A novel formulation for the explicit discretization of evolving boundaries with application to topology optimisation

R.O.S.S. da Costa[a,*], S.T. Pinho[a]

[a] *Department of Aeronautics, South Kensington Campus, Imperial College London. London SW7 2AZ, United Kingdom*

Abstract

Evolving boundaries are an intrinsic part of many physical processes and numerical methods. Most efforts to model evolving boundaries rely on implicit schemes, such as the level-set method (LSM). LSM provides the means to efficiently model the evolution of a boundary, but lacks the ability to transmit information or provide information directly at the boundary. Explicit alternatives based on remeshing or partial-remeshing are often computationally expensive and inherently complex to implement. This work proposes a solution to this dichotomy: a novel finite element method (FEM) based formulation capable of explicitly discretizing moving boundaries in an accurate and numerically-efficient way. It couples the floating node method (FNM) with LSM for the first time, which yield a methodology suitable for implementation as user-element in a generic FEM package. The explicitly discretized boundary allows for a new velocity-extension methodology, and a new LSM-reinitialization procedure, which show benefits in accuracy and efficiency. The potential of this formulation is showcased within topology optimisation, showing greater geometrical accuracy and improvements in the optimum solution attained when compared to implicit methods.

Keywords: evolving boundary, explicit discretization, conforming mesh, floating node method, level set method, topology optimization

1. Introduction

1.1. Background

An evolving boundary is an interface between two materials, or two phases of the same material, which is moving through space. These moving boundaries can exist in two dimensions (2D) (e.g. a circle increasing in diameter over time) or in three dimensions (3D) (e.g. a balloon inflating). Evolving boundaries are ubiquitous; some of the most relevant areas of study featuring evolving boundaries include: fluid-structure interaction [1]; multi-phase flow and multi-material deformable bodies [2]; additive manufacturing [3, 4]; solidification and phase change [5]; fracture mechanics [6, 7]; and topology optimisation [8, 9].

[*] Corresponding author
Email address: r.costa18@imperial.ac.uk (R.O.S.S. da Costa)

Studying physical processes which involve evolving boundaries is challenging. The strategies to model evolving boundaries can be grossly categorized into: (i) explict — if the method captures the boundary explicitly and retains points/nodes in the moving boundary [10–12]; or (ii) implicit — if the method, in contrast, resorts to some alternative and virtual way of capturing the position of the boundary without placing any points directly on it [8, 13, 14].

The level-set method (LSM) is a method used to implicitly represent a moving boundary. It uses a level-set (LS) function from which one can extract all the necessary information to locate and evolve the boundary through time, without the need for explicit definition. LSM has been implemented extensively in the literature; some examples include multi-phase flow and multi-material problems [2, 15], topology optimisation [16–21], and solidification studies [5]. Traditionally, LSM uses a finite-difference scheme on a structured grid to model the LS field [8, 15, 22]. However, there are several implementations in the literature of LSM within the finite element method (FEM) [2, 5, 17, 23, 24].

LSM has the ability to very efficiently and simply model processes which involve moving boundaries, by manipulating the LS field. However, when there is a need to transmit information across the boundary (such as in fluid-structure interaction [1]) or to obtain information from the boundary (such as in topology optimisation [8]), LSM does not provide any ready and accurate way to achieve this. In fact, implicit methods lack these features by design in order to be more efficient.

Explicit tracking of an evolving boundary in a FEM mesh is often achieved with remeshing [1, 25, 26]. Remeshing is a method which consists of deleting and creating a new mesh after every geometrical evolution. While this method provides an explicit discretization of the boundary at every time-step, it becomes inefficient, as the geometry gets larger and more complex, to recreate the mesh at every time-step [1, 10]. With remeshing, complex data-structure handling schemes are often required to handle the mesh changes [27]. From the point of view of implementation in a typical FEM package, this approach requires changing the input file of the analysis at every iteration. Essentially, at every iteration, a different input file is generated and a new analysis is started as if it were a different numerical study. Remeshing therefore typically requires an external piece of software specifically to handle these changes in geometry (e.g. Abaqus uses iSight to sequentially generate new input files [28]). For these reasons, and given that the explicit representation of evolving boundaries within FEM is an active area of research, it is often the case that the added complexity and computational cost to feature the boundary explicitly in the FE model outweigh the advantages that such a feature provides.

To alleviate some of the computational cost of remeshing, some work has been done on partial remeshing [10, 11, 29]. This work consists of creating algorithms capable of retaining the conformal mesh representation of the boundary while forsaking the need to do a complete mesh overhaul at every evolution. This can be achieved by remeshing parts of the domain [10], or simply by translating nodes close to the boundary such that their new position is at the boundary [11, 29]. Partial remeshing potentially maintains the explicit-tracking advantages of remeshing at a lower computational cost, but can lead to unfeasible meshes or highly distorted elements [10, 11, 29]. Partial remeshing requires smoothing after the mesh modification, which can be a costly operation on the entire mesh.

An alternative strategy to represent explicitly an evolving boundary, presented by Nguyen and Kim [27], consists of trimming hexahedral meshes. This element-trimming approach relies on a background mesh of regular hexahedral elements, and on the information provided by the zero-level isosurface of the LS function, to break the hexahedral elements into multiple polyhedral elements. The polyhedral elements provide good accuracy for complex-geometry parts, but they also require complex schemes to obtain the shape functions, often requiring the division into simpler (tetrahedral) sub-domains [27]. Furthermore, the marching cubes algorithm used requires changing element connectivities — and, consequently, the analysis input file [27].

Topology optimisation is one of the areas of research where modelling evolving boundaries is paramount. By nature, topology optimisation is a numerical problem with the remit of finding the optimum distribution of mass within a domain [30, 31]. Density-based methods for topology optimisation consider the material densities of cells as the design variables for the optimisation algorithm. These methods do not yield a clear boundary between void and solid and are characterized by pixelated contours resulting from the 'ersatz' material definition [8, 31, 32]. An alternative, is to rely on LSM to overcome the pixelated-boundary limitation of the density-based methods. With LSM, there is a well-defined, clear boundary in the LS field [31]. However, if one uses a similar 'ersatz' approach to translate the LS field into the mechanical analysis model, the boundary representation retains the pixelated contours [8, 32].

To the best knowledge of the authors, there is no methodology in the literature capable of modelling evolving boundaries explicitly whose implementation can be readily carried out in typical FEM packages. This is the main motivation to the present work.

1.2. Floating node method

The floating node method (FNM) is a method developed by Chen et al. [6] to overcome the limitations of the technologies available, at the time, to model crack propagation. It uses floating (extra) nodes which are not necessarily tied to a coordinate position (thus the designation 'floating'), see Figure 1. When modelling cracks, the edges and surfaces of elements get partitioned at specific positions. Floating degrees-of-freedom (DoF) can be assigned to these positions to create multiple sub-elements, defining the crack through the partitioning of the original element.

The solution to the FE problem requires computing the stiffness matrix of each sub-element separately and assembling them into the stiffness matrix of the parent element. In this way, FNM represents discontinuities locally and internally to an element, thereby allowing the method to be easily implemented in a typical FE software through user-element functionality. Furthermore, the need for floating nodes is specified *a priori*, therefore the elements connectivity do not change during the analysis — nor does the input file [6].

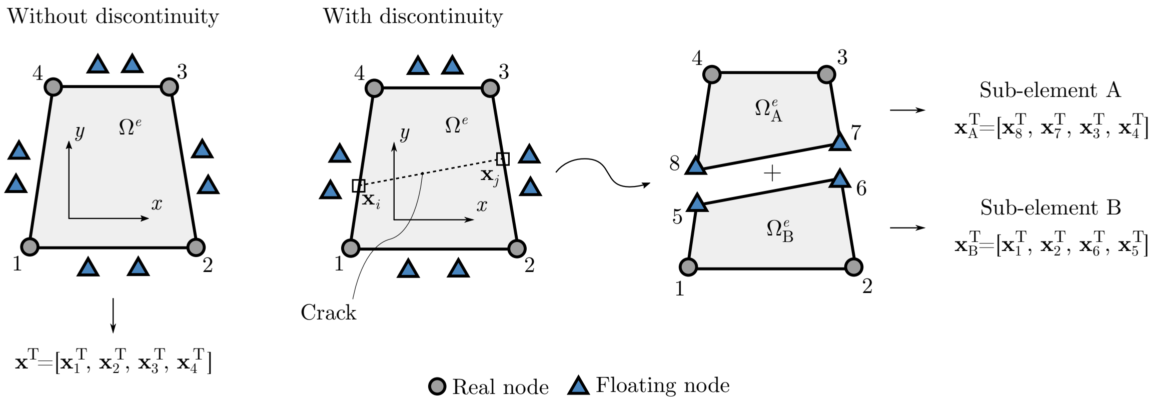

Without discontinuity
With discontinuity
4
3
1
2
y
x
Ω^e
$\mathbf{x}^{\mathrm{T}}$=[$\mathbf{x}_1^{\mathrm{T}}$, $\mathbf{x}_2^{\mathrm{T}}$, $\mathbf{x}_3^{\mathrm{T}}$, $\mathbf{x}_4^{\mathrm{T}}$]
$\mathbf{x}_i$
$\mathbf{x}_j$
Crack
Ω^e_{A}
Ω^e_{B}
5
6
7
8
+
Sub-element A
$\mathbf{x}_{\mathrm{A}}^{\mathrm{T}}$=[$\mathbf{x}_8^{\mathrm{T}}$, $\mathbf{x}_7^{\mathrm{T}}$, $\mathbf{x}_3^{\mathrm{T}}$, $\mathbf{x}_4^{\mathrm{T}}$]
Sub-element B
$\mathbf{x}_{\mathrm{B}}^{\mathrm{T}}$=[$\mathbf{x}_1^{\mathrm{T}}$, $\mathbf{x}_2^{\mathrm{T}}$, $\mathbf{x}_6^{\mathrm{T}}$, $\mathbf{x}_5^{\mathrm{T}}$]
● Real node
▲ Floating node

Figure 1: Floating node method for crack representation

FNM has proven to be a very powerful tool to model fracture [6, 7, 33, 34] and to realise polymorphic elements [35]. One can think of a crack as an example of an evolving boundary. Thus, the partitioning concept used for fracture mechanics can, potentially, be applied to model any evolving boundary. When modelling evolving boundaries, one can imagine that edges and surfaces also intersect the boundary, and consequently, can be partitioned to retain the boundary. Furthermore, floating nodes can be assigned to the different intersections, hence providing an explicit discretization of the interface.

1.3. Objective

The main objective of this work is to develop a new formulation that combines LSM and FNM for accurate and efficient representation of evolving boundaries within FEM, with the following original features:

- explicit discretization of the boundary without using remeshing (nor partial-remeshing) techniques or non-standard element formulations;
- simple element-partitioning scheme that leaves the initial input file unchanged during analysis, and makes the methodology suitable for implementation as user-element in a typical FEM package;
- a new heuristic approach to extend the boundary velocity that benefits from the explicit boundary discretization and is therefore more accurate; and
- a more efficient reinitialization procedure for the signed distance function that makes use of the explicit boundary discretization.

