. (b) A zoom-in to the interface region in (a). (c) The refined mesh after applying the Laplacian smoothing procedure according to Step 2 in Section 2.2. (d) A zoom-in to the interface region in (c). (e) The physical domain of the mechanical model. (f) A zoom-in to the interface region in (e). The extraordinary points in this subfigure are highlighted with green dots, while t-junctions are highlighted by green cross markers.

3. Continuity control between the phases

One of the advantages of the proposed approach is its ability to control the continuity of the solution field between the different phases, in the direction perpendicular to the interface, without affecting the shape or topology of the interfaces in the geometrical and mechanical models. As elaborated in Section 2, the continuity between the phases that is obtained by the procedure presented so far is C^0 . However, for certain multiphase engineering problems, different continuity conditions may be required between the phases. For example, in multiphase flow problems, both C^{-1} and C^0 continuity conditions are required to represent the pressure and the velocity fields, respectively. For the degree 3 polynomials that are used in the present paper, four different continuity conditions can be imposed between the phases: C^{-1} , C^0 , C^1 , and C^2 . Accordingly, we denote the meshes that correspond to the different continuity conditions by $\mathcal{M}^{C^{-1}}$, $\mathcal{M}^{C^0}$, $\mathcal{M}^{C^1}$, and $\mathcal{M}^{C^2}$.

To achieve the required continuity condition between different phases, the knot intervals and control points near the interface are modified. This process is demonstrated using the example shown in Figure 6a, which depicts the interface region in an $\mathcal{M}^{C^0}$ mesh that could be obtained according to Section 2. The corresponding physical domain is demonstrated in Figure 6b. The control points lying on the interface between the phases in $\mathcal{M}^{C^0}$ are denoted by $\mathbf{P}^0$, while the control points in the adjacent layers on Ω_1 and Ω_2 are denoted by $\mathbf{P}_{\Omega_1}^1$ and $\mathbf{P}_{\Omega_2}^1$, respectively. The control points in the following layers are denoted in a similar pattern by $\mathbf{P}_{\Omega_1}^2$, $\mathbf{P}_{\Omega_2}^2$, $\mathbf{P}_{\Omega_1}^3$, and $\mathbf{P}_{\Omega_2}^3$, as shown in Figure 6a.

Starting from $\mathcal{M}^{C^0}$, the meshes $\mathcal{M}^{C^{-1}}$, $\mathcal{M}^{C^1}$, and $\mathcal{M}^{C^2}$ are obtained as follows:

- $\mathcal{M}^{C^{-1}}$: This algorithm replicates all the control points $\mathbf{P}^0$ in $\mathcal{M}^{C^0}$, creating two separate sets of control points $\mathbf{P}_{\Omega_1}^0$ and $\mathbf{P}_{\Omega_2}^0$ (Lines 1-2). Then, it associates each set with its respective phase, leading to two disconnected domains, Ω_1 and Ω_2 (Lines 3-4).

Algorithm 1: The algorithm for modifying $\mathcal{M}^{C^0}$ to $\mathcal{M}^{C^{-1}}$.

Data: $\mathcal{M}^{C^0}$

Result: $\mathcal{M}^{C^{-1}}$

- 1 $\mathbf{P}_{\Omega_1}^0 = \mathbf{P}^0$;
 - 2 $\mathbf{P}_{\Omega_2}^0 = \mathbf{P}^0$;
 - 3 Assign $\mathbf{P}_{\Omega_1}^0$ to Ω_1 ;
 - 4 Assign $\mathbf{P}_{\Omega_2}^0$ to Ω_2 ;
-

The resulting $\mathcal{M}^{C^{-1}}$ mesh is presented in Figure 6c. The control points of Ω_1 are kept blue, and the control points of Ω_2 are marked in red and slightly decreased in size to demonstrate the multiple control points along the interface. The physical domain of $\mathcal{M}^{C^{-1}}$ is demonstrated in Figure 6d.

- $\mathcal{M}^{C^1}$: This algorithm modifies $\mathcal{M}^{C^0}$ to have a single layer zero knot intervals separating the different phases. In order to preserve the interface's geometry, the control points in the $\mathbf{P}_{\Omega_1}^1$ and $\mathbf{P}_{\Omega_2}^1$ layers are repositioned so that their average location matches the control points in the $\mathbf{P}^0$ layer (lines 5-6, 8-9). This repositioning is based on the distances between

the control points in $\mathbf{P}^0$, $\mathbf{P}_{\Omega_1}^2$, and $\mathbf{P}_{\Omega_2}^2$ to prevent overlapping faces in the modified mesh (lines 2-4). Finally, the algorithm removes the control points in the $\mathbf{P}^0$ layer (Line 10) and connects the control points in the $\mathbf{P}_{\Omega_1}^1$ and $\mathbf{P}_{\Omega_2}^1$ layers with zero knot intervals (lines 10-11).

Algorithm 2: The algorithm for modifying $\mathcal{M}^{C^0}$ to $\mathcal{M}^{C^1}$.

Data: $\mathcal{M}^{C^0}$
Result: $\mathcal{M}^{C^1}$

```

1 for each point in  $\mathbf{P}^0$  do
2    $d_1 = \text{dist}(\mathbf{P}^0, \mathbf{P}_{\Omega_1}^2);$ 
3    $d_2 = \text{dist}(\mathbf{P}^0, \mathbf{P}_{\Omega_2}^2);$ 
4   if  $d_1 < d_2$  then
5      $\mathbf{P}_{\Omega_1}^1 = \text{mean}(\mathbf{P}^0, \mathbf{P}_{\Omega_1}^2);$ 
6      $\mathbf{P}_{\Omega_2}^1 = 2\mathbf{P}^0 - \mathbf{P}_{\Omega_1}^1;$ 
7   else
8      $\mathbf{P}_{\Omega_2}^1 = \text{mean}(\mathbf{P}^0, \mathbf{P}_{\Omega_2}^2);$ 
9      $\mathbf{P}_{\Omega_1}^1 = 2\mathbf{P}^0 - \mathbf{P}_{\Omega_2}^1;$ 
10  Remove  $\mathbf{P}^0$ ;
11  Connect  $\mathbf{P}_{\Omega_1}^1$  and  $\mathbf{P}_{\Omega_2}^1$  with zero knot intervals.
```

The $\mathcal{M}^{C^1}$ mesh after applying this procedure and its corresponding physical domain are demonstrated in Figures 6e and 6f, respectively. In this case, the zero-measure faces between the phases in Figure 6e are used to compute the basis functions that are associated with both phases. Note that the single zero knot intervals along the interface leads to a multiplicity of $m = 2$ in the local knots of the basis functions in the interface region. Following the relation between the continuity, the polynomial degree, p , and the knots multiplicity, m , the obtained continuity along the interface is $C^{p-m} = C^{3-2} = C^1$.

- $\mathcal{M}^{C^2}$: This algorithm transforms $\mathcal{M}^{C^0}$ to $\mathcal{M}^{C^2}$ by eliminating zero knot intervals in the interface region. In order to preserve the interface's geometry, the control points in layer $\mathbf{P}_{\Omega_2}^1$ are first positioned at the same locations as the control points in layer $\mathbf{P}^0$ (line 2). Then, the control points in the layers $\mathbf{P}_{\Omega_1}^2$ and $\mathbf{P}_{\Omega_2}^2$ are repositioned so that their average location matches the control points in layer $\mathbf{P}^0$ (lines 6-7, 9-10). This repositioning is based on the distances between the control points in $\mathbf{P}^0$, $\mathbf{P}_{\Omega_1}^3$, and $\mathbf{P}_{\Omega_2}^3$ to prevent overlapping faces in the modified mesh (lines 3-4). Finally, the two layers of control points $\mathbf{P}^0$ and $\mathbf{P}_{\Omega_2}^1$ are removed, and the control points in layers $\mathbf{P}_{\Omega_1}^2$ and $\mathbf{P}_{\Omega_2}^1$ are connected with a non-zero knot intervals (lines 11-12).

Algorithm 3: The algorithm for modifying $\mathcal{M}^{C^0}$ to $\mathcal{M}^{C^2}$.

Data: $\mathcal{M}^{C^0}$
Result: $\mathcal{M}^{C^2}$

```

1 for each point in  $\mathbf{P}^0$  do
2    $\mathbf{P}_{\Omega_2}^1 = \mathbf{P}^0$ ;
3    $d_1 = \text{dist}(\mathbf{P}^0, \mathbf{P}_{\Omega_1}^3)$ ;
4    $d_2 = \text{dist}(\mathbf{P}^0, \mathbf{P}_{\Omega_2}^3)$ ;
5   if  $d_1 < d_2$  then
6      $\mathbf{P}_{\Omega_1}^2 = \text{mean}(\mathbf{P}^0, \mathbf{P}_{\Omega_1}^3)$ ;
7      $\mathbf{P}_{\Omega_2}^2 = 2\mathbf{P}^0 - \mathbf{P}_{\Omega_1}^2$ ;
8   else
9      $\mathbf{P}_{\Omega_2}^2 = \text{mean}(\mathbf{P}^0, \mathbf{P}_{\Omega_2}^3)$ ;
10     $\mathbf{P}_{\Omega_1}^2 = 2\mathbf{P}^0 - \mathbf{P}_{\Omega_2}^2$ ;
11    Remove  $\mathbf{P}^0$  and  $\mathbf{P}_{\Omega_1}^1$ ;
12    Connect  $\mathbf{P}_{\Omega_1}^2$  and  $\mathbf{P}_{\Omega_2}^1$  with non-zero knot interval.
```

The mesh after applying this procedure and its corresponding physical domain are demonstrated in Figures 6g and 6h, respectively. Note that in order to obtain an admissible mesh, all the knot intervals between $\mathbf{P}_{\Omega_1}^2$ and $\mathbf{P}_{\Omega_2}^1$ should have the same non-zero value.

The primary objective of these algorithms is to control the continuity between phases while preserving the interface geometry. It is worth noting, however, that Algorithms 2 and 3 for generating the $\mathcal{M}^{C^1}$ and $\mathcal{M}^{C^2}$ meshes may alter the shape of internal edges in the mesh in the proximity of the interface. Nevertheless, this paper focuses on 2D planar geometries, and the shape preservation of the internal edges is not required.

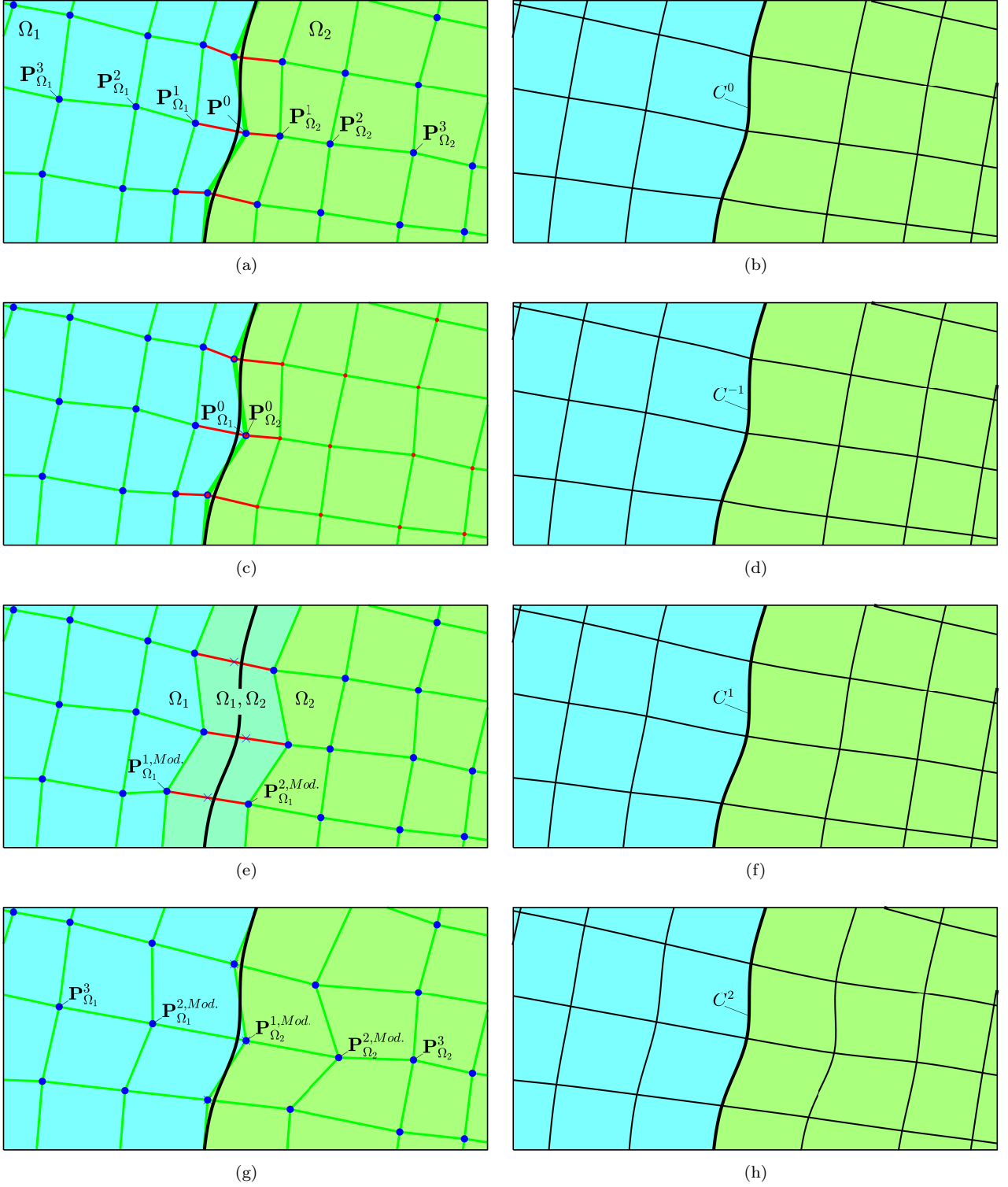


Figure 6: (a) A demonstration of $\mathcal{M}^{C^0}$ mesh that could be obtained after applying the algorithm in Section 2. (b) The physical domain of the mesh in (a). (c) The $\mathcal{M}^{C^{-1}}$ mesh. The replicated control points could be seen along the interface as coinciding blue and red dots. (d) The physical domain of (c). (e) The $\mathcal{M}^{C^1}$ mesh. The locations of the removed $\mathbf{P}^0$ layer are marked with blue cross-marks. (f) The physical domain of (e). (g) The $\mathcal{M}^{C^2}$ mesh. (h) The physical domain of (g). A superscript $\square^{Mod}$ is added to the modified control points.

4. Sensitivity analysis

In this section, we present the sensitivity of the mechanical model with respect to the control parameters of the multiphase domain. Since the level-set method is adopted in this paper to construct the multiphase domain, these control parameters are, in fact, the control parameters of the level-set function that are denoted by α in Section 2. For a clear presentation of the sensitivities, we partition the procedure for generating the mechanical model into five consecutively related differentiable parameters:

1. α – The control parameters of the level-set function,
2. $\mathbf{P}^G(\alpha)$ – The control points of the mesh after applying the untrimming procedure in Step 4 in Section 2.1,
3. $\mathbf{P}^{G\ell}(\mathbf{P}^G)$ – The control points of the mesh after applying the Laplacian smoothing in Step 5 in Section 2.1,
4. $\mathbf{P}^M(\mathbf{P}^{G\ell})$ – The control points of the refined model after applying Step 1 in Section 2.2,
5. $\mathbf{P}^{M\ell}(\mathbf{P}^M)$ – The control points of the final mechanical model after applying Step 2 in Section 2.2.

By this partition, the control points that represent the final mechanical model can be represented as:

$$\mathbf{P}^{M\ell}(\mathbf{P}^M(\mathbf{P}^{G\ell}(\mathbf{P}^G(\alpha)))), \quad (6)$$

Correspondingly, the sensitivity of the mechanical model with respect to the control design variables is expressed by:

$$\frac{\partial \mathbf{P}^{M\ell}}{\partial \alpha} = \frac{\partial \mathbf{P}^{M\ell}}{\partial \mathbf{P}^M} \frac{\partial \mathbf{P}^M}{\partial \mathbf{P}^{G\ell}} \frac{\partial \mathbf{P}^{G\ell}}{\partial \mathbf{P}^G} \frac{\partial \mathbf{P}^G}{\partial \alpha}. \quad (7)$$

The first term of Eq. (7) represents the sensitivity due to the Laplacian smoothing procedure in Step 2 in Section 2.2. The second term represents the sensitivity due to the refinement procedure in Step 1 in Section 2.2. The third term represents the sensitivity of the Laplacian smoothing procedure in Step 5 in Section 2.1. The fourth term represents the sensitivity of the untrimmed mesh after Step 4 in Section 2.1 with respect to the control parameters of the level-set function. These four terms are stated mathematically in Equations (5), (4), (3) and (1) and their sensitivities are given by:

$$\frac{\partial \mathbf{P}^{M\ell}}{\partial \mathbf{R}^M} = \mathcal{L}^M, \quad (8)$$

$$\frac{\partial \mathbf{P}^M}{\partial \mathbf{P}^{G\ell}} = \mathbf{T}, \quad (9)$$

$$\frac{\partial \mathbf{P}^{G\ell}}{\partial \mathbf{P}^G} = \mathcal{L}^G, \text{ and} \quad (10)$$

$$\frac{\partial \mathbf{P}^G}{\partial \alpha} = \mathcal{A}, \quad (11)$$

respectively.

Developing these sensitivities is essential for solving engineering problems with moving interfaces using the proposed approach. Engineering problems that require these sensitivities include topology optimization, multiphase flow, and phase transition (such as solidification and evaporation of liquids). Specifically for topology optimization, the sensitivity of the geometrical model, $\frac{\partial \mathbf{P}^{G\ell}}{\partial \boldsymbol{\alpha}} = \frac{\partial \mathbf{P}^{G\ell}}{\partial \mathbf{P}^G} \frac{\partial \mathbf{P}^G}{\partial \boldsymbol{\alpha}}$, might be required when dealing with physical or geometrical measures along the interface, such as limiting the maximal curvature of the interface in the optimized design.

5. Examples

This section is dedicated to demonstrating the ability of the proposed approach to solve discontinuous two-phase engineering problems. The examples include two-phase problems from different engineering fields. We begin with solving Poisson's equation, where the convergence and the effect of controlling the continuity between the phases are investigated. In the second example, a linear elasticity problem is solved both for a problem with an analytical solution and for a problem with a free-formed domain. In the third and fourth examples, the ability of the proposed approach to solve engineering problems with a moving interface is examined, where topology optimization and two-phase flow problems are solved. In the last two examples, the analytical sensitivities discussed in Section 4 are utilized to update the evolving interface.

In all examples, the domain of the problem and its boundary are denoted by Ω and $\partial\Omega$, respectively; The two different phases occupy the two open subdomains Ω_1 and Ω_2 such that $\overline{\Omega}_1 \cup \overline{\Omega}_2 = \overline{\Omega}$ and $\Omega_1 \cap \Omega_2 = \emptyset$; The Ω_1 phase corresponds to regions with positive values of the level-set function, indicated by the color red in all illustrations, while the Ω_2 phase corresponds to regions with negative values of the level-set function and colored blue. These phases are separated by the interface $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$, as depicted in Figure 7.

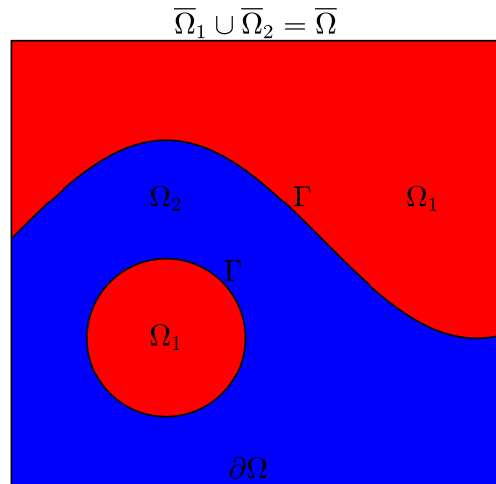


Figure 7: An illustration of a two-phase domain.

5.1. Example 1: Poisson's equation with different continuity conditions between phases

In the first example, we examine the convergence of the proposed approach for different continuity conditions between the phases. For this purpose, we solve the following Poisson's equation:

$$-\Delta\varphi = \frac{8k_i\pi^2}{l^2} \sin\left(\frac{2\pi x}{l}\right) \sin\left(\frac{2\pi y}{l}\right) \quad \text{in } \Omega, \quad (12)$$

where zero Dirichlet boundary conditions are imposed on the external boundary of the domain, $\partial\Omega$; k_i represents the material characteristics within Ω_i ; and l is a geometrical parameter depending on the dimensions of the domain Ω . In this particular example, Ω has a square shape, and l is selected to be equal to the square's side length.

The examination is conducted on a model of a square plate with a quarter-circle inclusion made of a different material in one of its corners. The side length of the plate is 2 units, while the radius of the inclusion is 1 unit. The control mesh of the mechanical model is presented in Figure 8a, with extraordinary points marked by green dots and the extensions of T-junctions indicated by green lines. The corresponding physical domain is presented in 8b. The development so far assumes a unified weight for all the basis functions, so the hole is not precisely circular. Instead, it will be a degree 3 B-spline whose control points are chosen to replicate as closely as possible the circular shape. This mechanical model consists of 520 Bézier patches and 699 control points.

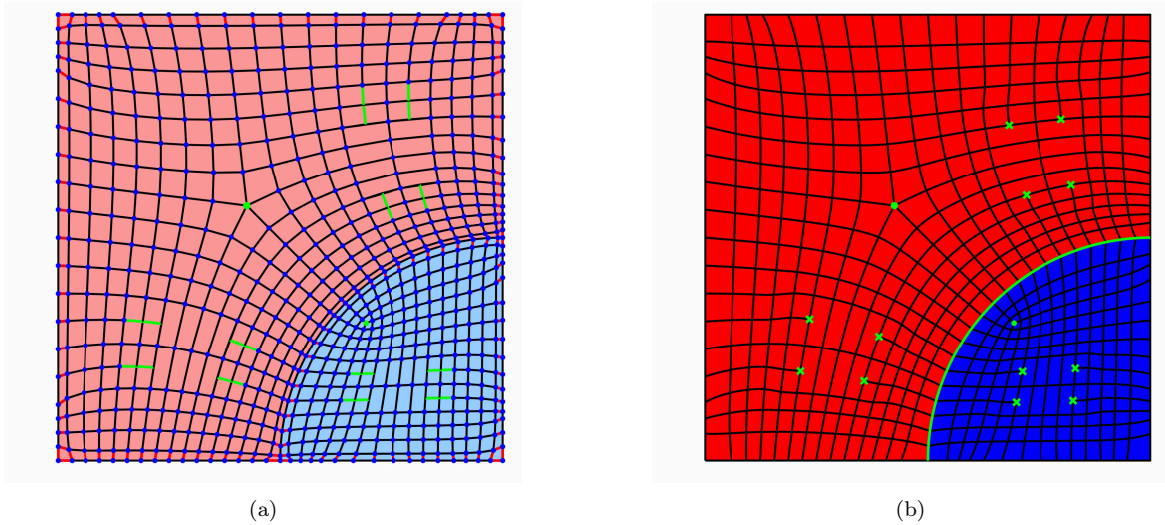


Figure 8: The mechanical model for a square plate with a quarter-circle inclusion. (a) The control mesh. (b) The physical domain.

First, to investigate convergence under different continuity conditions in the perpendicular direction to the interface, and to compare with an analytical solution, we assume that the two materials have the same characteristics, $k_1 = k_2 = k$. The corresponding analytical solution in this case is:

$$\varphi_a = k \sin\left(\frac{2\pi x}{l}\right) \sin\left(\frac{2\pi y}{l}\right). \quad (13)$$

The numerical solution, φ_n , for $k = 2$ with C^0 continuity between the phases is shown in Figure

9a. To enhance distinguishing between the different phases, the solution is presented once more using the original red-blue color scheme for the phases in Figure 9b.

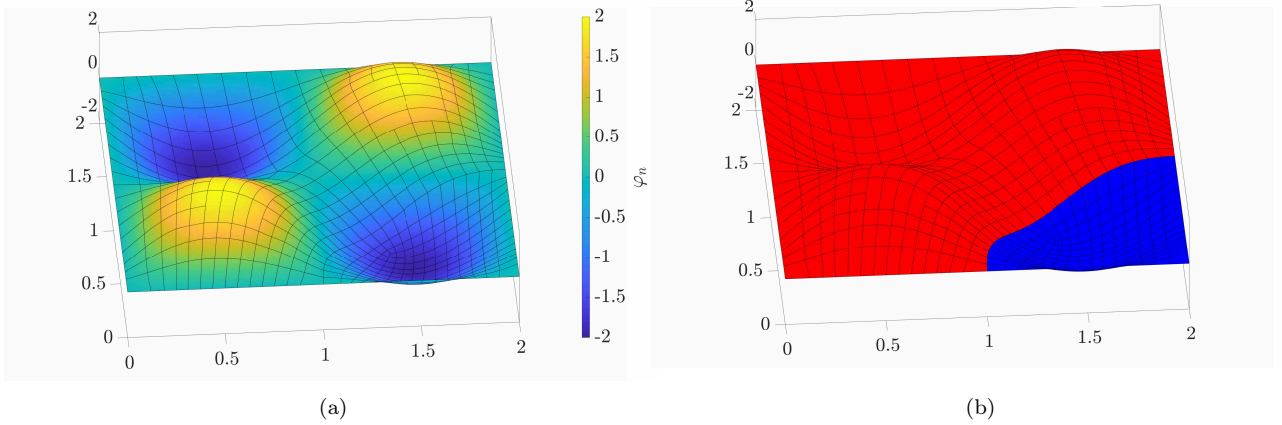


Figure 9: Numerical solution of Eq.(12) on the mechanical model shown in Figure 8 for $k = 2$ with C^0 continuity between phases. (a) Color-scaled representation. (b) Representation with the original red-blue color scheme for the phases.

To generate refined meshes, the T-junctions are initially eliminated from the model through local refinement in their vicinity. Subsequently, refined meshes are generated through a series of single h-refinement steps. The numerical solutions over these meshes for $k = 2$ with C^0 continuity between the phases are displayed in Figure 10. The numbers of degrees of freedom for all the refinement levels under different continuity conditions between phases are summarized in Table 2. the first row in the Table corresponds to the meshes presented in Figure 10. Note that the number of degrees of freedom in this example is equal to the number of control points in the mechanical model. The corresponding convergence graphs for the L^2 norm and the H^1 semi-norm with respect to the number of degrees of freedom are displayed in Figures 11a and 11b, respectively.

	RL^0	RL^1	RL^2	RL^3
$\mathcal{M}^{C^0}$	741	2857	11217	44449
$\mathcal{M}^{C^1}$	716	2808	11120	44256
$\mathcal{M}^{C^2}$	691	2759	11023	44063

Table 2: The numbers of degrees of freedom for all the refinement levels under different continuity conditions between phases. The first row in this table corresponds to the meshes presented in Figure 10. The abbreviation RL^i corresponds to - Refinement Level i .

The convergence graphs indicate the convergence of the proposed approach for different continuity conditions between the phases. Even though the continuity between the phases in the analytical solution is ∞ , increasing the continuity between the phases in the numerical solution led to a larger error. This is due to the reduction of the number of degrees of freedom that is accompanied with increasing the continuity. Note that as the continuity increases between the different phases, the number of degrees of freedom decreases. This is the result of applying the suggested algorithms in Section 3. It should be noted that developing algorithms that increase the number of degrees of freedom as continuity between phases increases could potentially reduce

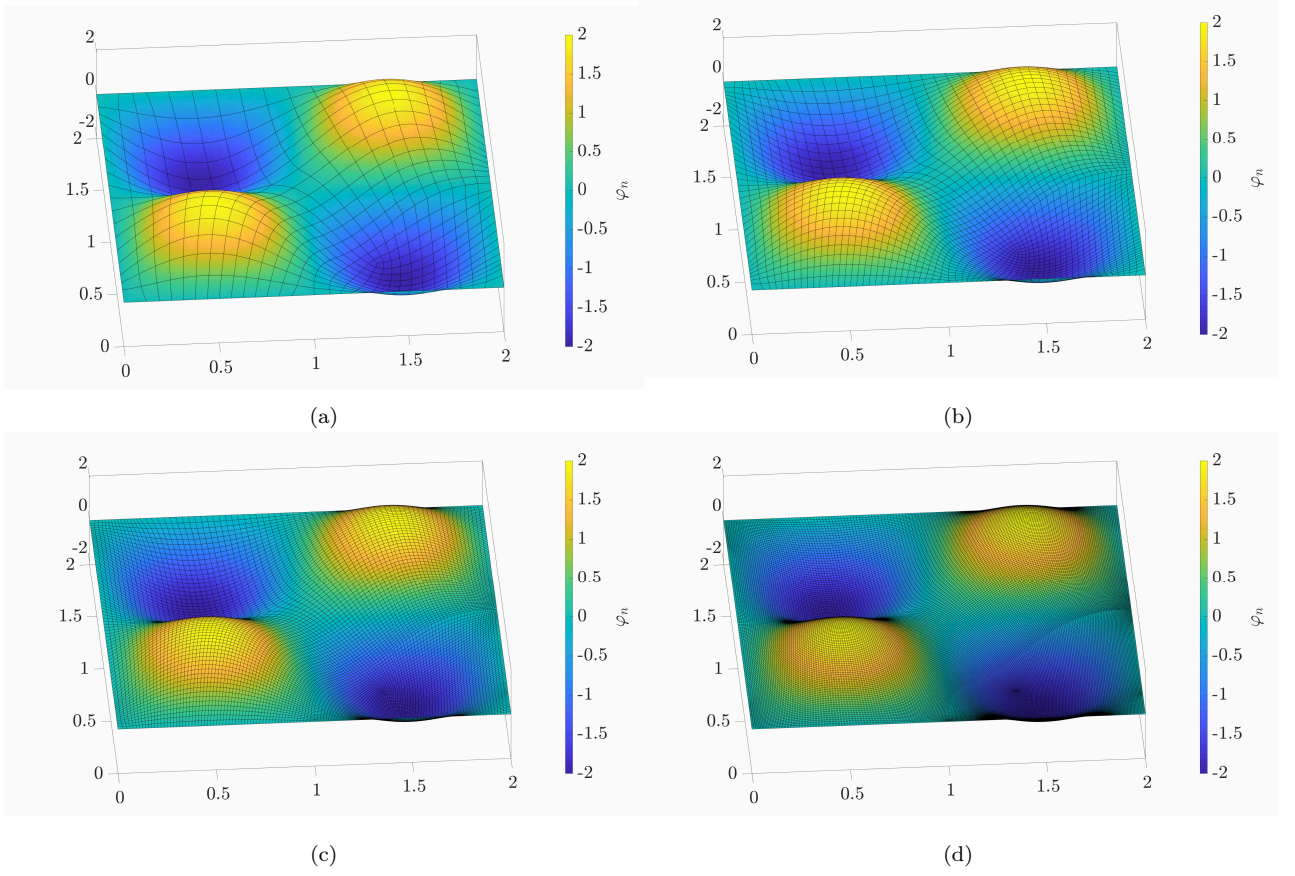


Figure 10: The numerical solutions over meshes with different refinement levels for $k = 2$ with C^0 continuity between the phases. Sub-figures (a) to (d) correspond to refinement levels 0 to 3, respectively. The numbers of degrees of freedom associated with these meshes are presented in the first row of Table 2.

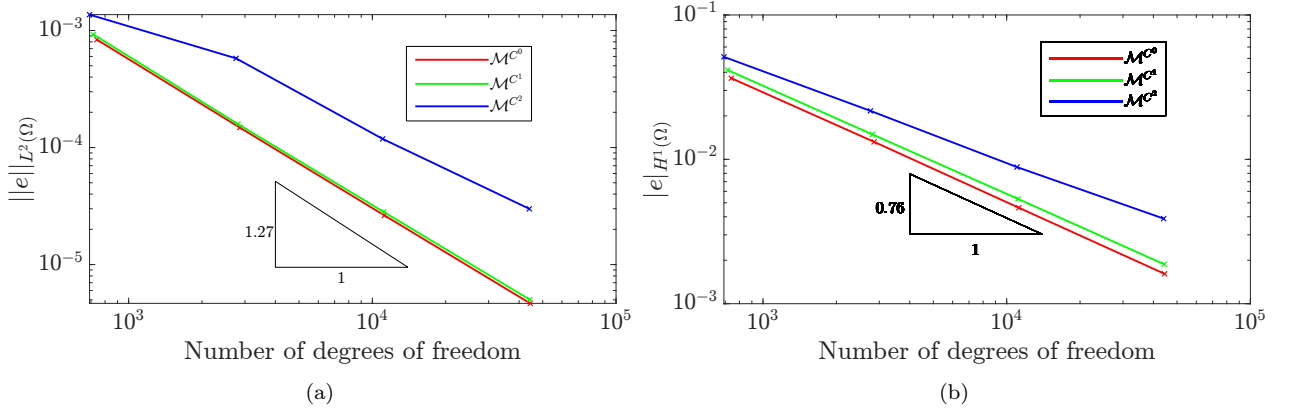


Figure 11: Convergence graphs for the Poisson's equation with $k_1 = k_2 = 2$ and different continuity conditions between the phases. (a) L^2 norm. (b) H^1 semi-norm.

the error, but this is beyond the scope of the current research as the error values for different continuity conditions are of the same order of magnitude and possess very close convergence rates.