In this paper, we will also show how the proposed methodology can be readily used to solve a topology optimisation problem, with the following original features:

- a more streamlined optimisation workflow that does not require specific software to modify the input file of the model after every design change (as is the case with leading approaches in the literature, namely with the Abaqus and iSight workflow [28]); and

- an optimisation process that intends to bridge the gap between numerical and CAD design by accurately capturing every topological iteration — from initial to final design — more accurately and more efficiently.

1.4. Structure of the paper

Section 2 outlines the methodology implemented in this work. The LSM (Section 2.1) is briefly exposed, alongside the corresponding FEM discretization. The implementation of FNM as a means to explicitly discretize the implicit LSM-representation of the moving boundary in the numerical model is laid out in Section 2.2. Finally, the baseline and proposed methodologies are summarized in Section 2.3.

Section 4 contains the analysis of the reinitialization procedure using a simple test case (Section 4.1). The main findings in terms of accuracy and efficiency are reported (Section 4.2) and analysed (Section 4.3).

Section 5 contains the analysis of the stress field reproduction using a classic test case (Section 5.1). The main findings in terms of variable and geometrical accuracy are reported (Section 5.2) and analysed (Section 5.3).

In Section 6, the formulation presented in this work is applied to a topology optimisation problem (Section 6.1). The optimisation results and some data regarding accuracy and computational efficiency are presented (Section 6.2) and analysed (Section 6.3).

The main findings and conclusions of this work are laid out in Section 8.

2. Method

2.1. Level set method

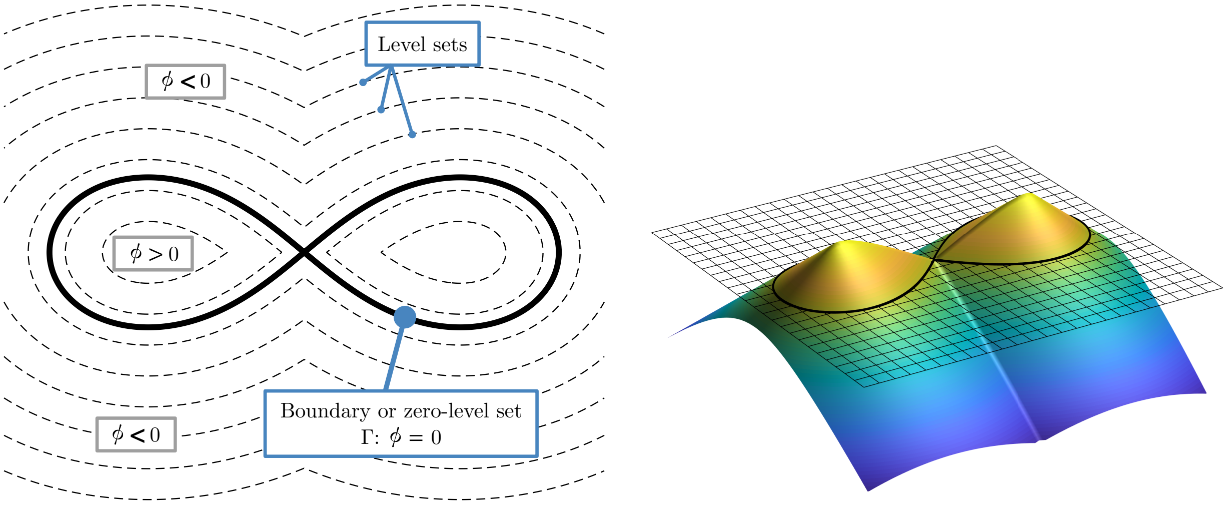
2.1.1. Introduction

The level-set method (LSM) is used to implicitly represent boundaries in space, and to enable their evolution in time when affected by some velocity field $\mathbf{v}(\mathbf{x})$. Figure 2a shows, schematically, the LSM representation of a boundary, Γ, in 2D space. Within the LSM, the boundary Γ is defined as

$$\Gamma = \{\, \mathbf{x} : \phi(\mathbf{x}) = 0,\, \mathbf{x} \in \mathrm{D} \,\}, \tag{1}$$

where D represents the entire domain 'box'. The effective domain, Ω, depends on the sign of the level-set (LS) function, ϕ, such that,

$$\begin{aligned} \phi(\mathbf{x}) > 0 &\implies \mathbf{x} \in \Omega\,, \\ \phi(\mathbf{x}) < 0 &\implies \mathbf{x} \in \mathrm{D} \setminus (\Omega \cup \Gamma)\,. \end{aligned} \tag{2}$$



(a) LSM graphic representation

(b) 3D representation of the LS signed distance field

Figure 2: The LSM representation of an arbitrary geometry through the signed distance function

In this work, we follow the common practice of adopting the signed distance function as the LS function [8, 22, 23]. Therefore, at any point, the LS function value represents the minimum distance of that point to Γ and its sign indicates whether the point is within Ω or not. In Figure 2b, the signed distance function, or the LS function, is overlaid as a 3D surface on a structured grid where Γ is implicitly represented.

The LSM uses an advection equation of the form

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0\,, \tag{3}$$

to evolve and propagate the LS function, and inherently to move boundary Γ.

If we consider that only the normal component exists in the velocity field, such that $\mathbf{v} = v_\mathrm{n}\mathbf{n}$, and given that the outward unit normal is $\mathbf{n} = -\frac{\nabla \phi}{\|\nabla \phi\|}$, one can write that

$$\frac{\partial \phi}{\partial t} - v_\mathrm{n} \,\|\nabla \phi\| = 0\,, \tag{4}$$

in which the normal-velocity field (represented by v_n) can be any function, depending on the intended movement for the boundary.

2.1.2. Reinitialization

The gradient of the LS function has the tendency to become either too flat or too steep, as time marches; thus, it is usually accepted in the literature that a reinitialization procedure is needed from time to time to maintain the quality of the LS function [8, 23]. The most common procedure reported in the literature involves the solution of a modified advection equation [8, 23], such as

$$\frac{\partial \phi}{\partial t} + S(\phi)\,\|\nabla \phi\| = S(\phi)\,, \tag{5}$$

where $S(\phi)$ is the sign function approximated by $S(\phi) = \frac{\phi}{\phi^2 + h^2 \|\nabla \phi\|^2}$; and h is the mesh size parameter.

Equation 5 in steady state ($\frac{\partial \phi}{\partial t} \approx 0$) leads to $\|\nabla \phi\| = 1$, indicating the recovery of the signed distance property [8, 23]. On the same note, one can define an error measure to quantify the deviation of the LS field to the signed distance function as [23]

$$\varepsilon_\phi = \sqrt{\frac{1}{n_p} \sum_{i=1}^{n_p} \left(\|\nabla \phi\| - 1\right)^2}\,. \tag{6}$$

where n_p represents the number of nodes.

During reinitialization, we want the boundary to remain in the same position while the LS field gets updated. In the literature, this is often achieved using a penalty constraint [23].

2.1.3. Velocity extension

There are problems in which the velocity field is defined at the boundary [8, 22]. However, the LS field is defined over the entire domain box, D. Consequently, the LSM velocity field for the boundary evolution must be defined in D. There are, in the literature, several methodologies to achieve the extension of the velocity field from the boundary to the entire domain [22]. In this work, the methodology chosen involves the solution of a modified advection equation [15], given by

$$\frac{\partial v_\mathrm{n}}{\partial t} + S(\phi) \frac{\nabla \phi}{\|\nabla \phi\|} \nabla v_\mathrm{n} = 0\,, \tag{7}$$

which extends the velocity field v_n to the entire domain box D.

The starting point of Equation 7 is a velocity field containing only the boundary velocity. In the literature, this is often achieved by using an approximation of the Dirac delta function, such as $\delta \approx \frac{1}{2}\|S(\phi)\|$ [8], which obtains an approximation of the velocity at the boundary from a set of points.

2.1.4. FEM discretization

LSM is normally implemented using finite differences [8, 15, 22]. However, the method we present in this work is based on FEM and is intended to be used within generic FE packages through user-element functionality. Consequently, the discretization of Equation 4 follows closely the one presented in [23] where the streamline diffusion method is used to stabilize the FEM solution and prevent spurious oscillations in the advection field. Therefore, consider the discretization of ϕ and a test function, ψ, as follows:

$$\begin{aligned} \phi &= \mathbf{N}_{\phi}\boldsymbol{\phi}\,, \\ \psi &= \mathbf{N}_{\phi}\boldsymbol{\psi}\,, \end{aligned} \tag{8}$$

where $\mathbf{N}_{\phi}$ is the vector of FEM linear shape functions; $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are DoF vectors for the LS and test functions, respectively.

The weak form of Equation 4, when applying the streamlined diffusion method, can be written as

$$\int_{\mathrm{D}} \left(\phi^{i+1} - \phi^{i}\right) \psi \,\mathrm{d}\Omega + \Delta t \int_{\mathrm{D}} \left(\mathbf{v}\cdot\nabla\phi\right)\left(\psi + \beta\mathbf{v}\cdot\nabla\psi\right)\,\mathrm{d}\Omega = 0\,, \tag{9}$$

which after some manipulation leads to the following FEM equations:

$$\begin{gathered} \left(\int_{\xi}\int_{\eta} \mathbf{N}_{\phi}^{\mathrm{T}}\mathbf{N}_{\phi}\,\mathrm{d}\xi\,\mathrm{d}\eta\right)\boldsymbol{\phi}^{i+1} = \int_{\xi}\int_{\eta}\left(\mathbf{N}_{\phi}^{\mathrm{T}}\phi^{i} + \Delta t\mathbf{W}^{\mathrm{T}}v_{\mathrm{n}}\left\|\nabla\phi^{i}\right\|\right)\mathrm{d}\xi\,\mathrm{d}\eta\,, \\ \mathbf{K}_{\phi}\boldsymbol{\phi}^{i+1} = \mathbf{f}_{\phi}, \end{gathered} \tag{10}$$

where $\mathbf{W} = \mathbf{N}_{\phi} + \beta\mathbf{v}\nabla\mathbf{N}_{\phi}$, and β is defined as [23]

$$\beta = \frac{1}{2\sqrt{(\Delta t)^{-2} + \left\|\mathbf{J}^{-1}\mathbf{v}\right\|^{2}}}\,. \tag{11}$$

In Equation 10, the scalar value of the velocity at the integration points is given by $v_{\mathrm{n}} = \mathbf{N}_{\phi}\mathbf{v}_{\mathrm{n}}$.

In the reinitialization procedure, the weak form of Equation 5 is:

$$\int_{\mathrm{D}} \left(\tilde{\phi}^{i+1} - \tilde{\phi}^{i}\right)\psi\,\mathrm{d}\Omega + c\,h^{2}\int_{\mathrm{D}}\left(\nabla\tilde{\phi}^{i+1}\nabla\psi\right)\,\mathrm{d}\Omega = -\Delta t\int_{\mathrm{D}}\left[\tilde{\mathbf{v}}\cdot\nabla\tilde{\phi}^{i}\left(\psi + \tilde{\beta}\tilde{\mathbf{v}}\cdot\nabla\psi\right)\right]\left[S\left(\psi + \tilde{\beta}\tilde{\mathbf{v}}\cdot\nabla\psi\right)\right]\,\mathrm{d}\Omega\,, \tag{12}$$

where the $\tilde{\cdot}$ indicates a reinitialization quantity; the reinitialization velocity is $\tilde{\mathbf{v}} = S(\phi)\mathbf{n}$.

The FEM equations for the reinitialization procedure are

$$\begin{gathered} \left[\int_{\xi}\int_{\eta}\left(\mathbf{N}_{\phi}^{\mathrm{T}}\mathbf{N}_{\phi} + c\,h^{2}\nabla\mathbf{N}_{\phi}^{\mathrm{T}}\nabla\mathbf{N}_{\phi}\right)\,\mathrm{d}\xi\,\mathrm{d}\eta\right]\tilde{\boldsymbol{\phi}}^{i+1} = \int_{\xi}\int_{\eta}\left[\mathbf{N}_{\phi}^{\mathrm{T}}\mathbf{N}_{\phi}\tilde{\boldsymbol{\phi}}^{i} + \Delta t\tilde{\mathbf{W}}^{\mathrm{T}}\left(S - \tilde{\mathbf{v}}^{\mathrm{T}}\nabla\tilde{\phi}^{i}\right)\right]\,\mathrm{d}\xi\,\mathrm{d}\eta\,, \\ \tilde{\mathbf{K}}_{\phi}\tilde{\boldsymbol{\phi}}^{i+1} = \tilde{\mathbf{f}}_{\phi}\,, \end{gathered} \tag{13}$$

where c is a parameter controlling the effect of the diffusion term; $\tilde{\mathbf{W}}^{\mathrm{T}} = \mathbf{N}_{\phi} + \tilde{\beta}\tilde{\mathbf{v}}\nabla\mathbf{N}_{\phi}$; and, $\tilde{\beta}$ is defined as

$$\tilde{\beta} = \frac{1}{2\sqrt{(\Delta t)^{-2} + \left\|\mathbf{J}^{-1}\tilde{\mathbf{v}}\right\|^{2}}}\,. \tag{14}$$

The FEM equations for the reinitialization procedure are only complete with a penalty constraint

$$\left(\tilde{\mathbf{K}}_\phi + \omega \times \mathbf{G}^{\mathrm{T}}\mathbf{G}\right)\tilde{\boldsymbol{\phi}}^{i+1} = \tilde{\mathbf{f}}_\phi\,, \tag{15}$$

where ω is a constant that represents the weight of the constraint; and $\mathbf{G}$ is a penalty matrix defined as:

$$\mathbf{G}_{ij} = \mathbf{N}_\phi(\xi_i^*, \eta_i^*)_j\,, \quad j = 1, \ldots, n^{node}\,, \tag{16}$$

where i represents the i^{th} intersection between the boundary and an element; ξ_i^* and η_i^* constitute the pair of coordinates of the i^{th} intersection point in the natural coordinate system.

Equation 7 for the velocity extension is discretized following a similar procedure to the LS equation, yielding

$$\begin{gathered}\left(\int_\xi\int_\eta \mathbf{N}_\phi^{\mathrm{T}}\mathbf{N}_\phi\,\mathrm{d}\xi\,\mathrm{d}\eta\right)\mathbf{v}_{\mathrm{n}}^{i+1} = \int_\xi\int_\eta\left(\mathbf{N}_\phi^{\mathrm{T}}\mathbf{N}_\phi\mathbf{v}_{\mathrm{n}}^{i} + \Delta t\mathbf{W}^{\mathrm{T}}S\frac{\nabla\phi}{\|\nabla\phi\|}\nabla\mathbf{v}_{\mathrm{n}}\right)\mathrm{d}\xi\,\mathrm{d}\eta\,,\\ \mathbf{K}_{\mathbf{v}}\mathbf{v}_{\mathrm{n}}^{i+1} = \mathbf{f}_{\mathbf{v}}\,.\end{gathered} \tag{17}$$

2.2. FNM for explicit boundary discretization

2.2.1. Proposed element

There are physical problems in which some quantities are exclusively defined at the boundary [1, 8]. Section 2.1 outlines the LSM which is used to implicitly capture the geometry of the moving boundary, and therefore not providing neither direct access to boundary information or an accurate geometric representation in the numerical model (e.g. for a mechanical analysis). In order to capture the zero-level set in the numerical model accurately, we propose for the first time to use FNM [6] to partition elements in the original mesh.

In the elements intersecting the moving boundary, there are real nodes with positive LS values and others with negative LS values. An edge in which the nodes have different LS signs has been intersected by Γ. In this work, we consider that the element size is small in comparison to the curvature of the LS field, which means that only one intersection will occur per edge. This assumption allows us to build the proposed element (Figure 3) with one floating node per edge; additionally, the assumption leads to three intersection scenarios (Figure 4) when considering a linear approximation of Γ.

From the LS values at each real node, the position of the zero-level set within the edge is given by linear interpolation. This position is then assigned to the corresponding floating node on that edge. Essentially, a floating node is activated whenever an intersection between an edge and the boundary occurs. But, since the existence of these floating nodes is enabled by design of the FNM, the activation of a floating node requires no changes to the elements connectivity, thus the initial input file of the analysis remains unchanged.

The floating nodes, representing the intersections, are topologically associated with a boundary. Thus,

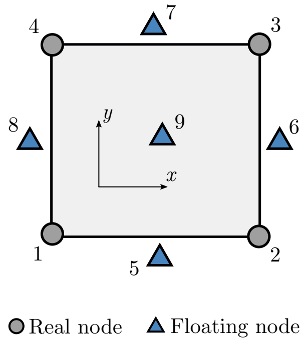


Figure 3: The finite element adopted for the FNM implementation with pre-allocated floating nodes

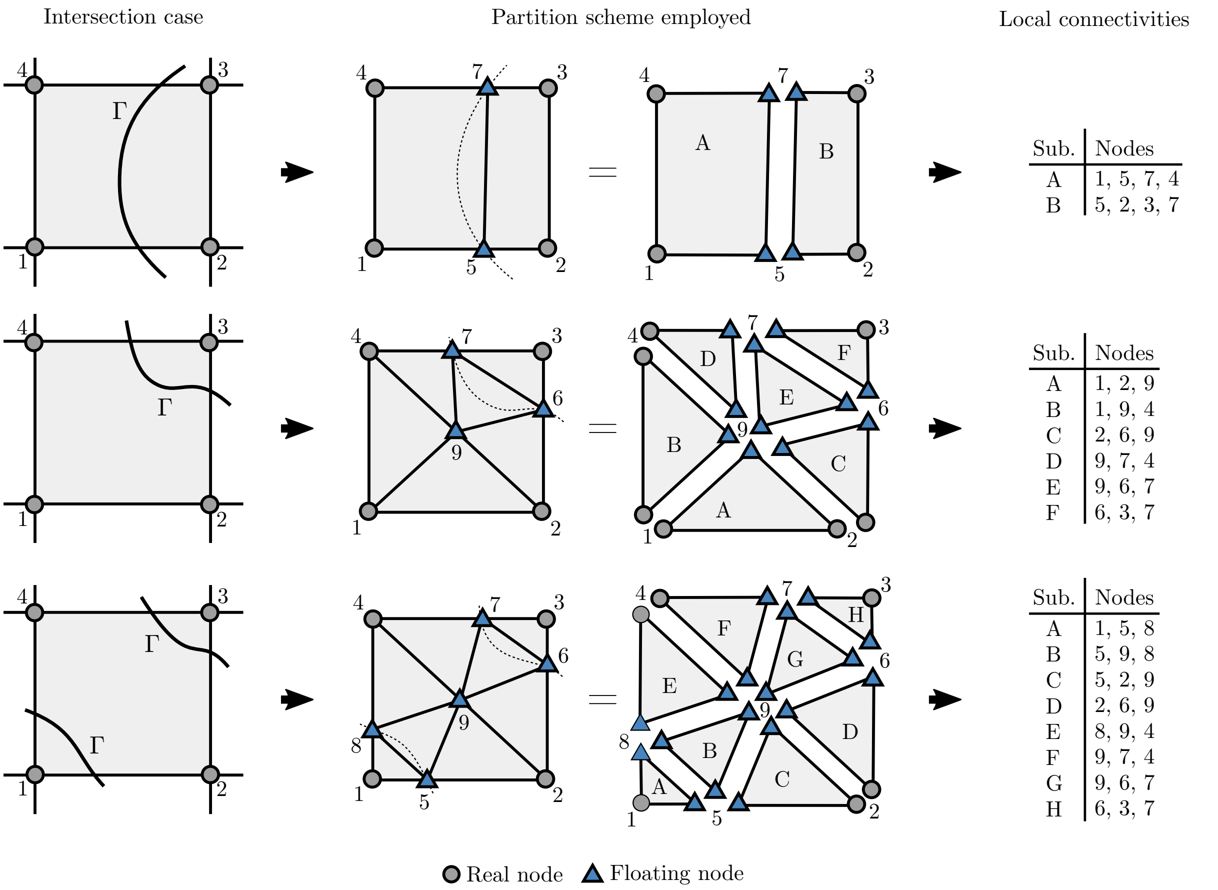
during the numerical analysis steps, they can provide information directly from the boundary. In order to include these nodes in the numerical model (e.g. for a mechanical analysis), the original element is partitioned into sub-elements. The floating nodes assigned to the area of the element (represented in Figure 3 at the centre of the proposed element) are needed to facilitate the element partitioning, and increase the overall sub-element quality.

After the partitioning, each sub-element is treated as its own element contributing to the stiffness matrix of the parent element (during assembly). As an example, focusing on the first case in Figure 4 (the intersection of opposing edges) outlined in detail in Figure 5, the partitioning leads to the creation of two quadrilateral sub-elements with coordinates $\mathbf{x}_A^{\mathrm{T}} = \left[\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_5^{\mathrm{T}}, \mathbf{x}_7^{\mathrm{T}}, \mathbf{x}_4^{\mathrm{T}}\right]$ and $\mathbf{x}_B^{\mathrm{T}} = \left[\mathbf{x}_5^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}}, \mathbf{x}_3^{\mathrm{T}}, \mathbf{x}_7^{\mathrm{T}}\right]$. The more complex partitions use triangular sub-elements, though the methodology is the same. Figure 5 outlines the relationship between the global connectivity matrix and the local partition scheme. It is possible to observe that the global connectivity matrix already accounts for all the parition schemes defined in Figure 4. This way, creating a local partition is just a matter of selectively activating the right combination of real and floating nodes without the need to change the initial input file and the global elemental connectivities. The local sub-element connectivities are, therefore, needed when calculating the local stiffness matrices, assembling the global stiffness matrix and for post-processing.

Having the boundary as part of the numerical model allows us to more efficiently select the active DoF for analysis at each iteration. Essentially, the mesh of domain Ω is now a subset of the mesh of D, thus reducing the overall number of DoF at each step.