Furthermore, the convergence rates illustrated in both convergence graphs align closely with the theoretical expectations detailed in [53]. In mechanical models incorporating extraordinary points, the convergence rates depend not only upon polynomial degrees but also upon the number and valence of the extraordinary points within the model. As discussed in Section 2.2, the continuity along the spoke edges of an extraordinary point is reduced from C^2 to C^0 . With

mesh refinement, the physical region supported by extraordinary points shrinks, causing the physical length of these C^0 continuity lines to decrease. And this has a direct effect on the convergence rates. For more details on convergence rates in mechanical models that incorporate extraordinary points, the reader is referred to [53]. Convergence for the case where the phases have different material characteristics is conducted in the subsequent example.

Finally, we display the solution of Eq. (12) over the mechanical model depicted in Figure 5e. The material parameters are $k_1 = 2$ and $k_2 = 4$ and continuity of C^0 between the phases. This model is a square with a side length of 8 units and consists of 2,516 Bézier patches and 3,007 control points. The numerical solution is presented in Figure 12.

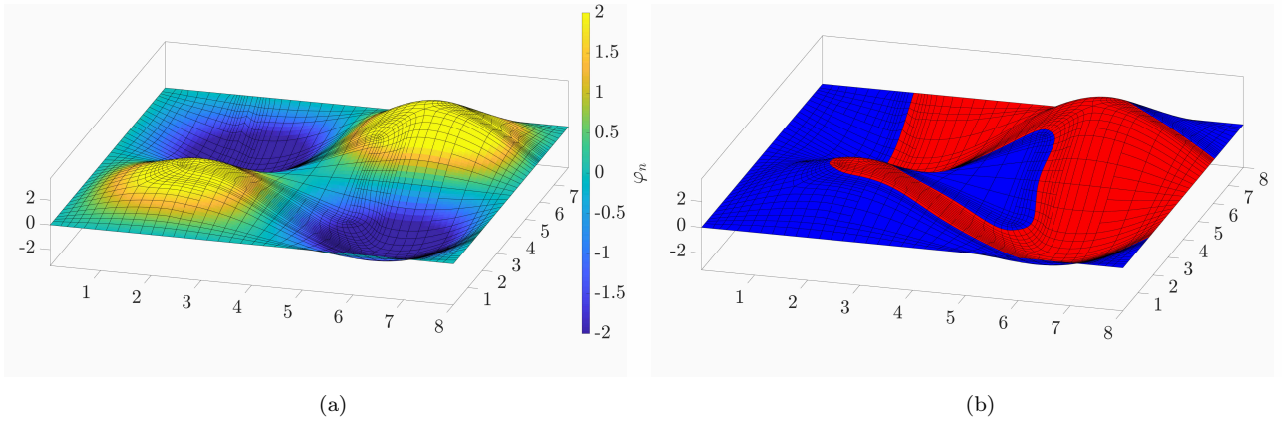


Figure 12: Numerical solution of Eq.(12) on the mechanical model shown in Figure 5e for $k = 2$ with C^0 continuity between phases. (a) Color-scaled representation. (b) Representation with the original red-blue color scheme for the phases.

5.2. Example 2: Static linear elasticity

This example demonstrates the effectiveness of the proposed approach in solving static linear elasticity problems within two-phase domains. Additionally, it showcases the approach's capability to ensure smooth and well-defined stresses along the explicitly defined interface. The example starts with a theoretical overview, followed by a showcase example that includes a comparison with an analytical solution. Finally, an illustrative example of a free-formed geometry is presented.

5.2.1. Theoretical overview

This example deals with static linear elasticity problems while neglecting body forces. The corresponding partial differential equation is:

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}, \quad \text{in } \Omega, \quad (14)$$

where $\boldsymbol{\sigma}$ represents the stress tensor, which characterizes the internal forces within the material. For each material phase, the stress-strain relationship is given by Hooke's Law: $\boldsymbol{\sigma} = \mathbf{C}_i : \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is the strain tensor and $\mathbf{C}_i$ is elasticity tensor, describing the material's mechanical properties in Ω_i . The strain tensor $\boldsymbol{\varepsilon}$ is related to the displacement field $\mathbf{u}$ as follows: $\boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$. Here, $\mathbf{u}$ corresponds to the displacement vector field, representing the material's deformation, and serves as the solution field for Eq. (14).

5.2.2. A disk plate with a circular inclusion

To assess the effectiveness of the proposed approach in solving static linear elasticity problems within two-phase domains, we first consider a problem with an analytical solution: a circular inclusion of a different material centered within a disk plate. The disk plate is subjected to a radial displacement of $u_r(r) = r$, where r represents the radius. The analytical solution for the plain strain case is provided in [54].

The inner and outer radii are set to $r_1 = 1$ and $r_2 = 3$, respectively. By exploiting the tangential symmetry of the problem, we model only a quarter of the disk plate, while the numerical model encompasses a bi-symmetric square region, as depicted in Figure 13. This numerical model covers a square with a side length of $a = 2$. Symmetric boundary conditions are applied along the two symmetry axes. Precise tractions, based on the analytical solution in [54], are applied to the remaining edges of the model. The Young's modulus and Poisson's ratio for the material in Ω_1 are $E_1 = 2$ and $\nu_1 = 0.2$, respectively. For the material in Ω_2 , these values are $E_2 = 1$ and $\nu_2 = 0.25$. The mechanical model in this example is identical to the one in the previous example, with its control mesh and physical domain presented in Figures 8a and 8b, respectively.

To examine the proposed approach, we initially present the numerical solutions for the radial and tangential stresses, σ_{rr} and $\sigma_{\theta\theta}$. These solutions are illustrated in Figures 14a and 14b, respectively. Recall that the solution to this physical problem herein is the displacement field, and its continuity in the perpendicular direction to the interface is C^0 . This leads to C^{-1} continuity in the numerical solutions of both σ_{rr} and $\sigma_{\theta\theta}$, in the direction perpendicular to the

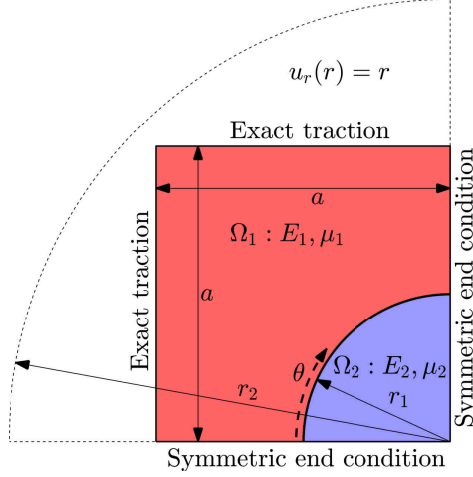


Figure 13: Problem settings and numerical model for the disk plate with inclusion example.

interface. However, the analytical solutions exhibit C^0 and C^{-1} continuity in the same direction for σ_{rr} and $\sigma_{\theta\theta}$, respectively. Therefore, despite the C^{-1} continuity of the numerical σ_{rr} in the direction perpendicular to the interface, we expect to obtain equal values of σ_{rr} from both sides of the interface. We consider this as an accuracy indicator of this method. In Figure 15, we present numerical solutions of σ_{rr} along the interface within the domain $30^\circ < \theta < 60^\circ$, obtained through varying mesh refinement levels. The analytical σ_{rr} is constant along the interface and it equals 2.09302. This solution is marked with a black line in the Figure. Dashed and continuous colored lines correspond to solutions within Ω_1 and Ω_2 respectively. From this graph, we can see that as the mesh is refined the difference between the values of σ_{rr} from opposing sides of the interface converge. Moreover, with increasing mesh refinement, these values approach proximity to the analytical solution. Moreover, due to the C^2 displacement field along the tangential direction of the interface, both numerical stresses exhibit C^1 continuity in that same direction.

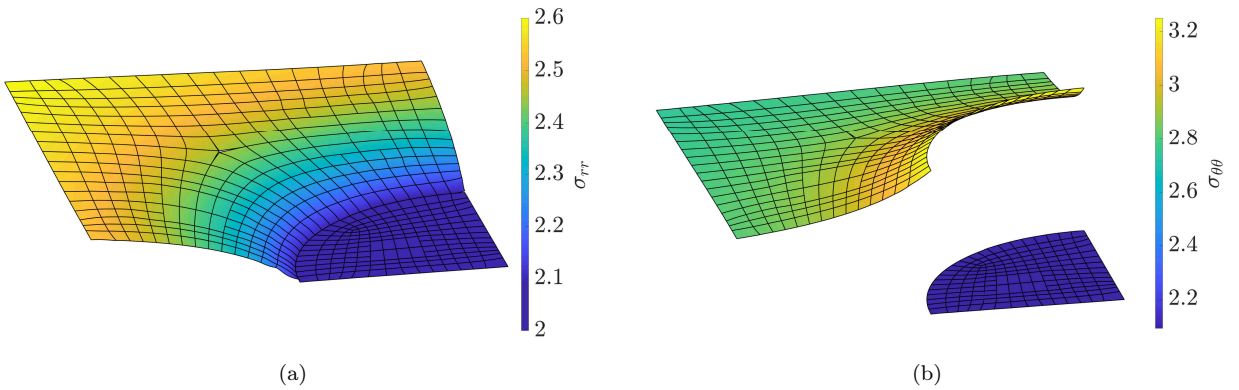


Figure 14: Numerical stresses over the disk plate with a centered circular inclusion example. (a) Radial stress, σ_{rr} ; (b) Tangential Stress, $\sigma_{\theta\theta}$.

Additionally, Figure 14 demonstrates the reduced continuity lines near an extraordinary point. As stated in Section 2, the construction of the basis functions in this paper results in a solution field of C^0 continuity along the spoke edges of an extraordinary point. Consequently, this leads to C^{-1} continuity in the stress field along these edges. However, it is essential to

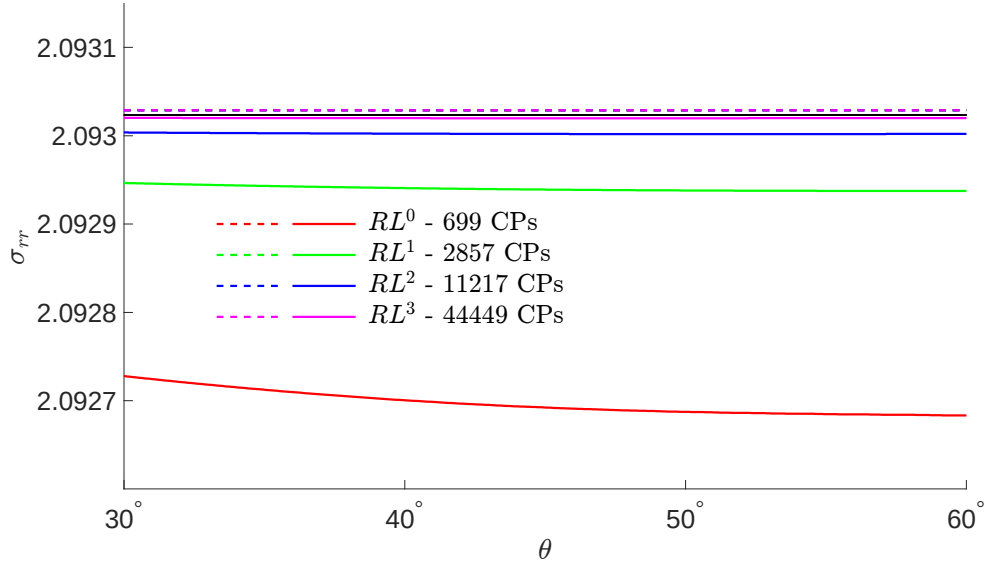


Figure 15: σ_{rr} along the circular interface within the domain $30^\circ < \theta < 60^\circ$. The analytical solution is marked by a solid black line, while dashed colored lines represent the numerical solution within Ω_1 , and continuous colored lines represent the numerical solution within Ω_2 . Here, RL and CPs correspond to refinement levels and control points, respectively.

emphasize that the extraordinary points are situated at least 3 patches away from the interface due to the refinement procedure outlined in Section 2. As a result, the reduced continuity lines surrounding an extraordinary point do not impact the smoothness along the interfaces. To enhance the continuity of the solution field near extraordinary points, the algorithm proposed in [49] can be used where G^1 continuity for the basis functions in the vicinity of extraordinary points can be obtained.

The convergence graph for the L^2 norm of the solution field in this example with respect to the number of degrees of freedom is displayed in Figure 16. This graph indicates the convergence of the proposed approach, while the convergence rate aligns with their theoretical expectations as presented in [53] and discussed in the previous example.

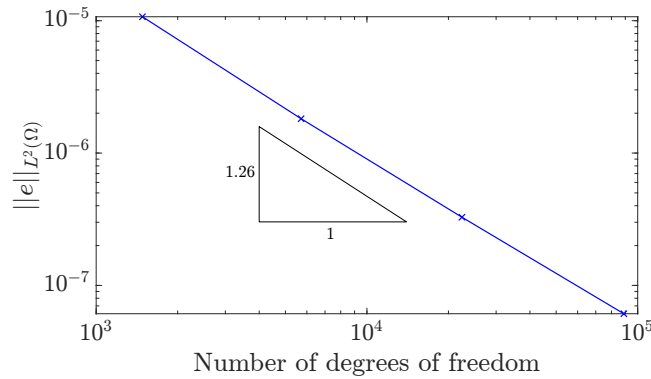


Figure 16: Convergence graph of the L^2 norm for the linear elasticity example.

5.2.3. General free-formed geometry

In this subsection, the proposed approach is tested for solving linear elasticity over a free-formed geometry. The examined structure is a simply supported beam with a uniformly distributed load where only the symmetric half part is modeled. The problem settings are depicted

in Figure 17. The elasticity moduli for the material in Ω_1 and Ω_2 are $E_1 = 10$ and $E_2 = 5$, respectively, while the Poisson's ratio is set to 0.25.

The geometrical model in this example is constructed based on the level-set function shown in Figure 18. The level-set function is defined over a mesh of 15×60 bi-cubic Bézier patches and 18×63 control points. The corresponding geometrical and mechanical models according to Section 2 are presented in Figures 19a and 19b, respectively. A zoomed-in views to the interface regions in the green boxes in Figures 19a and 19b are presented in Figures 19c and 19d, respectively. The geometrical model consists of 1,416 control points and 991 Bézier patches while the mechanical model consists of 11,243 control points and 10,527 Bézier patches.

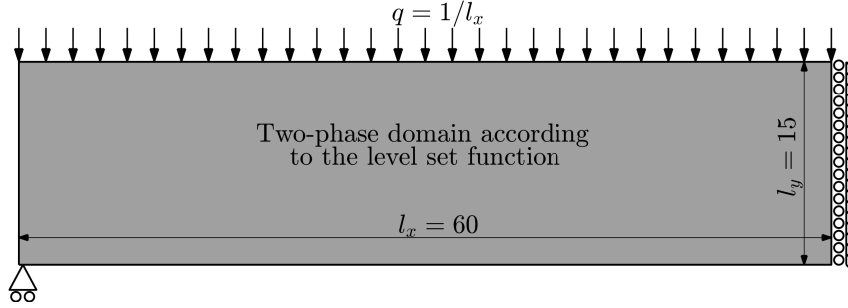


Figure 17: Linear elasticity over a free-formed geometry - Problem settings.

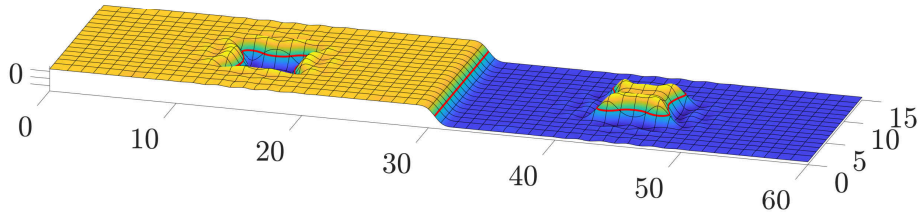


Figure 18: The level-set function for generating the geometrical model of the simply supported beam.