2.2.2. Proposed reinitialization scheme

With the newly-gained direct access to boundary nodes, it is possible to implement a new reinitialization procedure (Figure 6). This will be outlined herein for the case of the signed distance function. Since the boundary is now populated with floating nodes, one way to approximate the minimum distance to the boundary is to evaluate the distance to the closest floating node. Despite this approach being an



Sub.	Nodes
A	1, 5, 7, 4
B	5, 2, 3, 7

Sub.	Nodes
A	1, 2, 9
B	1, 9, 4
C	2, 6, 9
D	9, 7, 4
E	9, 6, 7
F	6, 3, 7

Sub.	Nodes
A	1, 5, 8
B	5, 9, 8
C	5, 2, 9
D	2, 6, 9
E	8, 9, 4
F	9, 7, 4
G	9, 6, 7
H	6, 3, 7

Figure 4: The three intersection cases considered in this work, and the element partitioning schemes chosen with the corresponding sub-element connectivities

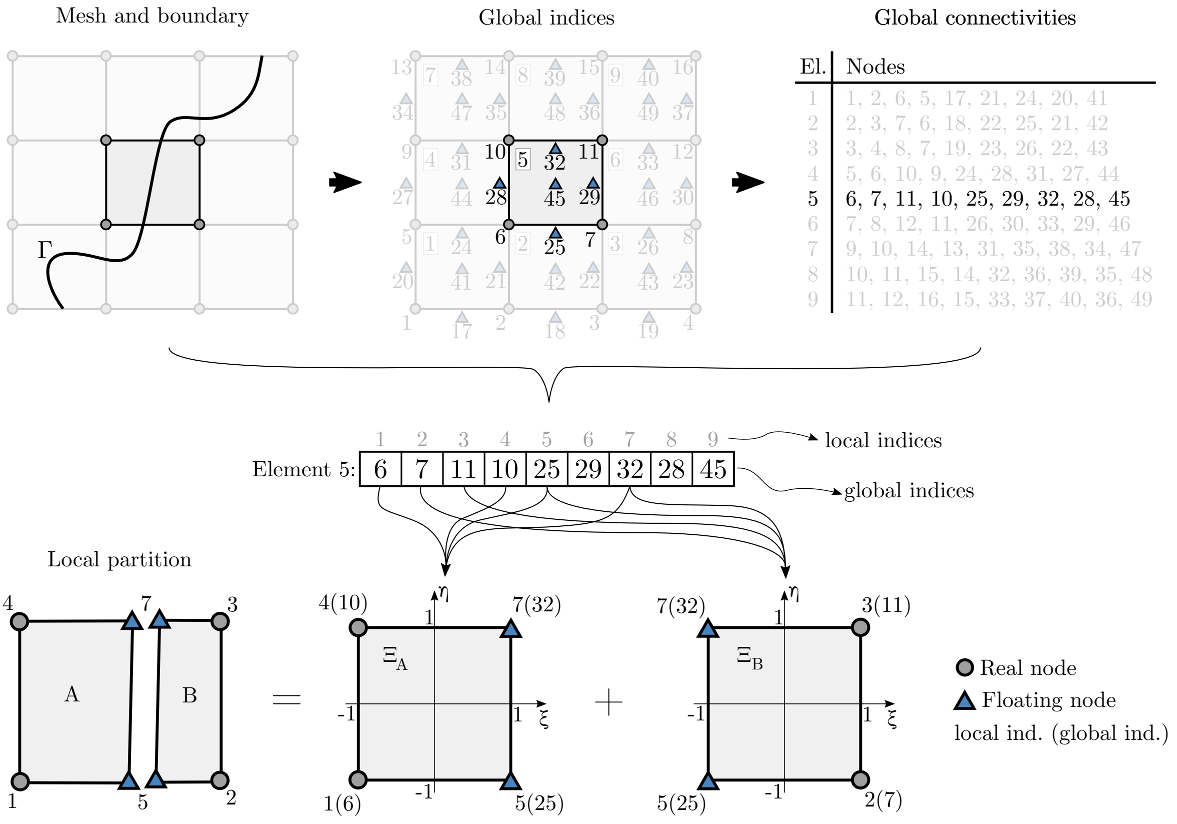
approximation, it is, conceptually, a simple and efficient way to obtain a near signed-distance property throughout the entire process.

With this new reinitialization scheme, we abandon the need to iterate through the modified advection equation for the reinitialization (Equation 13). Conversely, this new scheme allows us to achieve the best quality possible of the LS field, in one single step.

2.2.3. Modified velocity extension

In addition to the increased geometrical accuracy, the floating nodes on the boundary provide increased physical accuracy since one can, for example, probe displacements or stresses directly at the boundary.

As it is referred in Section 2.1.3, there are scenarios in which the velocity field is defined originally at the boundary, but needs to be extended to the entirety of D. Traditionally, the extension would be achieved by capturing a numerical approximation of the boundary velocity (e.g. through approximated Delta dirac functions) and then extending it by solving the appropriate advection equation. Alternatively, since in the



El.	Nodes
1	1, 2, 6, 5, 17, 21, 24, 20, 41
2	2, 3, 7, 6, 18, 22, 25, 21, 42
3	3, 4, 8, 7, 19, 23, 26, 22, 43
4	5, 6, 10, 9, 24, 28, 31, 27, 44
5	6, 7, 11, 10, 25, 29, 32, 28, 45
6	7, 8, 12, 11, 26, 30, 33, 29, 46
7	9, 10, 14, 13, 31, 35, 38, 34, 47
8	10, 11, 15, 14, 32, 36, 39, 35, 48
9	11, 12, 16, 15, 33, 37, 40, 36, 49

1	2	3	4	5	6	7	8	9
6	7	11	10	25	29	32	28	45

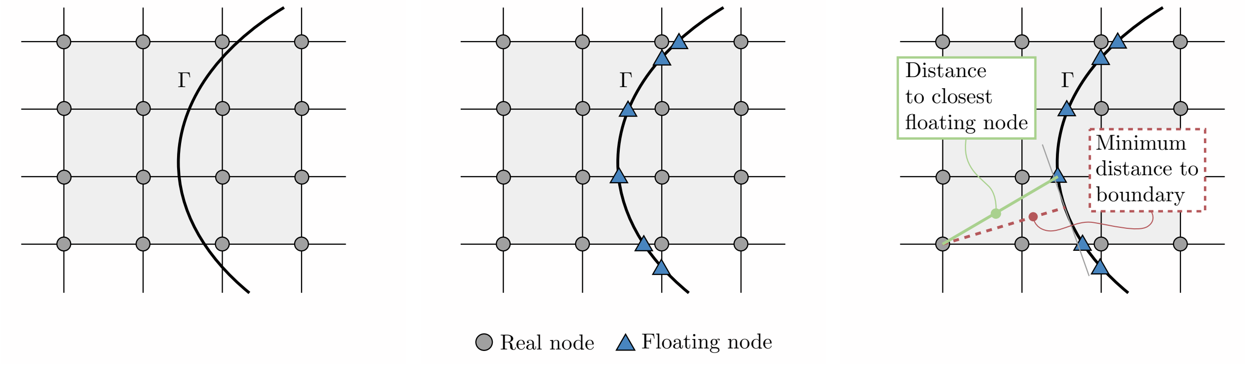
Figure 5: Schematic outlining the process to obtain local element partitions from an unchanging global connectivity matrix

proposed methodology the velocity can be calculated directly at the floating nodes on the boundary, the extension can be modified to benefit from the more accurate data of the floating nodes.

A more in-depth analysis of Equation 7 allows us to observe that in steady state ($\frac{\partial v_\mathrm{n}}{\partial t} \approx 0$) we obtain the following:

$$\nabla\phi \cdot \nabla v_\mathrm{n} = 0\,, \tag{18}$$

which indicates that the normal of the LS field and the normal of the velocity field are perpendicular to each other. Thus, it is reasonable to consider that, in the close vicinity of the boundary, the velocity field is constant in the normal direction to the boundary [15]. Following this insight, we calculate the projected velocities of each node on the boundary, and thus obtain an approximation of the velocity field for the real nodes closest to the boundary. This methodology is graphically represented in Figure 7, for the case in which a node is shared by two elements intersected by the boundary. In this case, the final approximated velocity is the average of the individual projections.



(a) Mesh superposed on Γ

(b) Discretization of Γ

(c) Approximation of the LS value

Figure 6: Novel LS reinitialization procedure using the proposed methodology

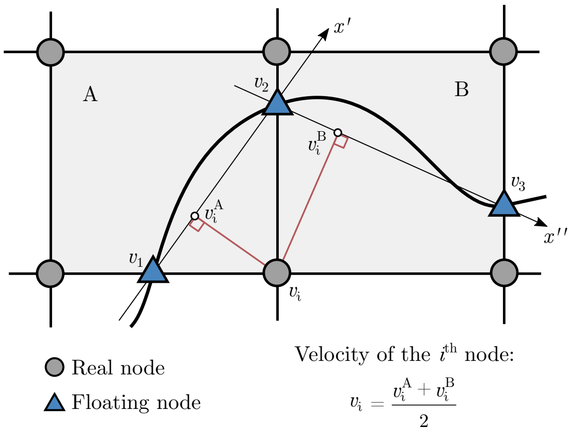


Figure 7: Interpolation of the boundary velocities (v) to the neighbouring real nodes in preparation for the step of velocity extension

2.3. Baseline and proposed methods

For the purpose of benchmarking, we implement in this work a baseline method alongside the proposed method. The baseline method aims at being representative of the state-of-the-art on implicit representation of evolving boundaries using FEM and LSM. Thus, this work will show how the proposed explicit-discretization method built upon an implicit LSM model compares, in performance, to a fully implicit methodology using an ersatz material approach. Figure 8 contains the flowchart representation of both methods, side-by-side.

The formulations described here were implemented in a *Matlab* code (available at this link). Furthermore, the results presented and discussed in this work (available at this link) were obtained with a standard desktop computer, and all timed runs were obtained under equivalent conditions of computer use.

The differences between the baseline and proposed methodologies are observable when comparing Figures 8a and 8b. Namely, in the proposed method, the reinitialization procedure is achieved in a single step, while the baseline method requires a loop to solve the FEM reinitialization equations, and a penalty constraint. Note that n, in Figure 8a, is the number of times the loop is meant to repeated; the choice of n impacts greatly the runtime and accuracy and no common practice can be found in literature (alternatively a convergence criterion could be devised, although the accuracy improvements do not appear to justify the much greater increase in runtime). Additionaly, the proposed method introduces mesh partitioning steps and the need to update the set of active DoF at every iteration, which does not happen in the baseline. The baseline method employs the ersatz material approach in which two phases (void and solid) are defined by a pseudo-density that multiplies with the material properties. This pseudo-density is based on the average of the nodal LS values: if negative, the pseudo-density is set to 0.001; if positive or zero the pseudo-density is set to 1.