The stress field σ_x is demonstrated in Figure 20a. One of the benefits of the proposed approach is demonstrated in Figure 20b, which displays a zoomed-in perspective view of the green box in the stress field shown in Figure 20a. This figure indicates to the smoothness of the stresses along the interface in both phases. Furthermore, as previously noted, the solution field maintains C^2 continuity in the tangential direction and C^0 continuity in the perpendicular direction to the interface. Consequently, the continuity of the stress in those directions is C^{-1} and C^1 , respectively. In addition, Figure 20b shows the reduced continuity lines near an extraordinary point.

To further emphasize the benefits of the proposed approach for solving static linear elasticity problems, we evaluate σ_x along the exact interface between the phases. For this purpose, the internal interface on the right-hand side in Figure 19b is selected. The stresses in both phases are presented in the graph in Figure 21. According to Section 2, the interface curve is represented as a periodic cubic B-spline curve. The B-spline curve is extracted according to Appendix B, and

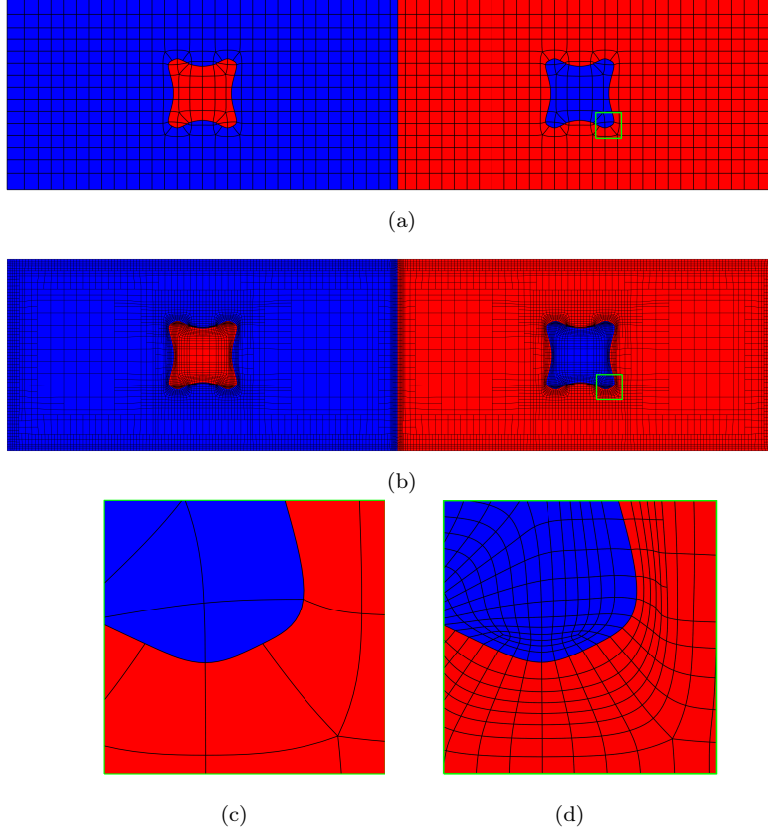


Figure 19: (a) The geometrical model of the simply supported beam corresponding to the level-set function in Figure 18. (b) The Mechanical model corresponding to (a). (c) A zoom-in to green square in (a). (d) A zoom-in to green square in (b).

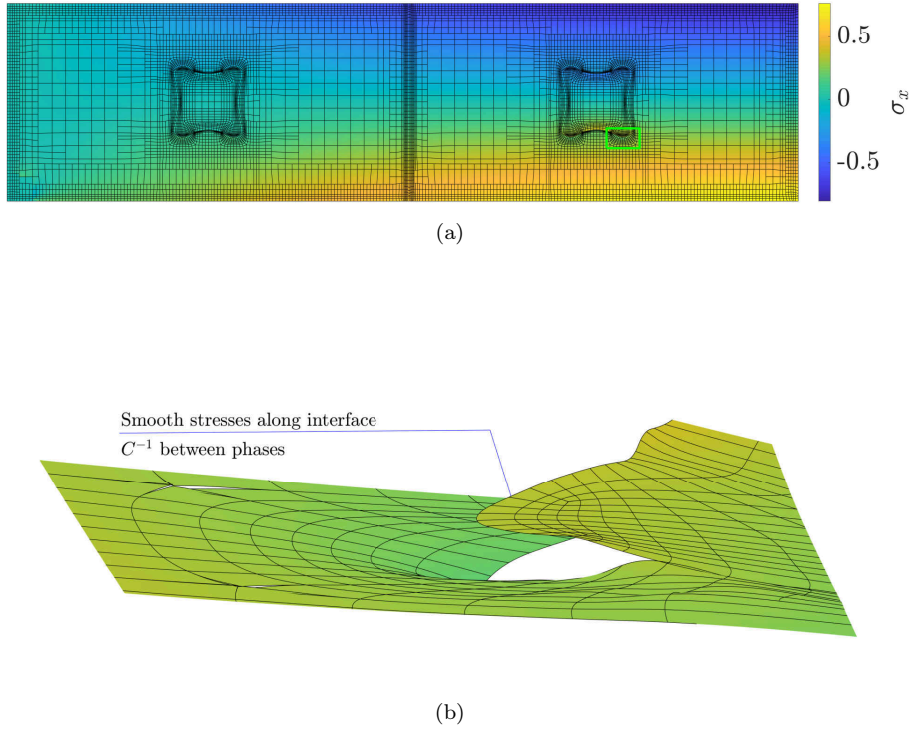


Figure 20: (a) The stress field σ_x over the domain of the beam.. (b) A zoom-in to the green box in Figure 20a.

it is marked with a magenta color in the figure. Five points, A-E, along the curve, are chosen to be located on the graphs. We note that the continuity of the displacement field along the

interface is C^2 . This leads to stresses of C^1 continuity along the interface.

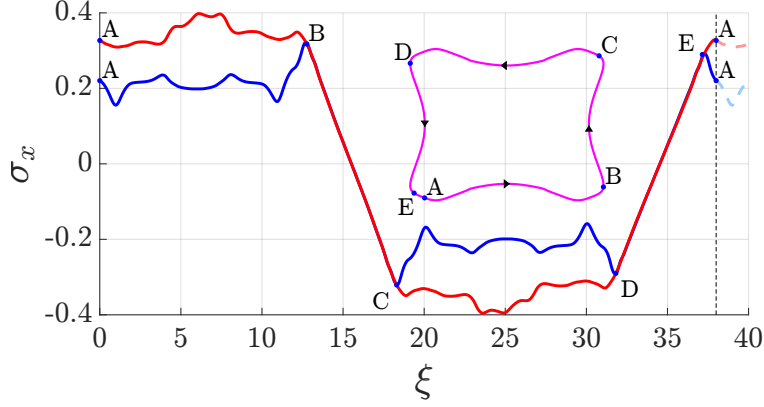


Figure 21: The stress σ_x evaluated along the exact interface in both phases. The stress field is C^1 continuous in the tangential direction and C^{-1} continuous in the perpendicular direction to the interface.

5.3. Example 3: Topology optimization for static linear elasticity

In this example, a topology optimization problem that involves static linear elasticity over a two-phase domain is solved. This example shows the ability of the proposed approach to deal with engineering problems that involve moving boundaries and topological changes. The problem settings are depicted in Figure 22. The two-phase domain consists of two materials with different properties. The optimization problem is solved to minimize the compliance of the structure while constraining the amount of the stiffer material, with the larger elasticity modulus. This optimization problem is commonly used to maximize the rigidity of the structure while controlling the amount of the constituent materials.

The formulation of the optimization problem is as follows:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize:}} \quad & J(\boldsymbol{\alpha}, \mathbf{u}), \\ \text{s.t.:} \quad & V_1(\boldsymbol{\alpha})/V_1^* - 1 \leq 0, \\ \text{with:} \quad & \mathbf{K}(\boldsymbol{\alpha})\mathbf{u}(\boldsymbol{\alpha}) = \mathbf{f}, \end{aligned} \tag{15}$$

where J is the compliance,

$$J = \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{u}) \mathbf{C} \boldsymbol{\epsilon}(\mathbf{u}) \, d\Omega. \tag{16}$$

$\boldsymbol{\alpha}$ is the vector of the optimization design variables, namely the z-coordinates of the control points parametrizing the level-set function; the expression $\boldsymbol{\epsilon}(\mathbf{u}) \mathbf{C} \boldsymbol{\epsilon}(\mathbf{u})$ represents the strain energy, whose minimization is equivalent to compliance minimization; $V_1(\boldsymbol{\alpha})$ is the volume of Ω_1 , and computed by

$$V_1(\boldsymbol{\alpha}) = \int_{\Omega_1} 1 \, d\Omega_1; \tag{17}$$

V_1^* is the maximum allowed volume of Ω_1 ; and $\mathbf{K}(\boldsymbol{\alpha})$ and $\mathbf{f}$ are the global stiffness matrix and the global external force vector, respectively. Without loss of generality, we assume for the current demonstration that the force vector $\mathbf{f}$ does not depend on the design variables $\boldsymbol{\alpha}$.

The sensitivities of the compliance and the volume expression with respect to the control design variables α are analytically computed by:

$$\frac{\partial J}{\partial \alpha} = \frac{\partial J}{\partial \mathbf{P}^\Gamma} \frac{\partial \mathbf{P}^\Gamma}{\partial \alpha}, \quad \text{and} \quad (18)$$

$$\frac{\partial V_1}{\partial \alpha} = \frac{\partial V_1}{\partial \mathbf{P}^\Gamma} \frac{\partial \mathbf{P}^\Gamma}{\partial \alpha}, \quad (19)$$

respectively, where $\mathbf{P}^\Gamma$ represent the control points along the interface. The derivatives $\frac{\partial J}{\partial \mathbf{P}^\Gamma}$ are computed as suggested by Allaire et al. [55], where the compliance derivative is calculated using the adjoint approach.

The design variables are updated using the Method of Moving Asymptotes (MMA) as suggested in [56]. These variables are bounded within the range of -1 to 1 . External move limits of 0.1 are enforced on MMA updates to the design variables. The optimization process terminates when the absolute difference in both the objective function and the constraint over three consecutive optimization cycles is less than 10^{-3} .

The structure solved in this example is a three-point bending beam where only one symmetric half is modeled and optimized. The initial B-spline surface consists of 25×60 Bézier patches and is constructed over a 28×63 control points mesh leading to 1764 optimization design variables. The maximum allowed volume, V_1^* , is set to 900, which is 60% of the volume of the design domain. The Young's modulus and Poisson's ratio for the material in Ω_1 are $E_1 = 10$ and $\nu_1 = 0.3$, respectively. For the material in Ω_2 , these values are $E_2 = 1$ and $\nu_2 = 0.25$.

The optimization started with the initial level-set configuration that is shown in Figure 23 where the zero-level-set is highlighted with red lines. The initial geometrical model is shown in Figure 24a. This model consists of 3474 control points and 2046 Bézier patches. The values of the objective function J and the volume V_1 corresponding to this topology are $J = 107001$ and $V_1 = 1025.4$, respectively. The optimization process concluded after 32 design cycles, resulting in a final objective function of $J = 72256$ and $V_1 = 900.3$. This design leads to a 33% reduction in the initial compliance value, while also meeting the volume constraint. Snapshots of the 7th, 13th, and the final (32nd) design cycles are shown in Figures. 24b-24d, respectively. The convergence graph of the optimization process is displayed in Figure 25 where the objective is scaled according to its value in the initial topology. This graph indicates the stability of the optimization procedure. In the first two iterations, the optimization process prioritizes the fulfillment of the volume constraint, resulting in an increase in the objective function value. Once the volume constraint is satisfied, the reduction of the objective function value begins.

The results shown in this example demonstrate the ability of the proposed approach to solve moving interface problems, particularly topology optimization, while maintaining a precise and explicit representation of the interfaces. This paves the way for further exploration of the proposed approach in addressing other optimization problems. In particular, those that involve engineering problems that are sensitive to the quality of the boundary representation, such as stress-constrained topology optimization or topology optimization for fluid-structure interac-

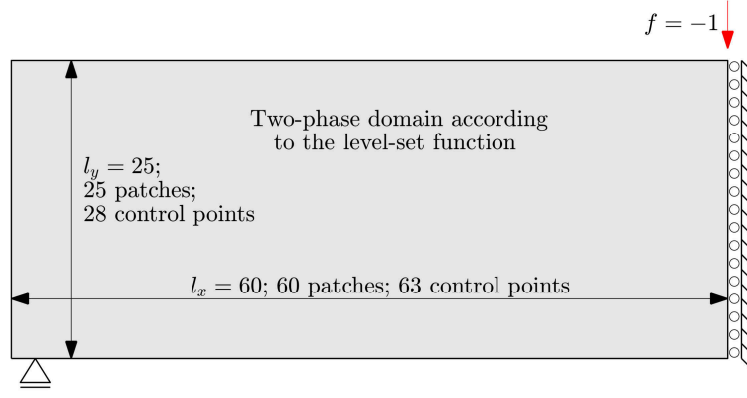


Figure 22: Problem settings for the topology optimization example.

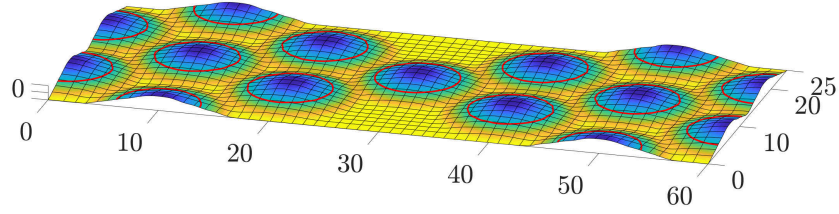
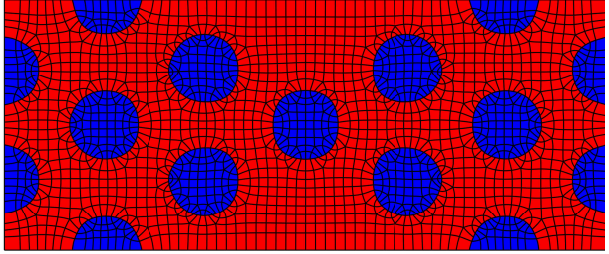
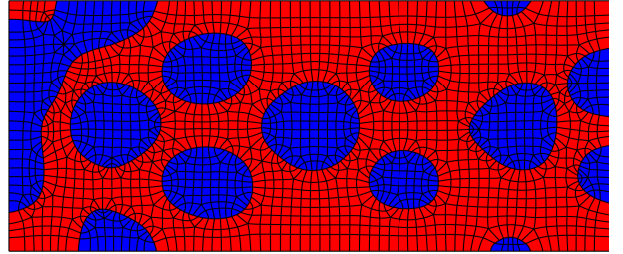


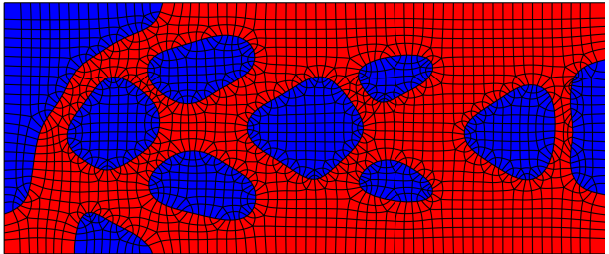
Figure 23: Initial level-set configuration for the topology optimization example.



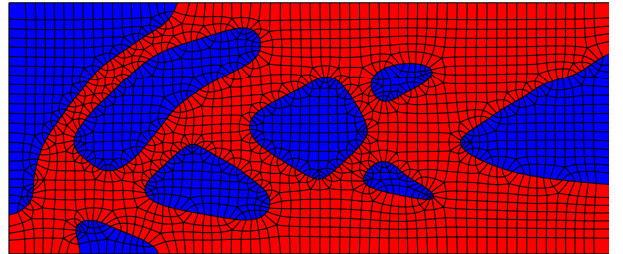
(a) Initial design, $J = 107001$, $V_1 = 1025.4$.



(b) Iteration 7, $J = 84467$, $V_1 = 893.6$.



(c) Iteration 13, $J = 73110$, $V_1 = 896.3$.



(d) Iteration 32, $J = 72256$, $V_1 = 900.3$.

Figure 24: Snapshots from the optimization procedure with the corresponding values of the objective, J , and constrained volume, V_1 .

tion problems. Additionally, and particularly beneficial for topology optimization applications, the optimal designs obtained using the proposed approach are compatible with state-of-the-art

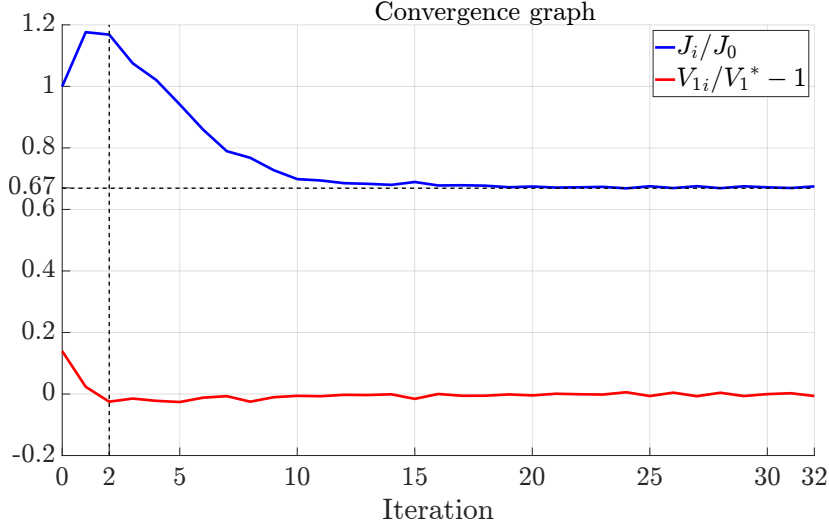


Figure 25: Convergence graph for the topology optimization example.

computer-aided design software due to their spline-based representation. This makes the proposed approach able to maintain a consistent representation across geometric, mechanical, and computer-aided design models, in all design cycles eliminating the transitional stages typically associated with adopting different representations for the different models, which can introduce inaccuracies into the optimization procedure.

5.4. Example 4: Stokes equation on two-phase domain

In this example, we demonstrate the ability of the proposed approach to address two-phase flow problems by solving the Stokes equation over a domain of two-phase fluids, neglecting the inertia forces. The problem settings of this example are shown in Figure 26. This example is characterized by constantly moving interfaces while the shape and topology of the different fluids evolve based on the solution of the governing equations. To provide a comprehensive understanding of the implementation of this example, we provide a brief overview of the theory and implementation remarks in the following subsection.

5.4.1. Theory and implementation remarks

The Stokes equations for two-phase flow are:

$$\nabla p - \mu_i \nabla^2 \mathbf{u} = \rho_i \mathbf{g} \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (20a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega_i, \quad i = 1, 2, \quad (20b)$$

where $\mathbf{u}$ and p are the variables of the governing equations and represent the velocity and pressure fields, respectively. μ_i and ρ_i are the dynamic viscosity and the density of the fluids in Ω_i , respectively; $\mathbf{g}$ is the gravitational acceleration field. The boundary conditions on the interface between the phases are:

$$[\mathbf{u}] = 0 \quad \text{on } \Gamma, \quad (21a)$$

$$[\boldsymbol{\sigma} \cdot \mathbf{n}_\Gamma] = -\gamma \kappa \mathbf{n}_\Gamma \quad \text{on } \Gamma, \quad (21b)$$

where $\boldsymbol{\sigma}$ is the Cauchy tensor and it is defined by $\boldsymbol{\sigma} = -p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$; $\mathbf{n}_\Gamma$ is the normal vector to the interface Γ ; γ is the surface tension coefficient; and κ is the curvature of Γ . The brackets $[\square]$ represent the difference between the physical quantities in the different phases along the interface, e.g. $[\mathbf{u}] = \mathbf{u}_1 - \mathbf{u}_2|_\Gamma$, $\mathbf{u}_i = \mathbf{u}|_{\Omega_i}$, $i = 1, 2$. The boundary conditions on $\partial\Omega$ are presented in Figure 26.

The term in (21a) ensures a continuous velocity field between the different phases. This condition is achieved by adopting $\mathcal{M}^{C^0}$ mesh for the velocity field according to Section 3. Conversely, the pressure field is constructed using $\mathcal{M}^{C^{-1}}$ mesh leading to the required C^{-1} continuity between phases. The term in (21b) represents the difference in the Cauchy tensor due to the surface tension force that acts along the interface.

To numerically solve the equations in (20), the numerical solutions for the velocities and the pressure fields and their corresponding test functions are constructed using the same bi-cubic spline-based representations following the Bézier extraction procedure discussed in Section 2. To stabilize the pressure field, the Pressure-Stabilizing Petrov-Galerkin approach (PSPG) has been adopted. Denoting the B-spline basis functions by N , the stabilized weak form of (20) can be represented as:

$$\int_{\Omega} \mu \nabla N \nabla \mathbf{u}^h - \nabla N p^h d\Omega = \int_{\Omega} N \rho \mathbf{g} d\Omega - \int_{\Gamma} N_{\Gamma} \gamma \kappa \mathbf{n}_{\Gamma} d\Gamma, \quad (22a)$$

$$\int_{\Omega} \nabla N p^h d\Omega + \sum_{patch=1}^{N_{patch}} \frac{\tau_{PSPG}}{\rho} \int_{\Omega_{patch}} \nabla N \cdot [\nabla p^h - \mu \nabla^2 \mathbf{u}^h - \rho \mathbf{g}] d\Omega_{patch} = 0, \quad (22b)$$

where $\mathbf{u}^h = (u_x^h, u_y^h)$ and p^h are the numerical solutions of the velocity and the pressure fields, respectively; Ω_{patch} is the domain of a single patch in Ω ; τ_{PSPG} is the stabilization parameter and is computed by $\tau_{PSPG} = 0.1 \frac{l^2}{4\nu}$, where l is the mean value of the maximum patch lengths in the two directions and ν is the kinematic viscosity, $\nu = \mu/\rho$;

One of the advantages of the proposed approach for solving two-phase problems is its ability to accurately evaluate physical measures on the exact geometry of the interface. This feature is effectively utilized in the present example to determine the curvature and normal vector of the interface, which are essential for evaluating the surface tension force term in equation (22a). Given the explicit B-spline representation of the interface, these measures can be precisely evaluated as follows:

$$\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{1.5}}, \quad (23)$$

$$\mathbf{n}_{\Gamma} = \frac{1}{(x'^2 + y'^2)^{0.5}} \begin{bmatrix} -y' \\ x' \end{bmatrix}, \quad (24)$$

where the $\square'$ and $\square''$ are the first and second derivatives that can be directly obtained by the derivatives of N_{Γ} .

The geometry of the fluids evolves by updating the level-set function based on the velocity field in the perpendicular direction to the interface, $\mathbf{u}_{\Gamma\perp}^h$. Since the interface is explicitly defined,

$\mathbf{u}_{\Gamma\perp}^h$ can be evaluated at precise locations along the interface as follows:

$$\mathbf{u}_{\Gamma\perp}^h = \mathbf{u}_{\Gamma}^h \cdot \mathbf{n}_{\Gamma} \quad \text{on } \Gamma. \quad (25)$$

The evolution of the interface is governed by modifying its corresponding control points. Denoting the interface control points by $\mathbf{P}^{\Gamma}$ ($\mathbf{P}^{\Gamma} \in \mathbf{P}^{M_{\ell}}$), their temporal derivatives are computed by:

$$\frac{\partial \mathbf{P}^{\Gamma}}{\partial t} = \int_{\Gamma} N_{\Gamma} \mathbf{u}_{\Gamma\perp}^h d\Gamma. \quad (26)$$

To update the level-set function, whose control parameters are denoted by $\boldsymbol{\alpha}$, the temporal derivatives $\frac{\partial \boldsymbol{\alpha}}{\partial t}$ are computed by solving the following system of equations:

$$\frac{\partial \mathbf{P}^{\Gamma}}{\partial \boldsymbol{\alpha}} \frac{\partial \boldsymbol{\alpha}}{\partial t} = \frac{\partial \mathbf{P}^{\Gamma}}{\partial t}, \quad (27)$$

where the term $\frac{\partial \mathbf{P}^{\Gamma}}{\partial \boldsymbol{\alpha}}$ is analytically computed following the sensitivity analysis presented in Section 4; and $\frac{\partial \boldsymbol{\alpha}}{\partial t}$ represents the unknown variables. This equation ensures finding the level-set function that leads to the desired geometrical changes of the interface according to equations (25) and (26). The system of equations in (27) consists of $2N_{P\Gamma}$ equations in 2D and N_{α} variables where $N_{P\Gamma}$ and N_{α} are the number of interface control points in the mechanical model and the number of control parameters $\boldsymbol{\alpha}$ that support $\mathbf{P}^{\Gamma}$. The number of equations in (27) does not necessarily equals to the number of variables, leading to either an underdetermined or overdetermined system of equations, depending on $N_{P\Gamma}$ and N_{α} . In the current example, this system of equations is solved using the least squares method.

Although the level-set method is commonly used for two-phase flow problems with moving interfaces, it does not naturally guarantee that the amount of fluid in each phase is preserved over time. To ensure that this amount remains equal to its value in the initial configuration, the level-set function is modified at every 5th time step following Algorithm 4. This algorithm receives as input the areas of phase 2 at times $t = 0$ and $t = t_i$, denoted by $A_2^{t_0}$ and $A_2^{t_i}$, respectively, along with the level-set controlling parameters at time $t = t_i$, represented by $\boldsymbol{\alpha}^{t_i}$ and the allowed tolerance, denoted by ϵ . It returns the modified level-set controlling parameters $\boldsymbol{\alpha}_{modified}^{t_i}$. The algorithm begins by computing the error between $A_2^{t_i}$ and $A_2^{t_0}$ (Line 1) and initializing $\boldsymbol{\alpha}_{modified}^{t_i}$ (Line 2). Then, as the error is larger than a specific threshold ϵ (Line 3) the negative values of $\boldsymbol{\alpha}_{modified}^{t_i}$ are then multiplied by $\left(\frac{A_2^{t_0}}{A_2^{t_i}}\right)^{0.5}$ (Line 4-5). This adjustment leads to increasing or decreasing these values in a manner that minimizes the current error ϵ_{t_i} (Line 6). The tolerance ϵ is chosen to be 0.01 based on numerical tests that showed a good balance between the accuracy of the fluid's area and the efficiency of modifying it.

Algorithm 4: This algorithm modifies the level-set function to preserve the initial amount of fluids in each phase over time.

Data: $A_2^{t_0}, A_2^{t_i}, \alpha^{t_i}, \epsilon$

Result: $\alpha_{modified}^{t_i}$

- 1 $\epsilon^{t_i} = \frac{A_2^{t_i} - A_2^{t_0}}{A_2^{t_0}};$
 - 2 $\alpha_{modified}^{t_i} = \alpha^{t_i};$
 - 3 **while** $absolute(\epsilon^{t_i}) > \epsilon$ **do**
 - 4 $idx = \text{indices where } \alpha < 0;$
 - 5 $\alpha_{modified}^{t_i}[idx] = \alpha_{modified}^{t_i}[idx] \left(\frac{A_2^{t_0}}{A_2^{t_i}} \right)^{0.5};$
 - 6 Evaluate $\epsilon^{t_i}(\alpha_{modified}^{t_i});$
-

5.4.2. Implementation

The procedure presented above is applied to the two-phase flow domain illustrated in Figure 26. The level-set function that is used to generate the initial multiphase domain is shown in Figure 27. This level-set function consists of 18×33 control points and 15×30 bi-cubic Bézier patches. The corresponding geometrical and mechanical models, along with zoomed-in views of the interface region, are presented in Figure 28. The geometrical model consists of 804 control points and 517 Bézier patches. The mechanical model consists of 8,053 control points and 7,403 Bézier patches.

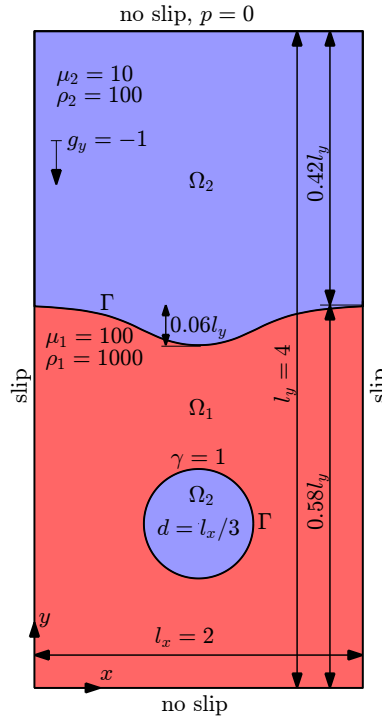


Figure 26: Problem settings for the two-phase flow example.

The evolution of the interface is computed over the time interval $t \in [0, 6]$, using a uniform time step of 0.025. The velocities and pressure solution fields at $t = 0$ are presented in Figure 29. This figure demonstrates the ability of the presented approach to assign distinct continuity between phases for the different types of solution fields. While using the algorithms in Section

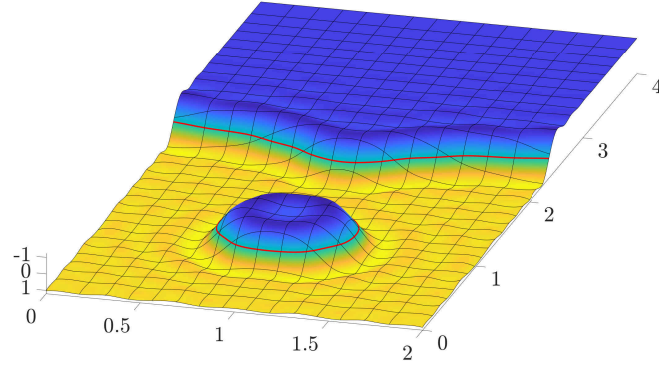


Figure 27: The level-set function that is used to generate the initial multiphase domain

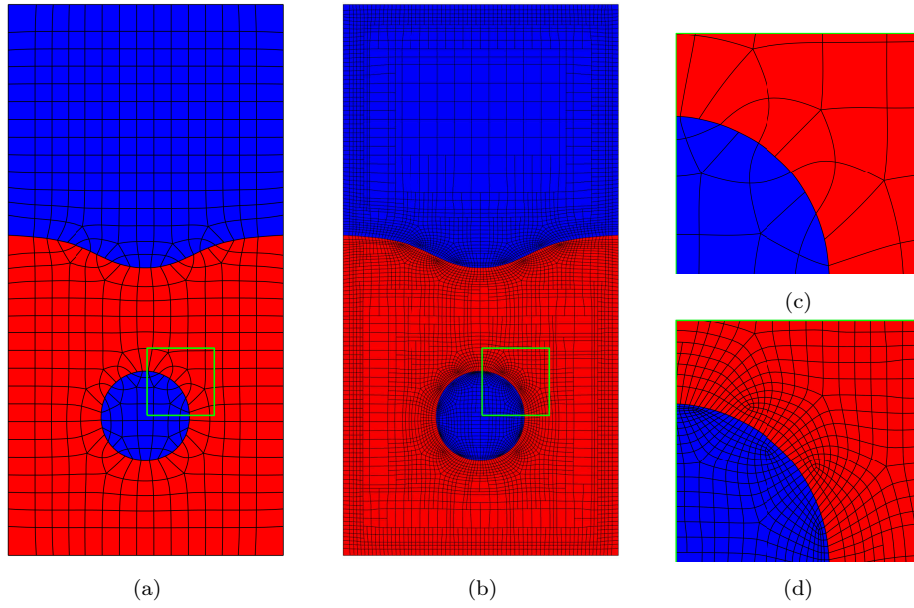
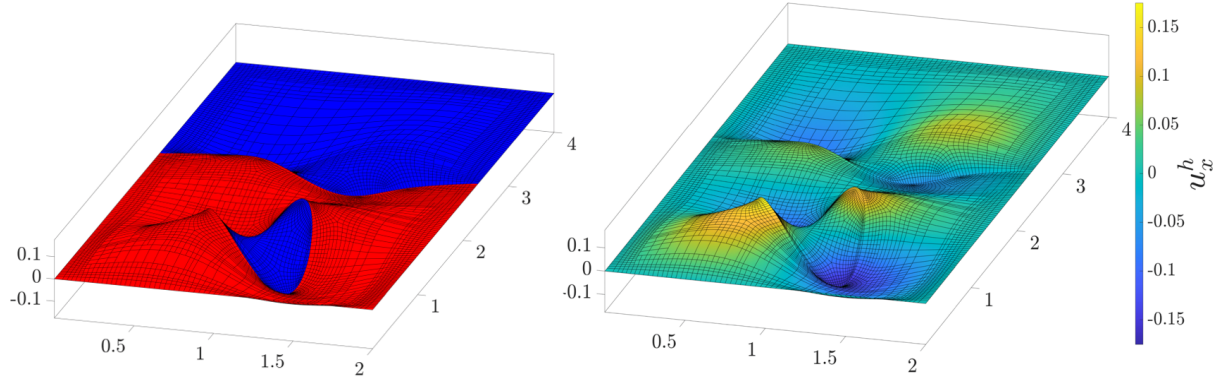


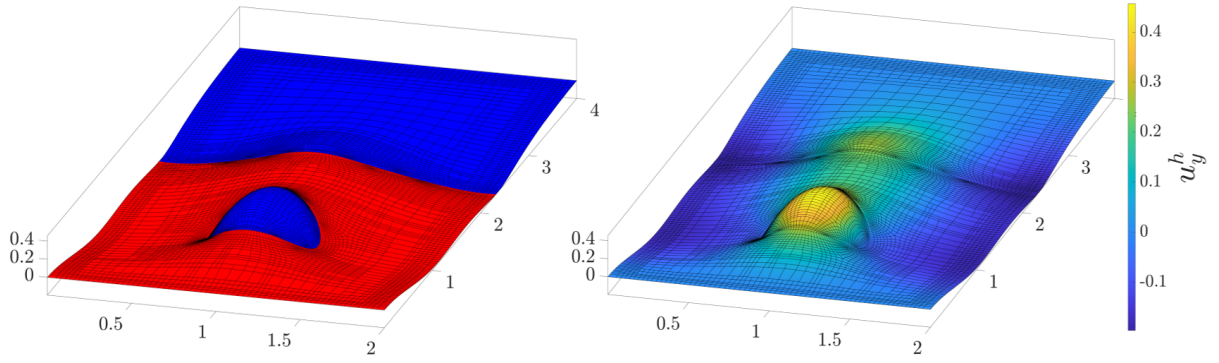
Figure 28: Two phase flow - initial configuration. (a) Geometrical model. (b) Mechanical model. (c) Zoom-in to the green box in (a). (d) Zoom-in to the green box in (b).