Highlighted in Figures 8a and 8b are the two boxes referring to the computation of the velocity field for the LSM. This is because the velocity calculation is problem-specific. In this work, we choose to apply the proposed methodology within topology optimisation. In the topology optimisation problem, the LS velocity field corresponds to the sensitivities calculated in the optimisation problem. Therefore, the nodal velocities (Appendix A) are given, at node j, by

$$\begin{aligned} \mathbf{v}_\mathrm{n}(j) &= \frac{1}{n^e}\sum_{i=1}^{n^e}\frac{v^e_{\mathrm{n}i}}{A^e_i} - \bar{\lambda} \quad (\text{implicit})\,, \\ \mathbf{v}_\mathrm{n}(j) &= \boldsymbol{\epsilon}^\mathrm{T}(j)\boldsymbol{\sigma}(j) - \bar{\lambda} \quad (\text{explicit})\,. \end{aligned} \tag{19}$$

where the superscript e indicates an elemental quantity; n^e is the number of elements intersecting the boundary to which the node j belongs; A^e_i is the area of an element; $\bar{\lambda}$ is a modified Lagrange multiplier; and, v^e_n is the elemental velocity given by $v^e_\mathrm{n} = \mathbf{q}_u^\mathrm{T}\mathbf{K}^e_u\mathbf{q}_u$ (see Appendix A); $\mathbf{q}_u$ is the vector of displacement DoF; and $\mathbf{K}_u$ is the stiffness matrix of the elasticity problem. The displacements are computed through the FEM equations (Appendix A) as follows:

$$\mathbf{K}_u\mathbf{q}_u = \mathbf{f}_u\,, \tag{20}$$

in which

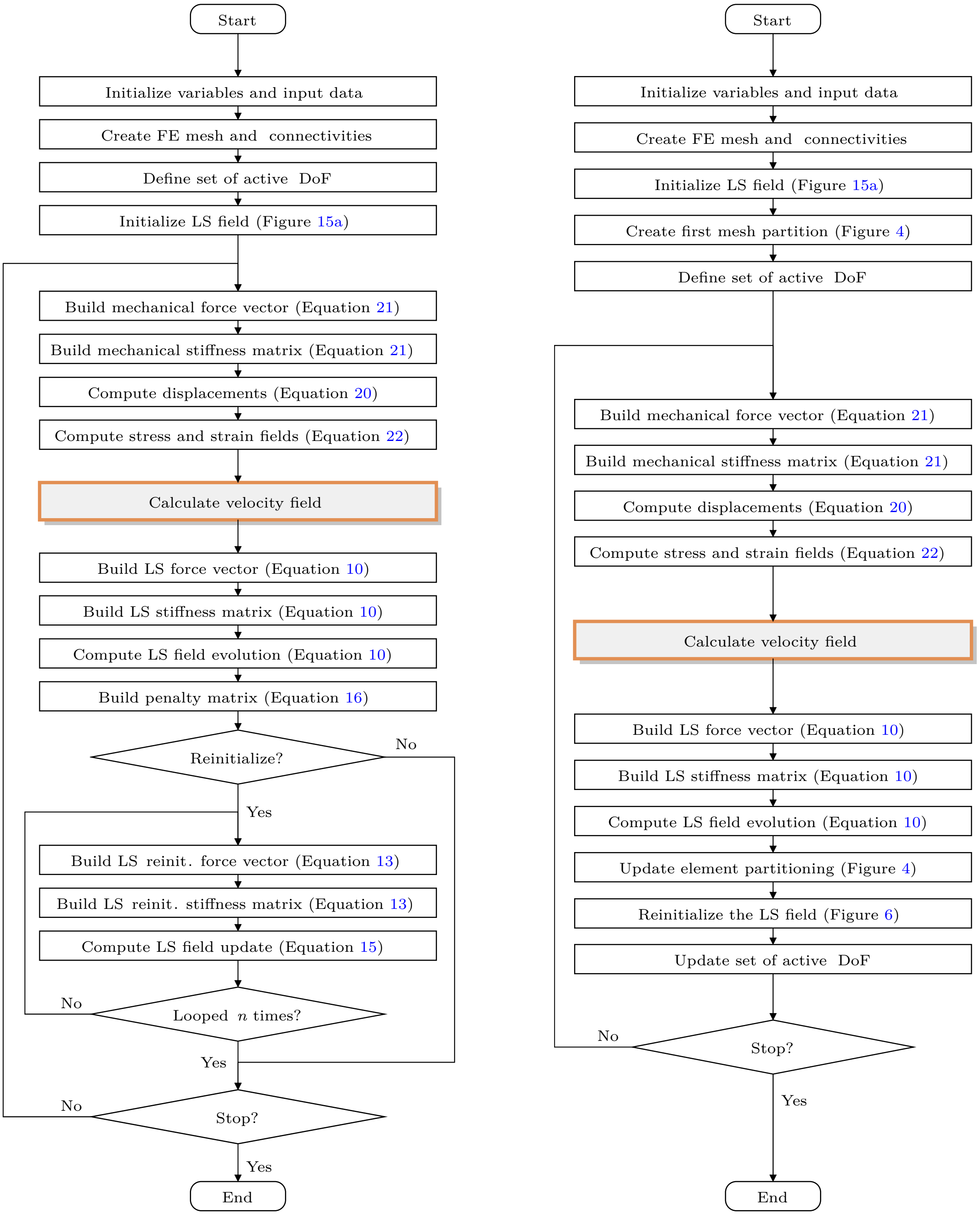
$$\begin{aligned} \mathbf{K}_u &= \rho\int_\xi\int_\eta \mathbf{B}^\mathrm{T}\mathbf{D}\mathbf{B}\det(\mathbf{J})\,\mathrm{d}\xi\,\mathrm{d}\eta\,, \\ \mathbf{f}_u &= \mathbf{f}\,, \end{aligned} \tag{21}$$

where $\mathbf{B} = \mathcal{L}_\xi\mathbf{J}^{-1}$ and $\mathbf{J}$ the Jacobian matrix; $\mathbf{D}$ is the elastic plane-stress constitutive matrix, and ρ is the pseudo-density of the ersatz material in the baseline methodology; $\mathbf{f}_u$ is the force vector; and $\mathbf{f}$ corresponds to the vector of applied loads. Strain and stress tensors are computed as

$$\begin{aligned}\boldsymbol{\epsilon} &= \mathbf{B}\mathbf{q}_u \,, \\ \boldsymbol{\sigma} &= \mathbf{D}\boldsymbol{\epsilon} \,.\end{aligned} \tag{22}$$

It is important to note that, the proposed method has, theoretically, no barriers on the extension to other FEM formulations dealing with different physics or on the extension to a three-dimensional variant. For the implementation in three-dimensions the main areas requiring extension are:

- the partitioning cases - these would need to be defined in order to account for surface and face intersections and the 3D-elements topologies;
- the local sub-element connectivities - these would need to be modified to account for different element geometries and the new partitioning cases;
- the partitioning algorithm - the concept of comparing nodal values to check edge intersection would need to be extended to check for surface intersection; and
- the velocity field extension - the proposed scheme relies on a 2D interpretation of the equations and would need to be expanded to account for the third dimension.


Start
Initialize variables and input data
Create FE mesh and connectivities
Define set of active DoF
Initialize LS field (Figure 15a)
Build mechanical force vector (Equation 21)
Build mechanical stiffness matrix (Equation 21)
Compute displacements (Equation 20)
Compute stress and strain fields (Equation 22)
Calculate velocity field
Build LS force vector (Equation 10)
Build LS stiffness matrix (Equation 10)
Compute LS field evolution (Equation 10)
Build penalty matrix (Equation 16)
Reinitialize?
No
Yes
Build LS reinit. force vector (Equation 13)
Build LS reinit. stiffness matrix (Equation 13)
Compute LS field update (Equation 15)
No
Looped n times?
Yes
No
Stop?
Yes
End
Start
Initialize variables and input data
Create FE mesh and connectivities
Initialize LS field (Figure 15a)
Create first mesh partition (Figure 4)
Define set of active DoF
Build mechanical force vector (Equation 21)
Build mechanical stiffness matrix (Equation 21)
Compute displacements (Equation 20)
Compute stress and strain fields (Equation 22)
Calculate velocity field
Build LS force vector (Equation 10)
Build LS stiffness matrix (Equation 10)
Compute LS field evolution (Equation 10)
Update element partitioning (Figure 4)
Reinitialize the LS field (Figure 6)
Update set of active DoF
No
Stop?
Yes
End

(a) Baseline method (implicit)

(b) Proposed method (explicit)

Figure 8: Flowchart representation of both methodologies implemented

3. Comparison with different discretization methodologies

From Section 2 it is possible to conclude that the proposed methodology is based on the implicit LSM and discretizes the boundary explicitly in the mechanical model after every topology change. The quantitative analysis of the proposed methodology, its features and applications will follow in the next sections; in this section, we will focus on its qualitative analysis and comparison to other analogous discretization methodologies.

Numerical methodologies such as the one we propose and others alike in literature can be assessed and compared on a set of characteristics or criteria. These are:

- accuracy - not in general terms but regarding information at the boundary. Methodologies which are able to provide information at the boundary without interpolation/extrapolation or other numerical approximations, apart from the ones inherent to FEM, will generally lead to more accurate information at the boundary;
- computational load - measures how the methodology-specific algorithms affect overall performance (e.g. requiring more memory, accessing memory more often, performing more calculations, and so on); and
- implementation difficulty - a measure of both theoretical and practical difficulty of implementation. More theoretically-involved methods often lead to a steeper learning curve and harder implementation. Also, measures the ability to implement the method on more closed, generic FE packages.

For this comparison exercise let us assume that all methodologies are implemented on FEM, and that apart from the methodology-specific algorithms all of the methodologies are on equal footing regarding the comparison criteria above (e.g. they use the same type of element and integration scheme so that any accuracy difference is due to the specific methodology; they use the same general algorithmic and solution approach so that any computational load difference is due to the specific methodology).

Comparison with remeshing and partial-remeshing

In terms of accuracy of solution fields at the boundary, remeshing has, theoretically, the potential to be the most accurate methodology. Remeshing always retains the boundary explicitly in the numerical model [18, 26]; furthermore, assuming that for a particular topology the mesh obtained through remeshing is the best posssible mesh for that shape, then the solution accuracy should follow as being the best possible for the particular domain and mesh characteristics (density, element type, and so on).

The main differences of using the proposed methodology instead of remeshing are:

- conceptually FNM can be thought of as a methodology realising local remeshing at the element level. However, computationally, FNM is different to remeshing in that the original mesh remains unchanged, as do the elemental connectivities; in remeshing the the problem is 'reset' after every evolution as if an entirely new and disconnected domain is being analysed at each time step [18, 26];

- in the proposed method the partition operations are localised to each element independently which provides a framework for parallelisation of the algorithm; in remeshing the process if often global and not readily localisable [18, 26]; and

- in the proposed methodology the element geometry and quality is dependent on the initial mesh and the current topology which can lead to distorted sub-elements; remeshing gives complete control over the mesh generation allowing, in theory, to obtain the best quality mesh possible for the given topology [18, 26].

Apart from the extent of the mesh modification, partial remeshing is equal to remeshing in every way. In fact, some partial remeshing techniques rely on operations over the entire mesh to maintain the mesh quality [11]. Thus, the points made for remeshing are also valid in the case of partial remeshing. Additionally, partial remeshing is closer to the proposed methodology in the sense that a compromise between quality and computational cost is made in relation to remeshing.

When it comes a remeshing-based approach, the proposed methodology is expected to provide the following advantages:

- better computational efficiency, both in terms of wall time and in terms of use of resources; and

- ability to parallelise algorithms and to implement the methodology inside a generic FE-package through user-element functionality;

The expected disadvantages in regards to remeshing are:

- some compromise in accuracy due to possibly distorted sub-elements;

Comparison with XFEM approaches

XFEM is a methodology that uses enrichment functions at the element level in order to account for discontinuities in the solution field [5, 36]. A XFEM approach of particular interest in the scope of this paper is the phantom node method (PNM) [37]. PNM has been shown to be equivalent to XFEM with Heaviside enrichments when it comes to displacement approximation [6].

When comparing to general XFEM approaches, the main differences to the proposed methodology are:

- The proposed methodology introduces floating nodes at the boundary to partition the element into sub-elements therefore creating an explicit boundary in the model. While with XFEM the number of nodes is kept constant and the boundary is implicitly captured within the element through the use of modified shape functions [5, 36];

- in the proposed methodology only displacement DoF are considered, whilst with XFEM assigns extra DoF at each node of an enriched element [5, 36];

- in the proposed methodology each sub-element, in a partitioned element, is transformed into natural coordinates separately and its unique-full domain is used for integration; in XFEM each enriched element is integrated multiple times for partial domains [5, 36]; and

- numerically, the proposed methodology leads to standard FEM equations for the given topology, and apart from uncertainties due to element distortion the solution of both should match; XFEM approaches are known to introduce an error representing discontinuities in natural coordinates [6].

In addition to the above, the main difference between the proposed methodology based on FNM and one based on PNM lies in the fact that in the proposed methodology the floating nodes rest at the boundary. While in a PNM-based approach, the phantom nodes rest atop the real element nodes, thus no explicit boundary discretization is attained [6, 37].

When it comes a XFEM-based approach, the proposed methodology is expected to provide the following advantages:

- better accuracy of information at the boundary due to the floating nodes allowing for an explicitly discretized numerical model; and

- easier to understand concept and implementation, both theoretically and practically;

4. Reinitialization procedure

4.1. Test Case

A simple case study was devised to test the efficiency and accuracy of the proposed reinitialization procedure; it consists of a simple square with a hole in the centre (Figure 9). A uniform unit velocity is imposed over the entire domain, such that the hole increases in diameter over time. The test consists of modelling the evolution of the boundary, by considering both methodologies (baseline and proposed) and different reinitialization strategies: (i) reinitializing after every iteration; (ii) reinitializing after every five iterations; and (iii) reinitializing after every ten iterations. The parameters chosen are $L = 1\,\text{mm}$, $D = 0.2\,\text{mm}$, $\|v\| = 1\,\text{mms}^{-1}$, $\alpha = 0.5$, $c = 0.1$, $\omega = 10^5$, and a mesh of 150×150 isoparametric quadrilateral and triangular elements with standard Gauss integration (4 integration points for the quadrilaterals, 3 integration points for the triangles).

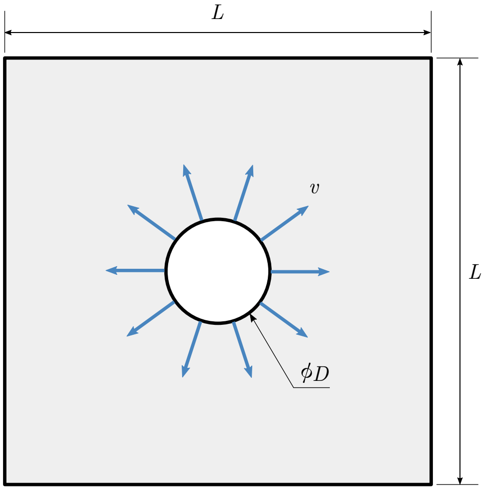
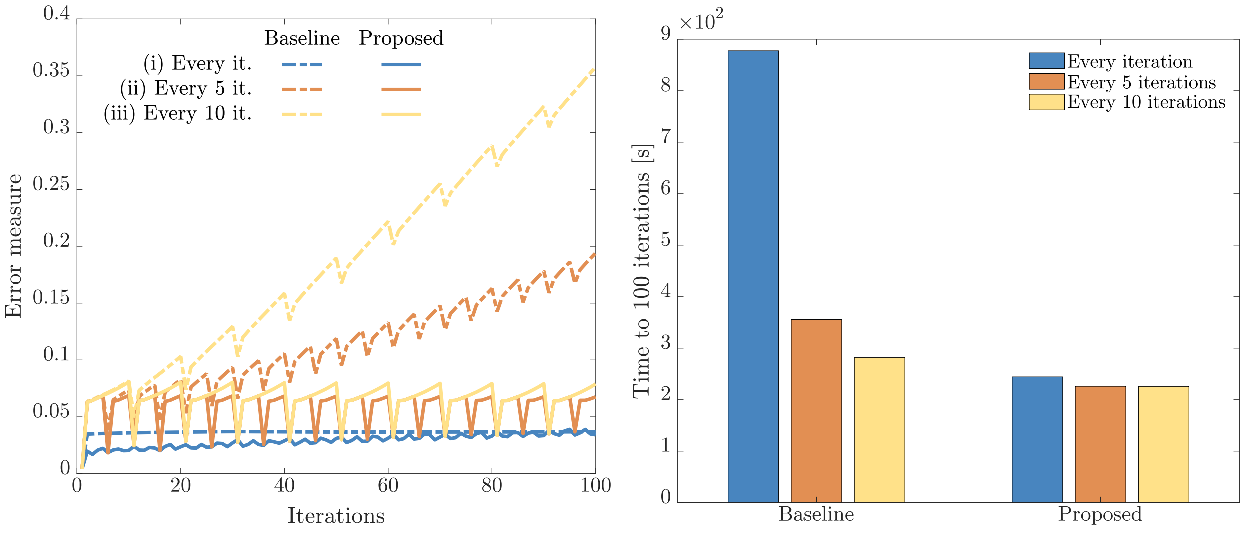


Figure 9: Schematic of the reinitialization case test

4.2. Results

Figure 10a shows the evolution of the error measure (Equation 6) over time for each strategy and method.



(a) Quality of the LS field over time

(b) Time taken for the reinitialization test

Figure 10: Effect of how often the LS field is reinitialised on the accuracy and runtime of the standard and proposed methods.

The baseline methodology, with strategy (i), yields good accuracy over the iteration range without any noticeable variability. The same methodology, with strategies (ii) and (iii), shows several 'drops' that get smaller as the iterations increase, and also leads to a significant and increasing deviation from the signed distance function. In comparison, the proposed methodology, with strategy (i), produced the best accuracy, though the curves show more variability in the error; furthermore, the accuracy of the proposed methodology, with strategy (i), shows an increasing trend surpassing the corresponding baseline results at the 100th iteration. The proposed methodology, using strategies (ii) and (iii), yields oscillatory error levels with no error escalation over time, and with the 'drops' matching closely the error level of the proposed method with strategy (i).

Figure 10b shows the time it took each method, with all three strategies, to run 100 iterations. In both methods, strategy (i) ran the slowest and (iii) the fastest. Also, it can be observed that the proposed methodology exhibits a much smaller difference between strategies.

4.3. Analysis

Figure 10a shows that several of the curves exhibit periodic 'valleys' or 'drops', which mark the iteration of the reinitialization procedure (notice that the period for strategy (iii) is double that for strategy(ii)). Hence, these 'drops' convey a gain in accuracy resulting from the reinitialization procedure. Furthermore, using the baseline methodology, with strategies (ii) and (iii), it is possible to observe a decaying impact of the reinitialization procedure, i.e., the accuracy gain decreases as iterations increase. In contrast, the reinitialization procedure in the proposed methodology always appears to yield the lowest possible error attainable by the method (that of strategy (i)) regardless of the strategy.

The increasing error of the proposed methodology, with strategy (i), and the presence of variability, conveys that the proposed reinitialization procedure is topology-dependent, i.e., its accuracy is not constant but changes with the evolving geometry. This feature stems from the fact that the method approximates the minimum distance to the boundary to the position of the closest floating node. Therefore, the level of accuracy of the method is dependent on how well this approximation holds for the geometry and mesh partition at a particular iteration. In this benchmark a single boundary is considered, which is increasing in length over the iteration range. This length increase means that the approximation in the proposed reinitialization is worse and will, eventually, lead to higher errors than those of the corresponding baseline strategy. However, in a real example (such as the ones in Section 7) there are more boundary segments each of smaller length which will, in theory, give the advantage to the proposed reinitialization scheme.

Overall, the proposed methodology performs the fastest regardless of strategy. Also, depending on the boundary topology and strategy employed the proposed methodology can yield better accuracy. Hence, the proposed methodology can be advantageous in both error (Figure 10a) and computational cost (Figure 10b) of the reinitialization procedure.

5. Stress field accuracy

5.1. Test Case

A test is conducted to verify the capability of the proposed method to accurately represent physical fields (such as stress fields) around the boundary. This is expected to provide important insights into the importance of explicitly discretizing the boundary with respect to implicit discretization.

For this, the standard calculation of the stress concentration factor (SCF) for an infinite elastic square plate with a central hole under remote stress σ^{∞} is chosen as test case (Figure 11), for which the analytical solution is SCF $= \frac{\sigma^{\max}}{\sigma^{\infty}} = 3$. The dimensions and loads considered are $L = 3\,\mathrm{mm}$, $D = 0.2\,\mathrm{mm}$, and $\sigma^{\infty} = 1\,\mathrm{N mm^{-1}}$. Various tests are performed on uniform meshes with element sizes ranging from 0.075 mm to 0.0075 mm following the series $\{\frac{0.075}{1}, \frac{0.075}{2}, \frac{0.075}{3}, \ldots, \frac{0.075}{10}\}$. We employ, in this test case, the same isoparametric elements as in section 4.1.

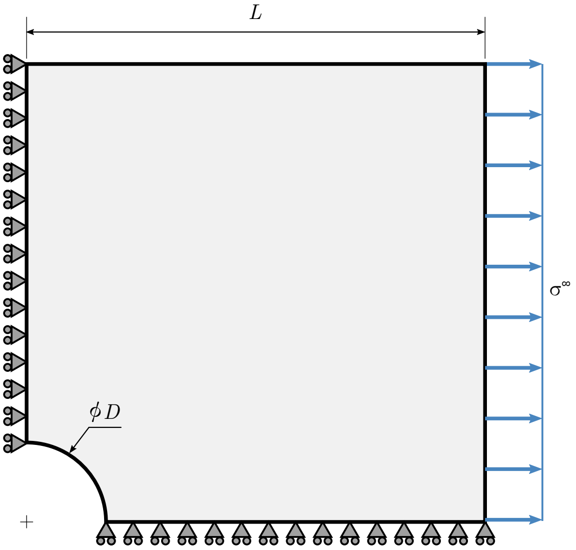


Figure 11: Schematic of the quarter plate with the central hole used as test case to study the stress field accuracy

5.2. Results

In Figure 12, the relative error in the SCF prediction is plotted for the proposed and baseline methods over the different mesh densities. It can be seen that the proposed method is consistently more accurate than the baseline method, with the exception of one point. The proposed method converges to a small error as the mesh is refined, while the baseline method shows larger error variability which does not appear to reduce as the mesh is refined.