3 the continuity in the perpendicular direction to the interface is C^0 for the velocity fields and C^{-1} for the pressure field.

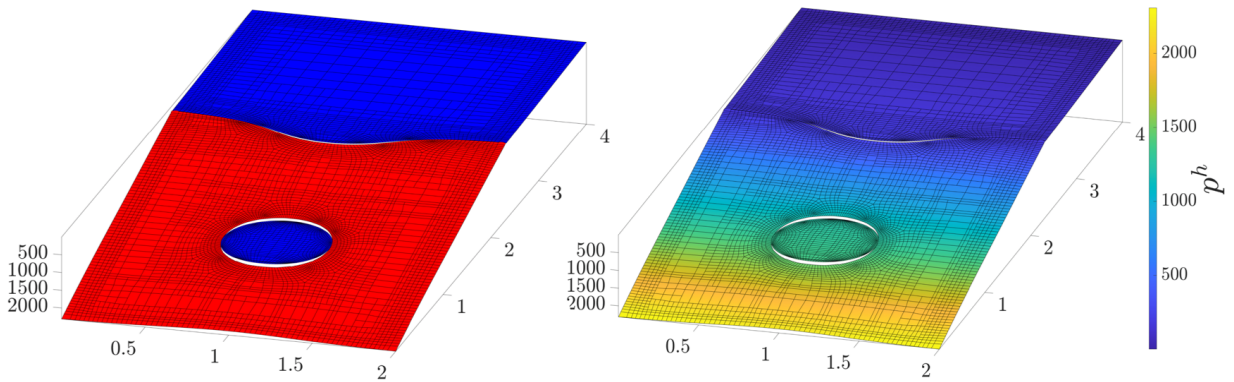
As stated earlier, the topological evolution of the fluids is governed based on the normal velocity that is evaluated along the precise interface. A demonstration of this velocity is shown in Figure 30. The geometry of the fluids along with contour levels of the corresponding total velocity field at various time points are shown in Figure 31. At time $t = 3.25$ the two domains associated with Ω_2 merge, demonstrating the ability of the presented approach to deal with topological changes. At time $t = 6.0$ the interface between the different phases flattens, leading to an equilibrium state with very small fluid velocities.



(a) $u_x^h(t = 0)$



(b) $u_y^h(t = 0)$



(c) $p^h(t = 0)$

Figure 29: The Velocities and pressure solution fields at $t = 0$. (a) The numerical velocity field in the x direction. (b) The numerical velocity field in the y direction. (c) The numerical pressure field with the expected C^{-1} continuity between the different phases is clearly visible.

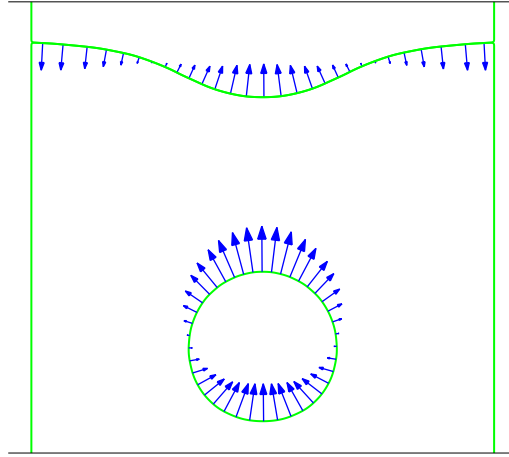


Figure 30: The velocity in the direction perpendicular to the interface/

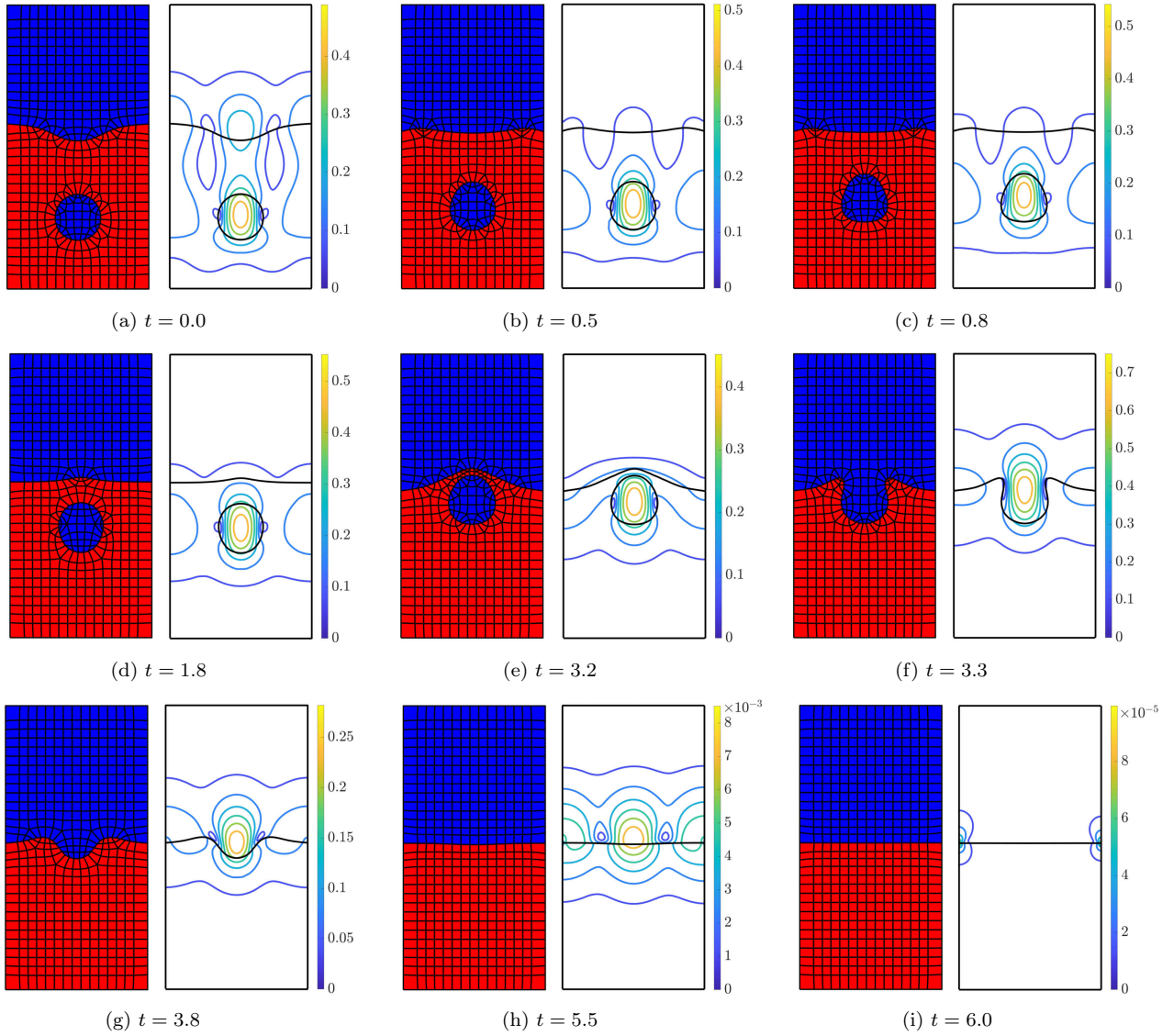


Figure 31: The geometry of the fluids and the contour levels of the corresponding total velocity field at various points in time. The color bars in these figures represent the total velocity of the fluids.

6. Discussion

In most level-set-based applications, the level-set-based topology is typically considered as the final geometrical model. However, due to the implicitness of the level-set-based topology, constructing analysis-suitable mechanical models and evaluating physical measures along the interface often require approximating the geometry. Additionally, since the level-set-based representation is not directly suitable for analysis, a mesh-based model is reconstructed for analysis purposes, leading to two different representations for the geometrical and mechanical models. A key advantage of the proposed approach is its ability to avoid the undesired implicitness of the level-set-based topology by mapping it, using a differentiable procedure, into an explicitly defined spline-based topology that represents the geometrical model. We clarify that the level-set-based topology in this work serves only as an intermediate step in constructing the geometrical model. The mapping procedure in this paper is developed to produce an explicit geometry that can be precisely replicated by the mechanical model. Future implementations might adopt different mapping procedures leading to different geometrical models. However, as long as the mapping procedure is differentiable and does not lead to overlapping patches, it will still be considered valid to replace the mapping procedure used in this paper.

The proposed approach relies on refinement techniques to generate analysis-suitable mechanical models, specifically to ensure the linear independence of basis functions near extraordinary points. These refinement techniques lead to an increased number of degrees of freedom in regions where it is not needed for the accuracy of the numerical solution. Nowadays, there is ongoing research focused on constructing analysis-suitable basis functions that achieve linear independence near extraordinary points without relying on refinement, such as the recent work by Takacs and Toshniwal [57]. Therefore, investigating these new approaches for constructing basis functions in the proposed approach could potentially yield mechanical models with fewer degrees of freedom, while still ensuring suitability for analysis.

This paper primarily focuses on solving engineering problems over two-phase domains. Nevertheless, it is worth noting that the methodology can be extended to address domains with more than two phases. The representation of such domains using the level-set method is well-established in the literature and is achieved by employing multiple level-set functions simultaneously to distinguish between different phases [58]. While the untrimming procedure in this paper already encompasses patches that accommodate exactly two phases, this extension should focus on enriching the trimming cases presented in Table 1 with trimmed patches that accommodate 3 and 4 different phases.

One of the important aspects that should be examined to assess the effectiveness of the proposed approach is the conditionality of the obtained linear systems. As this problem often arises in approaches involving enhancing the integration near the interfaces, linear systems with high condition numbers may be obtained when Bézier patches with very small sizes are generated near the interface. In this work, the Laplacian smoothing scheme presented in Section 2 is adopted as a remedy for this problem. However, a broader study should be conducted to examine different Laplacian smoothing schemes and their influence on the conditionality of the

obtained linear systems. Note that each Bézier patch along the interface layers is defined by 4 layers of control points surrounding the interface curve due to its bi-cubic representation, while the two closest layers to the interface are separated with zero-knots intervals. The conditionality study might examine the effect of choosing different weights for the different layers on the element sizes near the interface and by that on the conditionality of the obtained linear systems. Additionally, choosing different Laplacian smoothing schemes for the geometrical and mechanical models should be considered, along with the effect of including the interface control point in the Laplacian smoothing procedure of the geometrical model.

One major advantage of the proposed approach is its spline-based interface representation, which is beneficial for both geometric modeling and mechanical simulations. Spline-based representations are widely used in most CAD applications nowadays, making the proposed approach CAD-compatible and eliminating the need for transition procedures between spline-based CAD models and the different representations used for simulations or other purposes. This eliminates the inaccuracies typically associated with such transition procedures. Furthermore, the explicit feature of the interface allows for a straightforward control on its physical characteristics, such as curvature, perimeter, and slope.

7. Conclusion

This paper presents a new approach for solving two-phase engineering problems. This approach is based on untrimming techniques to generate unstructured spline-based representations to define mechanical and geometrical models for geometrically complex two-phase domains. The interface between the phases is precisely and explicitly defined by a set of cubic B-spline curves. This allows for the evaluation of physical measures on points that are precisely located along the interface. The mechanical model precisely replicates the geometrical model using h-refinement techniques. The IGA approach is adopted to solve the engineering problem where all required integrations are performed on the exact physical domain without the need to use approximation techniques. Algorithms that allow for changing the continuity of the solution field between the phases are provided. An analytical sensitivity analysis for generating the geometrical and mechanical models is developed to allow for solving engineering problems with moving interfaces following the gradient-based approach.

The applicability of the proposed approach is demonstrated in four different engineering problems. The results presented in Section 5 reveal the ability of the proposed approach to solve two-phase engineering problems while accurately representing the interface. Specifically, the third and fourth examples demonstrate the proposed approach's ability to address moving interface problems. In particular, the fourth example illustrates the proposed approach's ability to solve problems requiring different continuity conditions between phases for different types of unknown solution fields. This paves the way for further studies to examine the advantages of the proposed approach in each two-phase engineering problem separately.

Acknowledgments The author would like to thank Prof. Bert Jüttler and associate Prof. Oded Amir for the informative discussions and valuable suggestions during the work on this paper.

Funding No funding was received to assist with the preparation of this manuscript.

Conflict of interest The author declares that he has no conflict of interest.

References

- [1] J. B. Collins, H. Levine, Diffuse interface model of diffusion-limited crystal growth, *Physical Review B* 31 (1985) 6119. doi:<https://doi.org/10.1103/PhysRevB.31.6119>.
- [2] I. Singer-Loginova, H. Singer, The phase field technique for modeling multiphase materials, *Reports on progress in physics* 71 (2008) 106501. doi:10.1088/0034-4885/71/10/106501.
- [3] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3, Cambridge university press, 1999. doi:DOI:<https://doi.org/10.1017/S0263574799212404>.
- [4] J. A. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, *Journal of computational physics* 169 (2001) 503–555. doi:<https://doi.org/10.1006/jcph.2000.6657>.
- [5] R. Saurel, C. Pantano, Diffuse-interface capturing methods for compressible two-phase flows, *Annual Review of Fluid Mechanics* 50 (2018) 105–130. doi:<https://doi.org/10.1146/annurev-fluid-122316-050109>.
- [6] B. Bourdin, A. Chambolle, Design-dependent loads in topology optimization, *ESAIM: Control, Optimisation and Calculus of Variations* 9 (2003) 19–48. doi:<https://doi.org/10.1051/cocv:2002070>.
- [7] B. Bourdin, A. Chambolle, The phase-field method in optimal design, in: *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials*, Springer, 2006, pp. 207–215. doi:https://doi.org/10.1007/1-4020-4752-5_21.
- [8] R. Folch, J. Casademunt, A. Hernández-Machado, L. Ramirez-Piscina, Phase-field model for hele-shaw flows with arbitrary viscosity contrast. i. theoretical approach, *Physical Review E* 60 (1999) 1724. doi:<https://doi.org/10.1103/PhysRevE.60.1724>.
- [9] I. Aranson, V. Kalatsky, V. Vinokur, Continuum field description of crack propagation, *Physical review letters* 85 (2000) 118. doi:<https://doi.org/10.1103/PhysRevLett.85.118>.
- [10] T. Belytschko, T. Black, Elastic crack growth in finite elements with minimal remeshing, *International journal for numerical methods in engineering* 45 (1999) 601–620. doi:[https://doi.org/10.1002/\(SICI\)1097-0207\(19990620\)45:5<601::AID-NME598>3.0.CO;2-S](https://doi.org/10.1002/(SICI)1097-0207(19990620)45:5<601::AID-NME598>3.0.CO;2-S).