Figure 13 shows a detail of the geometry of the hole with the respective stress field. Qualitatively, it can be observed that the two methods reproduce the geometry very differently. While the baseline method

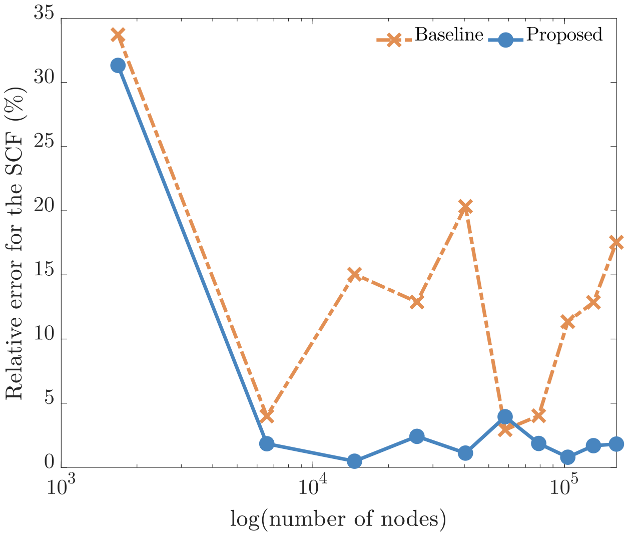


Figure 12: Relative error of the resulting stress concentration factor for different mesh densities when compared to the analytical solution

presents a 'pixelated' hole boundary, the proposed method conveys an accurate geometric representation with a smooth boundary. In terms of the stress field, the baseline method produces stress concentrations on every 'pixel' while the proposed method produces an artefact-free stress field.

5.3. Analysis

The variability in the relative error shown by the baseline method (Figure 12) is due to the lack of geometrical accuracy in the mechanical model. The domain represented in the mechanical model is, effectively, different to the domain represented in the LS field. For this reason, there will be mesh sizes for which the stress values are closer to the analytical solution and other mesh sizes for which the stress values are very different, leading to the observed variability.

In comparison, in the proposed method, some points show a slight increase in error despite the increase in mesh density. This characteristic stems from the fact that, for each mesh density, the mesh partition is different, and consequently so is the shape and quality of the sub-elements. While the proposed method may ocasionally generate a partitioning configuration leading to a lower quality sub-element, it still yields SCF predictions which are consistently more accurate than the baseline method.

Figure 13 shows that the proposed methodology can capture the geometry represented in the LS field and accurately compute the mechanical response to a load.

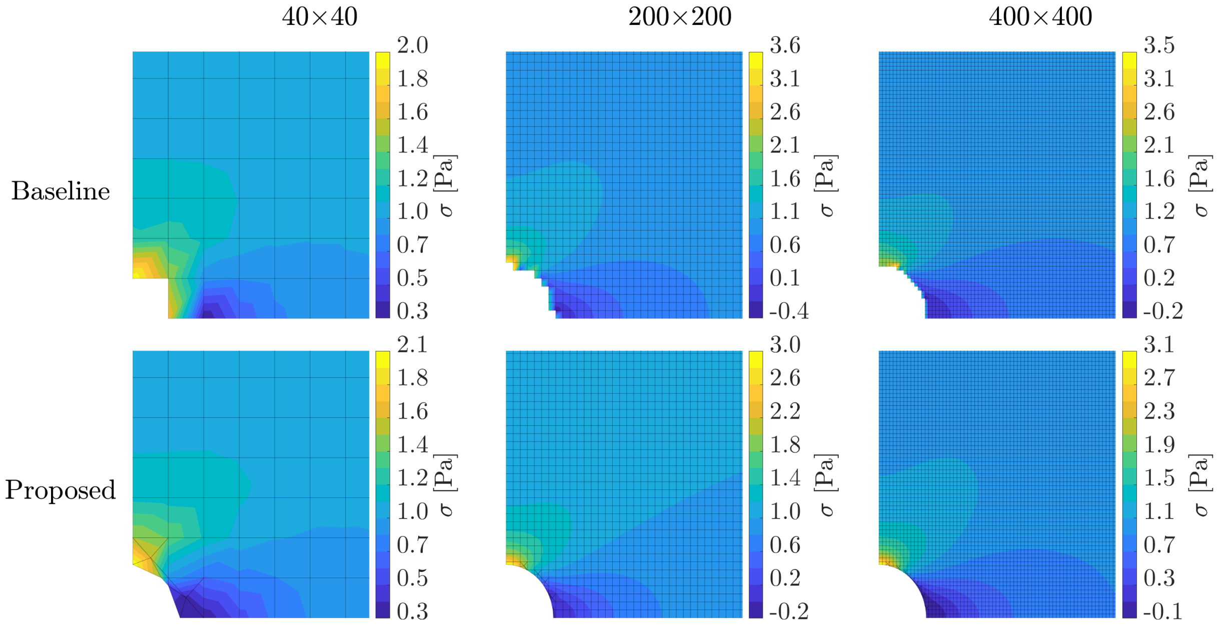


Figure 13: Details of the hole geometries and stress fields for different mesh densities

6. Benchmark: topology optimisation of a 2D cantilever beam

6.1. Problem definition

Structural topology optimization is an ideal possible field where the proposed method can be applied with expected benefits. In this work, the proposed methodology will be showcased in the solution of the classic 2D cantilever beam problem [8], in which the topology of a rectangular domain (Figure 14), $L \times H$, is optimised in order to achieve a configuration of minimum mean compliance within the limits of a volume constraint.

A unit point load is applied in the middle of the right edge pointing in the downwards direction, and the beam is clamped in the left edge. The optimisation algorithm is initiated with a grid of circular holes such that the initial volume is close to the desired final volume (Figure 15a). The parameters used are $L = 2\,\mathrm{mm}$, $H = 1\,\mathrm{mm}$, $F = 1\,\mathrm{N}$, $E = 1\,\mathrm{MPa}$, $\nu = 0.3$, $\alpha = 1$, $\tau = 0.9$, $\lambda^0 = \gamma^0 = 1$ (parameters in the optimisation process as detailed in Appendix A), $c = 0.1$, $\omega = 10^5$, desired volume fraction 0.5, and a mesh of 100×50 (unless stated otherwise) isoparametric elements as in section 4.1.

6.2. Results

Representative zero-level set curves from the proposed method are shown in Figures 15a to 15d. The topology of the converged solution is in accordance with the results reported in literature, under the same conditions of domain initialization and loading [8, 23].

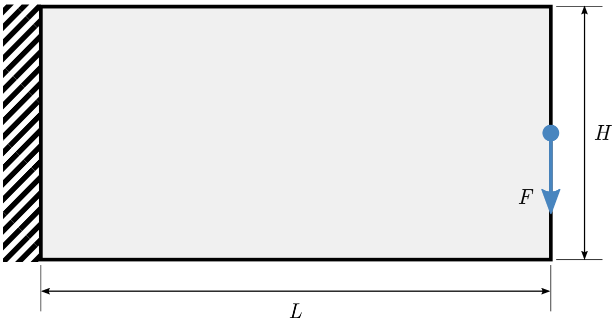


Figure 14: Schematic of the cantilever beam

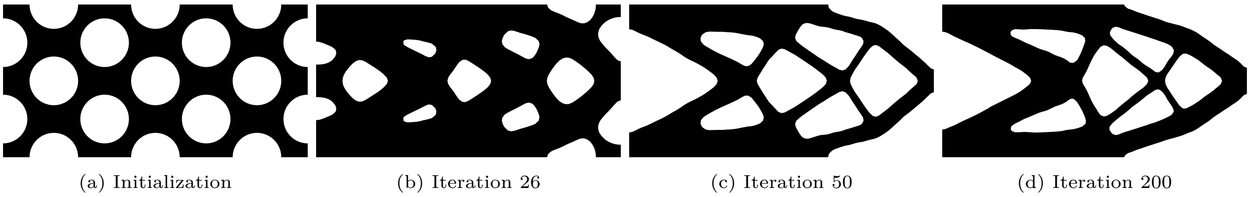


Figure 15: LS curves from selected time iterations of the proposed methodology

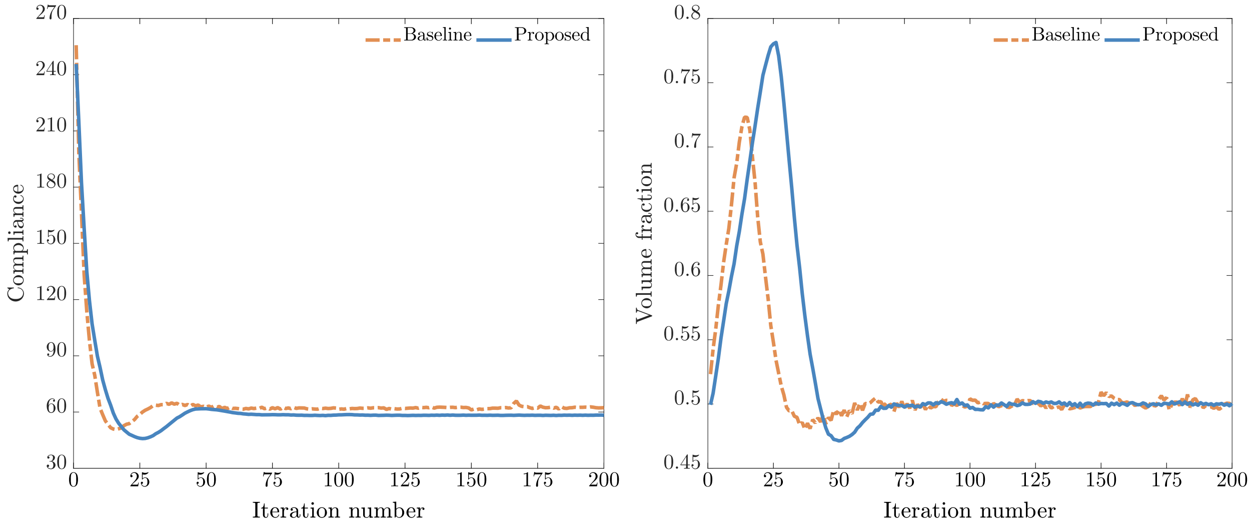
The results of the optimisation algorithms can be seen in the form of the objective and volume constraint curves in Figures 16a and 16b. Both methods stabilize both constraint and objective functions around the same point (circa the 75th iteration). The converged compliance value of the proposed method is lower, while retaining the same volume.

This improvement in compliance, J (see Appendix A), is quantified in Figure 17 for different mesh densities — from 100×50 to 300×150 elements. It can be seen that the converged compliance obtained with the proposed method is $\approx 6\%$ to $\approx 1.5\%$ lower than that obtained with the baseline method. Furthermore, the proposed method is particularly more effective for coarser meshes.

Figure 18 shows side-by-side the zero-level set curves and numerical (mechanical) models from the baseline and proposed methods, in the exact same conditions. The numerical model for the baseline method is characterized by pixelated edges, while the same for the proposed method shows a nearly exact geometric translation between the LS field and the numerical model.

The average number of active DoFs over the iteration range is shown in Figure 19a. It can be observed that the proposed method uses fewer DoFs at any one iteration, when compared to the baseline method.