- [11] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, *International journal for numerical methods in engineering* 46 (1999) 131–150. doi:[https://doi.org/10.1002/\(SICI\)1097-0207\(19990910\)46:1%3C131::AID-NME726%3E3.0.CO;2-J](https://doi.org/10.1002/(SICI)1097-0207(19990910)46:1%3C131::AID-NME726%3E3.0.CO;2-J).
- [12] E. Burman, M. A. Fernández, P. Hansbo, Continuous interior penalty finite element method for Oseen’s equations, *SIAM journal on numerical analysis* 44 (2006) 1248–1274. doi:<https://doi.org/10.1137/040617686>.
- [13] E. Burman, S. Claus, P. Hansbo, M. G. Larson, A. Massing, CutFEM: discretizing geometry and partial differential equations, *International Journal for Numerical Methods in Engineering* 104 (2015) 472–501. doi:<https://doi.org/10.1002/nme.4823>.
- [14] C. H. Villanueva, K. Maute, CutFEM topology optimization of 3D laminar incompressible flow problems, *Computer Methods in Applied Mechanics and Engineering* 320 (2017) 444–473. doi:<https://doi.org/10.1016/j.cma.2017.03.007>.
- [15] C. S. Andreasen, M. O. Elingaard, N. Aage, Level set topology and shape optimization by density methods using cut elements with length scale control, *Structural and Multidisciplinary Optimization* (2020) 1–23. doi:<https://doi.org/10.1007/s00158-020-02527-1>.
- [16] J. Chessa, T. Belytschko, An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension, *International journal for numerical methods in engineering* 58 (2003) 2041–2064. doi:<https://doi.org/10.1002/nme.946>.
- [17] H. Sauerland, T.-P. Fries, The stable XFEM for two-phase flows, *Computers & Fluids* 87 (2013) 41–49. doi:<https://doi.org/10.1016/j.compfluid.2012.10.017>.
- [18] M. Lawry, K. Maute, Level set topology optimization of problems with sliding contact interfaces, *Structural and Multidisciplinary Optimization* 52 (2015) 1107–1119. doi:<https://doi.org/10.1007/s00158-015-1301-5>.
- [19] T.-P. Fries, A corrected XFEM approximation without problems in blending elements, *International Journal for Numerical Methods in Engineering* 75 (2008) 503–532. doi:<https://doi.org/10.1002/nme.2259>.
- [20] P. Hansbo, M. G. Larson, S. Zahedi, A cut finite element method for a Stokes interface problem, *Applied Numerical Mathematics* 85 (2014) 90–114. doi:<https://doi.org/10.1016/j.apnum.2014.06.009>.
- [21] S. Claus, P. Kerfriden, A CutFEM method for two-phase flow problems, *Computer Methods in Applied Mechanics and Engineering* 348 (2019) 185–206. doi:<https://doi.org/10.1016/j.cma.2019.01.009>.
- [22] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian–Eulerian Methods, *Encyclopedia of computational mechanics* (2004).

- [23] P. Kagan, A. Fischer, P. Z. Bar-Yoseph, New B-spline finite element approach for geometrical design and mechanical analysis, *International Journal for Numerical Methods in Engineering* 41 (1998) 435–458. doi:[https://doi.org/10.1002/\(SICI\)1097-0207\(19980215\)41:3<435::AID-NME292>3.0.CO;2-U](https://doi.org/10.1002/(SICI)1097-0207(19980215)41:3<435::AID-NME292>3.0.CO;2-U).
- [24] P. Kagan, A. Fischer, P. Z. Bar-Yoseph, Mechanically based models: Adaptive refinement for B-spline finite element, *International Journal for Numerical Methods in Engineering* 57 (2003) 1145–1175. doi:<https://doi.org/10.1002/nme.717>.
- [25] T. J. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer methods in applied mechanics and engineering* 194 (2005) 4135–4195. doi:<https://doi.org/10.1016/j.cma.2004.10.008>.
- [26] J. A. Cottrell, T. J. Hughes, Y. Bazilevs, *Isogeometric analysis: toward integration of CAD and FEA*, John Wiley & Sons, 2009.
- [27] H.-J. Kim, Y.-D. Seo, S.-K. Youn, Isogeometric analysis for trimmed CAD surfaces, *Computer Methods in Applied Mechanics and Engineering* 198 (2009) 2982–2995. doi:<https://doi.org/10.1016/j.cma.2009.05.004>.
- [28] F. Scholz, A. Mantzaflaris, B. Jüttler, First order error correction for trimmed quadrature in isogeometric analysis, in: *Advanced Finite Element Methods with Applications: Selected Papers from the 30th Chemnitz Finite Element Symposium 2017* 30, Springer, 2019, pp. 297–321. doi:doi.org/10.1007/978-3-030-14244-5_15.
- [29] M. Meßmer, T. Teschemacher, L. F. Leidinger, R. Wüchner, K.-U. Bletzinger, Efficient CAD-integrated isogeometric analysis of trimmed solids, *Computer Methods in Applied Mechanics and Engineering* 400 (2022) 115584. doi:<https://doi.org/10.1016/j.cma.2022.115584>.
- [30] M. Loibl, L. Leonetti, A. Reali, J. Kiendl, Patch-wise quadrature of trimmed surfaces in isogeometric analysis, *arXiv preprint arXiv:2306.14521* (2023).
- [31] X. Wei, B. Marussig, P. Antolin, A. Buffa, Immersed boundary-conformal isogeometric method for linear elliptic problems, *Computational Mechanics* 68 (2021) 1385–1405. doi:<https://doi.org/10.1007/s00466-021-02074-6>.
- [32] Q. X. Lieu, J. Lee, A multi-resolution approach for multi-material topology optimization based on isogeometric analysis, *Computer Methods in Applied Mechanics and Engineering* 323 (2017) 272–302. doi:<https://doi.org/10.1016/j.cma.2017.05.009>.
- [33] B. S. Hosseini, S. Turek, M. Möller, C. Palmes, Isogeometric analysis of the Navier–Stokes–Cahn–Hilliard equations with application to incompressible two-phase flows, *Journal of Computational Physics* 348 (2017) 171–194. doi:<https://doi.org/10.1016/j.jcp.2017.07.029>.

- [34] J. Yan, S. Lin, Y. Bazilevs, G. Wagner, Isogeometric analysis of multi-phase flows with surface tension and with application to dynamics of rising bubbles, *Computers & Fluids* 179 (2019) 777–789. doi:<https://doi.org/10.1016/j.compfluid.2018.04.017>.
- [35] L. Noël, M. Schmidt, K. Doble, J. Evans, K. Maute, XIGA: An extended isogeometric analysis approach for multi-material problems, *Computational Mechanics* 70 (2022) 1281–1308. doi:<https://doi.org/10.1007/s00466-022-02200-y>.
- [36] M. Schmidt, L. Noël, K. Doble, J. A. Evans, K. Maute, Extended isogeometric analysis of multi-material and multi-physics problems using hierarchical B-splines, *Computational Mechanics* 71 (2023) 1179–1203. doi:<https://doi.org/10.1007/s00466-023-02306-x>.
- [37] L. Noël, M. Schmidt, C. Messe, J. Evans, K. Maute, Adaptive level set topology optimization using hierarchical B-splines, *Structural and Multidisciplinary Optimization* 62 (2020) 1669–1699. doi:<https://doi.org/10.1007/s00158-020-02584-6>.
- [38] S. Khatir, D. Boutchicha, C. Le Thanh, H. Tran-Ngoc, T. Nguyen, M. Abdel-Wahab, Improved ANN technique combined with Jaya algorithm for crack identification in plates using XIGA and experimental analysis, *Theoretical and Applied Fracture Mechanics* 107 (2020) 102554. doi:<https://doi.org/10.1016/j.tafmec.2020.102554>.
- [39] J. Gu, T. Yu, S. Tanaka, H. Yuan, T. Q. Bui, et al., Crack growth adaptive XIGA simulation in isotropic and orthotropic materials, *Computer Methods in Applied Mechanics and Engineering* 365 (2020) 113016. doi:<https://doi.org/10.1016/j.cma.2020.113016>.
- [40] W. Qiu, Q. Wang, L. Gao, Z. Xia, Evolutionary topology optimization for continuum structures using isogeometric analysis, *Structural and Multidisciplinary Optimization* 65 (2022) 1–17. doi:<https://doi.org/10.1007/s00158-022-03215-y>.
- [41] W. Qiu, Q. Wang, L. Gao, Z. Xia, Stress-based evolutionary topology optimization via XIGA with explicit geometric boundaries, *International Journal of Mechanical Sciences* 256 (2023) 108512. doi:<https://doi.org/10.1016/j.ijmecsci.2023.108512>.
- [42] G. Sangalli, T. Takacs, R. Vázquez, Unstructured spline spaces for isogeometric analysis based on spline manifolds, *Computer Aided Geometric Design* 47 (2016) 61–82. doi:<https://doi.org/10.1016/j.cagd.2016.05.004>.
- [43] D. Toshniwal, H. Speleers, T. J. Hughes, Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations, *Computer Methods in Applied Mechanics and Engineering* 327 (2017) 411–458. doi:<https://doi.org/10.1016/j.cma.2017.06.008>.
- [44] M. Kapl, G. Sangalli, T. Takacs, Isogeometric analysis with C^1 functions on planar, unstructured quadrilateral meshes, *The SMAI journal of computational mathematics* 5 (2019) 67–86. doi:[10.5802/smai-jcm.52](https://doi.org/10.5802/smai-jcm.52).

- [45] E. Shakour, O. Amir, Topology optimization with precise evolving boundaries based on IGA and untrimming techniques, *Computer Methods in Applied Mechanics and Engineering* 374 (2021) 113564. doi:<https://doi.org/10.1016/j.cma.2020.113564>.
- [46] E. Shakour, O. Amir, Stress-constrained topology optimization with precise and explicit geometric boundaries, *Structural and Multidisciplinary Optimization* 65 (2022) 1–21. doi:<https://doi.org/10.1007/s00158-021-03115-7>.
- [47] A. Gomes, I. Voiculescu, J. Jorge, B. Wyvill, C. Galbraith, *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*, Springer London, 2009.
- [48] R. Wenger, *Isosurfaces: Geometry, Topology, and Algorithms*, An A K Peters Book, CRC Press, 2013.
- [49] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, T. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, *Computer Methods in Applied Mechanics and Engineering* 254 (2013) 197–221. doi:<https://doi.org/10.1016/j.cma.2012.11.001>.
- [50] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, *ACM transactions on graphics (TOG)* 22 (2003) 477–484. doi:<https://doi.org/10.1145/882262.882295>.
- [51] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, T. J. Hughes, Isogeometric finite element data structures based on bézier extraction of T-splines, *International Journal for Numerical Methods in Engineering* 88 (2011) 126–156. doi:<https://doi.org/10.1002/nme.3167>.
- [52] M. Scott, X. Li, T. Sederberg, T. Hughes, Local refinement of analysis-suitable T-splines, *Computer Methods in Applied Mechanics and Engineering* 213-216 (2012) 206–222. doi:<https://doi.org/10.1016/j.cma.2011.11.022>.
- [53] T. Takacs, Approximation properties over self-similar meshes of curved finite elements and applications to subdivision based isogeometric analysis, *arXiv preprint arXiv:2307.10403* (2023). doi:<https://doi.org/10.48550/arXiv.2307.10403>.
- [54] N. Sukumar, D. Chopp, N. Moës, T. Belytschko, Modeling holes and inclusions by level sets in the extended finite-element method, *Computer Methods in Applied Mechanics and Engineering* 190 (2001) 6183–6200. doi:[https://doi.org/10.1016/S0045-7825\(01\)00215-8](https://doi.org/10.1016/S0045-7825(01)00215-8).
- [55] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, *Journal of Computational Physics* 194 (2004) 363–393. doi:<https://doi.org/10.1016/j.jcp.2003.09.032>.

- [56] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, *International journal for numerical methods in engineering* 24 (1987) 359–373. doi:<https://doi.org/10.1002/nme.1620240207>.
- [57] T. Takacs, D. Toshniwal, Almost-C1 splines: Biquadratic splines on unstructured quadrilateral meshes and their application to fourth order problems, *Computer Methods in Applied Mechanics and Engineering* 403 (2023) 115640. doi:<https://doi.org/10.1016/j.cma.2022.115640>.
- [58] Y. Wang, Z. Luo, Z. Kang, N. Zhang, A multi-material level set-based topology and shape optimization method, *Computer Methods in Applied Mechanics and Engineering* 283 (2015) 1570–1586. doi:<https://doi.org/10.1016/j.cma.2014.11.002>.

Appendix A. Control points locations

This appendix refers to Step 4 in Section 2.1, which deals with determining the locations of the control points $\mathbf{P}^G$. Since the geometry in this paper is primarily developed based on a level-set function, the locations of $\mathbf{P}^G$ are represented as functions of the control parameters of the level-set function α , i.e., $\mathbf{P}^G(\alpha)$. The example in Section 2 is used to present the computation procedure. Figure A.32 shows the physical domain of S^0 with cross marks that are located at the corners of the Bézier patches. Computing $\mathbf{P}^G$ starts by determining the values of the level-set function, ϕ , at these points. This computation is done as follows:

1. Compute the level-set function values at the inner corners of the patches in the physical domain. These points are highlighted with blue cross marks in Figure A.32. This computation depends on the level-set function representation. In this paper, the level-set function is represented as a B-spline surface; therefore, the value of the level-set function at a certain point i with the parametric coordinates of (ξ_i, η_i) is computed by:

$$\phi_i = \sum_{j=1}^{N_\alpha} \alpha_j B_j(\xi_i, \eta_i), \quad (\text{A.1})$$

where N_α is the number of control parameters of the level-set function and B_j is the basis function that corresponds to the control parameter α_j

2. To avoid untrimming complications along the boundary of the design domain, assign all the evaluation points that are located on the boundary of the design domain with ϕ values that correspond to the adjacent layer of evaluation points to the interior of the design domain. I.e. in Figure A.32, all the red cross mark points are assigned with the ϕ values that correspond to the blue cross mark points in the adjacent layer.

Afterwards, the control points, $\mathbf{P}^G$, of the trimmed patches are located according to the trimming manner across the patch. For each trimmed edge in a trimmed patch we define a weighting parameter w_{i-j} such as:

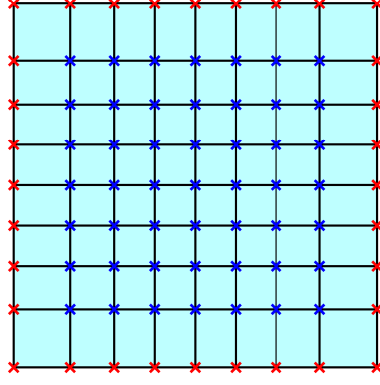


Figure A.32: Physical domain of S^0 - Demonstrating the evaluation points of the level-set function.

$$w_{i-j} = 1 - \frac{\phi_i}{\phi_i - \phi_j}, \quad (\text{A.2})$$

where i and j are the indices of the vertices at the end of the trimmed edge. ϕ_i and ϕ_j represent the values of the level-set function at the vertices i and j . Since this paper deals with quadrilateral faces with four edges that could be trimmed, four weighting parameters could be defined: w_{1-2} , w_{2-3} , w_{3-4} and w_{1-4} as shown in Figure A.33. If an edge is not trimmed, then the associated weighting parameter is not defined. Note that the direction of computing the weighting parameter is important while $w_{i-j} = 1 - w_{j-i}$.

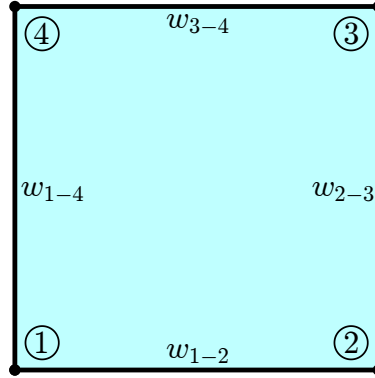


Figure A.33: Demonstrating the weighting parameters w_{i-j} .

Having the weighting parameters for a trimmed patch, the control points $\mathbf{P}^G$ that correspond to the same patch are computed as follows:

$$P_{x_i}^G = \sum_{j=1}^4 f_{ij}^x(w_{1-2}, w_{2-3}, w_{3-4}, w_{1-4}) X_j, \quad (\text{A.3a})$$

$$P_{y_i}^G = \sum_{j=1}^4 f_{ij}^y(w_{1-2}, w_{2-3}, w_{3-4}, w_{1-4}) Y_j, \quad (\text{A.3b})$$

where

$$f_{ij}^x = a_{ij}^x + b_{ij}^x w_{1-2} + c_{ij}^x w_{2-3} + d_{ij}^x w_{3-4} + e_{ij}^x w_{1-4}; \quad (\text{A.4a})$$

$$f_{ij}^y = a_{ij}^y + b_{ij}^y w_{1-2} + c_{ij}^y w_{2-3} + d_{ij}^y w_{3-4} + e_{ij}^y w_{1-4}; \quad (\text{A.4b})$$

X_j and Y_j are the coordinates of the control points corresponding to the corner of the face. The parameters a_{ij}^x , b_{ij}^x , c_{ij}^x , d_{ij}^x , e_{ij}^x , a_{ij}^y , b_{ij}^y , c_{ij}^y , d_{ij}^y and e_{ij}^y differs for each case and are chosen in a way that does not cause an overlapping between the faces in the mesh. In the following, we provide the non-zero values of these parameters for each case:

Case 1:

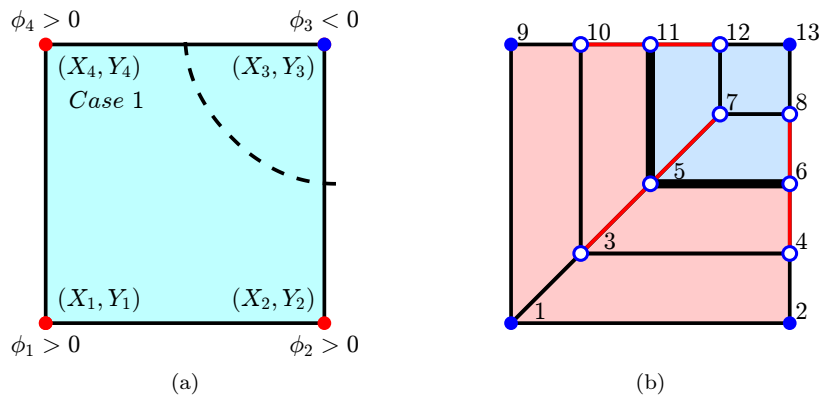


Figure A.34: Case 1 - Locating the control points $\mathbf{P}^G$.

a_{ij}^x	1	2	3	3	3	4	5	6	7	7	7	8	9	10	11	12	12	13
i	1	2	3	3	4	5	6	7	7	8	9	10	11	12	12	13		
j	1	2	1	2	2	2	4	2	2	3	4	2	4	4	4	3	4	3
v	1	1	0.5	0.25	0.25	1	0.5	0.5	1	0.33	0.33	0.33	1	1	1	0.5	0.5	1

d_{ij}^x															
i	3	3	5	5	7	7	10	11	11	12	12				
j	3	4	3	4	3	4	3	4	4	3	4				
v	0.25	-0.25	0.5	-0.5	0.33	-0.33	0.5	-0.5	1	-1	0.5	-0.5			

a_{ij}^y	1	2	3	4	5	6	7	8	9	10	11	12	13
i	1	2	3	4	5	6	7	8	9	10	11	12	13
j	1	2	1	3	2	3	3	3	4	3	3	3	3
v	1	1	0.5	0.5	0.5	0.5	1	1	1	1	1	1	1

c_{ij}^y																
i	3	3	4	4	5	5	6	6	7	7	8	8				
j	2	3	2	3	2	3	2	3	2	3	2	3				
v	0.25	-0.25	0.5	-0.5	0.5	-0.5	1	-1	0.33	-0.33	0.5	-0.5				

Case 2:

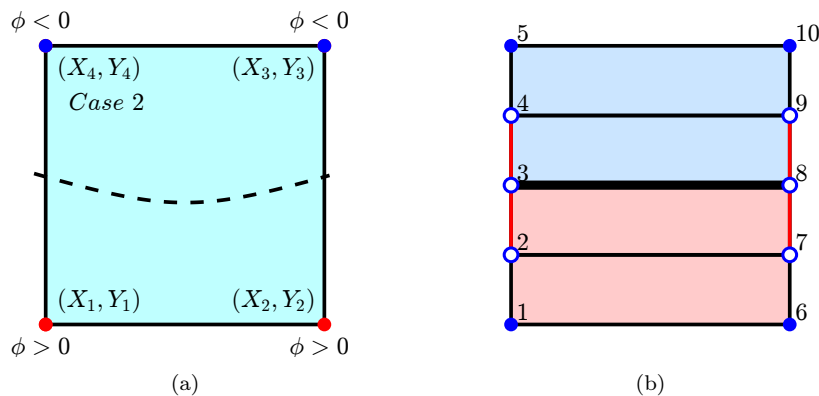


Figure A.35: Case 2 - Locating the control points $\mathbf{P}^G$.

a_{ij}^x										
i	1	2	3	4	5	6	7	8	9	10
j	1	1	1	1	4	2	2	2	2	3
v	1	1	1	1	1	1	1	1	1	1

a_{ij}^y										
i	1	2	2	3	4	5	6	7	7	8
j	1	1	4	4	4	4	2	2	3	3
v	1	0.5	0.5	1	1	1	1	0.5	0.5	1

c_{ij}^y							
i	7	7	8	8	9	9	
j	2	3	2	3	2	3	
v	0.5	-0.5	1	-1	0.5	-0.5	

e_{ij}^y							
i	2	2	3	3	4	4	
j	1	4	1	4	1	4	
v	0.5	-0.5	1	-1	0.5	-0.5	

Case 3:

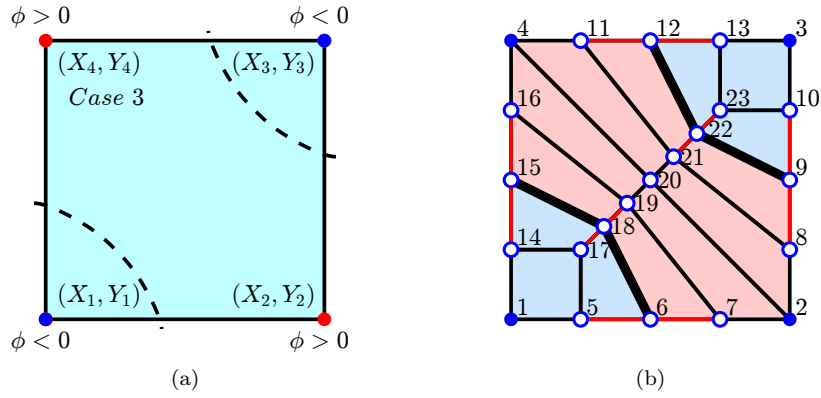


Figure A.36: Case 3 - Locating the control points P^G .

a_{ij}^x																				
i	1	2	3	4	5	5	6	7	8	9	10	11	12	13	13	14	15	16	17	17
j	1	2	3	4	1	2	2	2	2	2	2	4	4	3	4	1	1	1	1	2
v	1	1	1	1	0.5	0.5	1	1	1	1	1	1	1	0.5	0.5	1	1	1	0.67	0.33

b_{ij}^x																				
i	5	5	6	6	7	7	17	17	18	18	19	19	20	20	21	21				
j	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2				
v	0.5	-0.5	1	-1	0.5	-0.5	0.33	-0.33	0.5	-0.5	0.38	-0.38	0.25	-0.25	0.12	-0.12				

d_{ij}^x																				
i	11	11	12	12	13	13	19	19	20	20	21	21	22	22	23	23				
j	3	4	3	4	3	4	3	4	3	4	3	4	3	4	3	4				
v	0.5	-0.5	1	-1	0.5	-0.5	0.12	-0.12	0.25	-0.25	0.38	-0.38	0.5	-0.5	0.33	-0.33				

a_{ij}^y																				
i	1	2	3	4	5	6	7	8	8	9	10	11	12	13	14	14	15	16	17	17
j	1	2	3	4	1	1	1	2	3	3	3	3	3	3	1	4	4	4	1	4
v	1	1	1	1	1	1	1	0.5	0.5	1	1	1	1	1	0.5	0.5	1	1	0.67	0.33

c_{ij}^y																				
i	8	8	9	9	10	10	19	19	20	20	21	21	22	22	23	23				
j	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3				
v	0.5	-0.5	1	-1	0.5	-0.5	0.12	-0.12	0.25	-0.25	0.38	-0.38	0.5	-0.5	0.33	-0.33				

e_{ij}^y																				
i	14	14	15	15	16	16	17	17	18	18	19	19	20	20	21	21				
j	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4				
v	0.5	-0.5	1	-1	0.5	-0.5	0.33	-0.33	0.5	-0.5	0.38	-0.38	0.25	-0.25	0.12	-0.12				

Case 4:

a_{ij}^x										
i	1	2	3	4	5	6	7	8	9	10
j	1	1	1	1	4	2	2	2	2	3
v	1	1	1	1	1	1	1	1	1	1

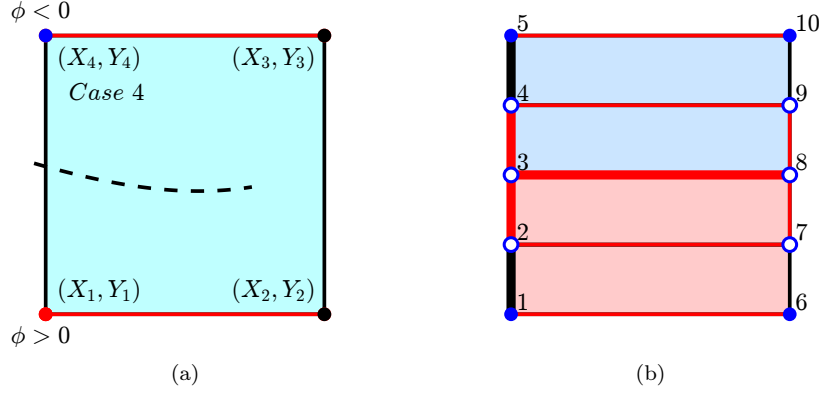


Figure A.37: Case 4 - Locating the control points $\mathbf{P}^G$.

a_{ij}^y												
i	1	2	2	3	4	5	6	7	7	8	9	10
j	1	1	4	4	4	4	2	2	3	3	3	3
v	1	0.5	0.5	1	1	1	1	0.5	0.5	1	1	1

e_{ij}^y												
i	2	2	3	3	4	4	7	7	8	8	9	9
j	1	4	1	4	1	4	2	3	2	3	2	3
v	0.5	-0.5	1	-1	0.5	-0.5	0.5	-0.5	1	-1	0.5	-0.5

Appendix B. Boundary representation

A significant advantage of the proposed approach is its ability to represent explicitly and precisely the boundaries of the phases in the mechanical and the geometrical models as a set of B-spline curves. Corresponding to the bi-cubic spline-based meshes that are used in this paper, all the B-spline boundary curves will be of degree 3. These boundaries are described as periodic curves whose control points and knots vectors are assembled according to the boundary polygons of the unstructured mesh.

We demonstrate the procedure for extracting the boundary curves using the example in Figure B.38. Black and red lines in Figures B.38a and B.38b represent unit and zero knot intervals. The boundary polygons are drawn with thick lines. Boundary control points are colored blue, and all the other control points are colored black. This example has two boundary curves. The inner boundary curve, that constitutes the interface between the phases, and the outer boundary curve which is also the boundary of the design domain. We show how these two curves are extracted, while first, we focus on extracting the inner boundary curve.

Initially, the control points and knots intervals of each boundary are sequentially collected in vectors $\mathbf{P}^c$ and $\mathbf{ed}$. The number of control points in $\mathbf{P}^c$ is denoted by N . The order of control points of each boundary curve is depicted in Figure B.38b. For the inner boundary curve, these vectors are:

$$\mathbf{P}^c = [P_1^c \ P_2^c \ P_3^c \ P_4^c \ P_5^c \ P_6^c \ P_7^c \ P_8^c \ P_9^c \ P_{10}^c \ P_{11}^c \ P_{12}^c],$$

$$\mathbf{ed} = [ed_1 \ ed_2 \ ed_3 \ ed_4 \ ed_5 \ ed_6 \ ed_7 \ ed_8 \ ed_9 \ ed_{10} \ ed_{11} \ ed_{12}] = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1].$$

Afterwards, denoting the vector of control points and the knots vector of a boundary curve by $\tilde{\mathbf{P}}^c$ and $\tilde{\mathbf{k}}$, respectively, these vectors are assembled according to the following algorithm:

Algorithm 5: The algorithm for assembling the control points and the knots vectors for a single boundary curve.

Data: $\mathbf{P}^c, \mathbf{ed}, N$

Result: $\tilde{\mathbf{P}}^c, \tilde{\mathbf{k}}$

```

7   $k_1 = -ed_N - ed_{N-1}$ 
8   $k_2 = -ed_N$ 
9   $k_3 = 0$ 
10 for  $i = 1 : N + 3$  do
11   if  $i \leq N$  then
12      $\tilde{P}_i^c = P_i^c$ ;
13      $k_{i+3} = k_{i+2} + ed_i$ ;
14   else
15      $\tilde{P}_i^c = P_{i-N}^c$ ;
16      $k_{i+3} = k_{i+2} + ed_{i-N}$ ;
17  $k_{N+7} = k_{N+6} + ed_4$ ;

```

After applying Algorithm 5 to $\mathbf{P}^c$ and $\mathbf{ed}$ of the inner boundary polygon, the obtained control points and knots vector of the boundary curve are:

$$\tilde{\mathbf{P}}^c = [P_1^c \ P_2^c \ P_3^c \ P_4^c \ P_5^c \ P_6^c \ P_7^c \ P_8^c \ P_9^c \ P_{10}^c \ P_{11}^c \ P_{12}^c \ P_1^c \ P_2^c \ P_3^c],$$

$$\mathbf{k} = [-2 \ -1 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16].$$

Applying the same procedure for the outer boundary polygon the following vectors are obtained:

$$\mathbf{P}^c = [P_1^c \ P_2^c \ P_3^c \ \dots \ P_{28}^c],$$

$$\mathbf{ed} = [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0].$$

$$\tilde{\mathbf{P}}^c = [P_1^c \ P_2^c \ P_3^c \ \dots \ P_{28}^c \ P_1^c \ P_2^c \ P_3^c]$$

$$\mathbf{k} = [-1 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 5 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 10 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 15 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20 \ 20 \ 21 \ 22 \ 23].$$

The B-spline boundary curves are highlighted in green on top of the physical domain in Figure B.38c.

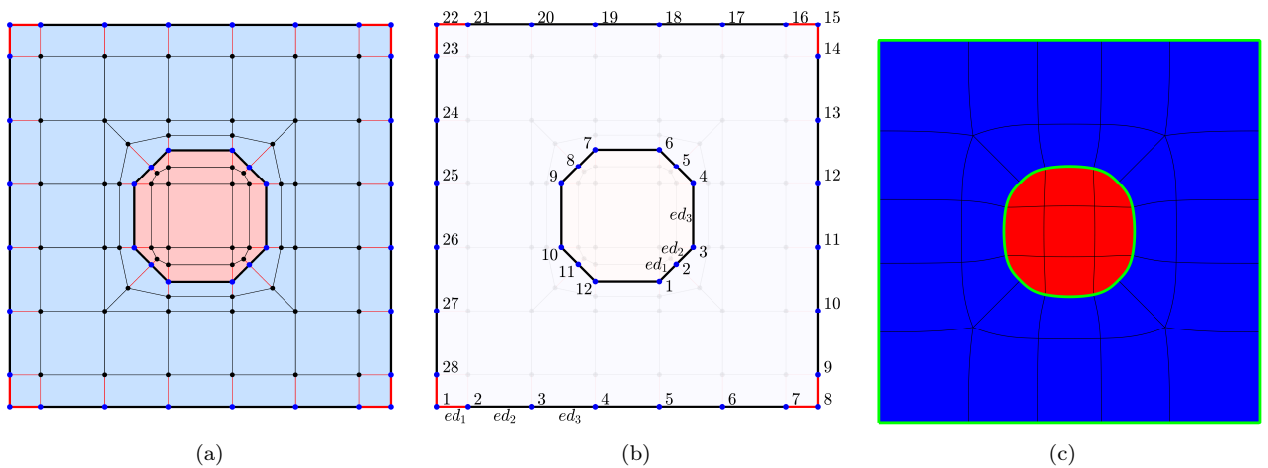


Figure B.38: An example to demonstrate the procedure of extracting the boundaries of the phases.