The average computational time (out of 10 runs) for the example considered here is displayed in Figure 19. It is possible to observe that when considering a more efficient scenario — reinitializing after every 5 iterations — the proposed methodology runtime is slightly higher, being within 5% of the baseline method; when considering a more accurate scenario, i.e.reinitializing every iteration, the proposed methodology is around 25% faster in average.



(a) Objective functions

(b) Volume constraint curves

Figure 16: Comparison of the results from the baseline and proposed optimisation methods

6.3. Analysis

Due to the differences in the baseline and proposed methods, it is expected that the optimisation results are different, despite both starting in the same conditions. Essentially, both methods capture the geometry of the domain differently. This leads to the computation of different displacement fields, and consequently different velocities for the LS field evolution. This may explain the difference in behaviour and final solutions observed from both methodologies.

The accurate geometric representation of the LS field in the numerical (mechanical) model has a direct impact on the accuracy of the mechanical analysis, as reported in Section 5.2. The static mechanical analysis is the basis for the velocity computation (see Appendix A), thus playing an important role in the quality and accuracy of the optimisation process. Consequently, increasing the accuracy of the displacement field by accurately retaining the topology in the model at every iteration leads to an increase in accuracy of the velocity field for the LSM. Furthermore, having the boundary explicitly discretized allows for the direct access to the boundary velocities which is more accurate than using approximated versions of the Dirac delta and Heaviside functions. Ultimately, the increase in accuracy of the physical model and the direct access to the boundary leads to an improvement of the converged optimum solution of the optimisation algorithm, when using the proposed methodology.

It is in coarser meshes that the differences between the methods are most evident, and the proposed method proves more advantageous. Firstly, there is an improvement in the converged optimum solution by using the proposed method; Secondly, the proposed method allows for smaller geometrical features in the topology. By allowing the partitioning of the elements, the method is able to accurately represent features

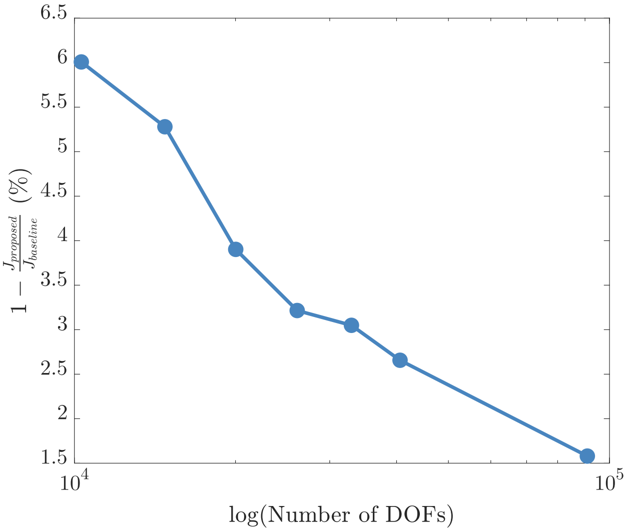


Figure 17: Improvement of the converged optimum solution by using the proposed methodology

smaller than the initial element size.

Floating nodes are used by the proposed methodology to partition elements and effectively populate the boundary with DoFs. Therefore, the initial mesh and element connectivities of the proposed method account for this extra set of nodes and DoFs. Despite this, at any one iteration, since there is a clear boundary and, therefore, a clear separation between void and domain, one can choose a subset of the initial mesh as the active set of DoFs for the analysis. Consequently, the combination of an optimised choice of active DoFs with a more efficient reinitialization scheme allows the proposed methodology to balance the extra computational time associated with the element partitioning and the velocity calculation in the sub-elements; ultimately, this achieves a similar average computation time for the strategy with reinitialization every 5 iterations, and a 25% reduction in computational time for the strategy with reinitialization at every iteration.

The proposed method can therefore be applied to topology optimisation problems, and successfully capture the solution reported from literature while improving on the solution attainable by the baseline methodology.

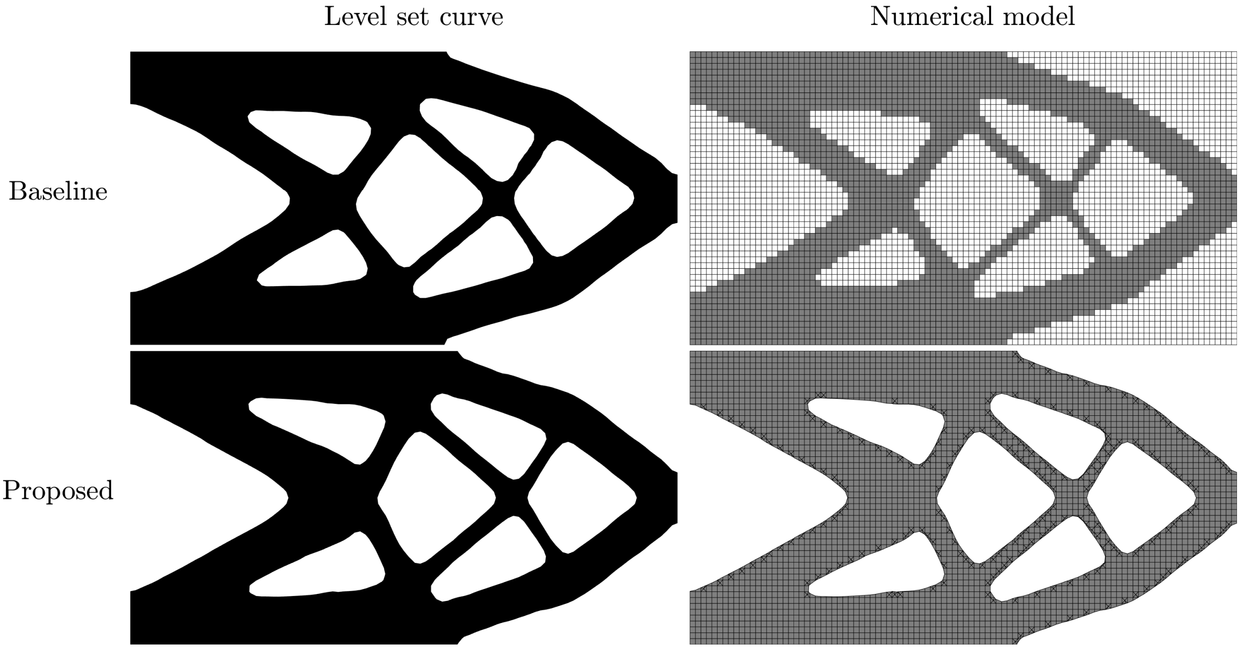
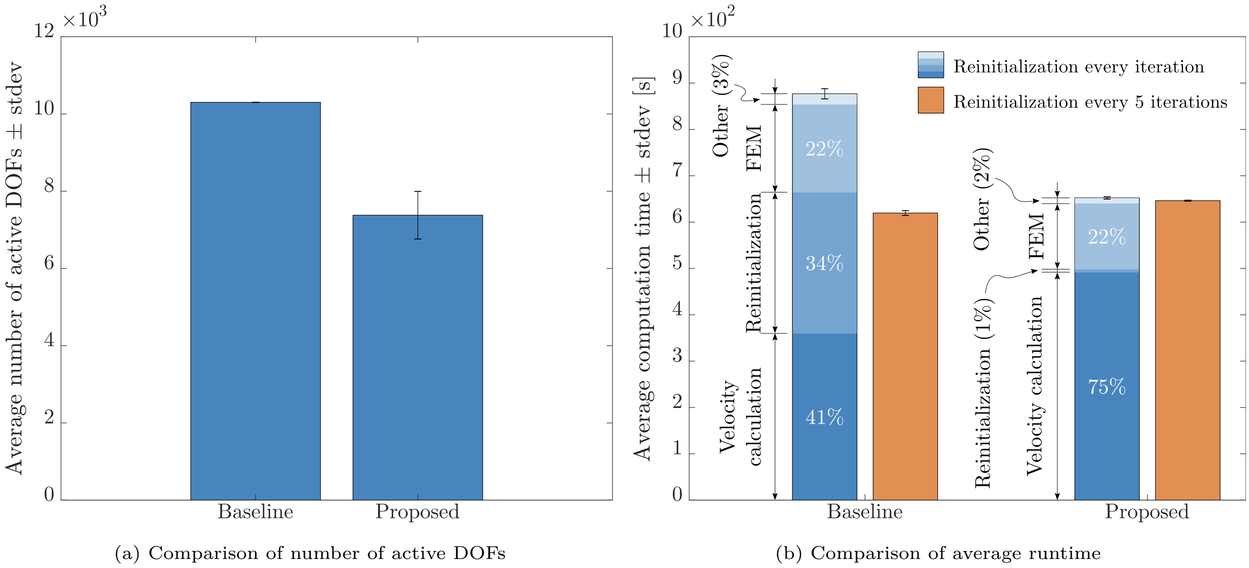


Figure 18: Comparison of the relationship between the numerical models and zero-level set curves of the baseline and the proposed methods



(a) Comparison of number of active DOFs

(b) Comparison of average runtime

Figure 19: Impact on resource efficiency (number of active DOFs) and time efficiency (runtime) of the proposed method when using a 100×50 grid and 200 iterations

7. Application: 2D topology optimisation examples

7.1. *MBB beam*

In this section we introduce a classic topology optimisation problem — the Messerschmitt-Bölkow-Blohm (MBB) beam — for application [30, 38]. This beam is represented in its half-model form in Figure 20. Symmetry boundary conditions are applied in the left edge, a rolling support is introduced in the lower right corner, and a point load is also applied in the top left corner. The parameters used are $L = 3\,\mathrm{mm}$, $H = 2\,\mathrm{mm}$, $F = 1\,\mathrm{N}$, $E = 1\,\mathrm{MPa}$, $\nu = 0.3$, $\alpha = 1$, $\tau = 0.9$, $\lambda^0 = \gamma^0 = 1$ (parameters in the optimisation process as detailed in Appendix A), $c = 0.1$, desired volume fraction 0.5, and a mesh of 60×40 isoparametric elements as in section 4.1.

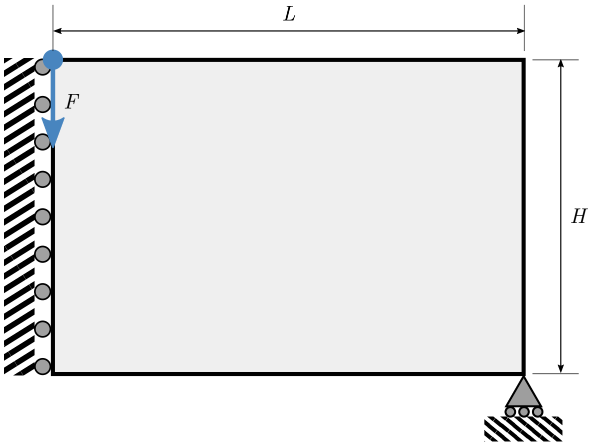


Figure 20: Schematic of the MBB beam (using half-model symmetry)

7.2. *L-bracket*

One other classic topology optimisation example is the L-bracket [38, 39]. This application example consists of a L-shaped domain (Figure 21) in which a load is applied in the middle of the lower vertical right edge.

For this example, the L-shaped domain is obtained by subtracting a small square domain $(0.6L \times 0.6H)$ from the full square domain $(L \times H)$ using the FNM capabilities of the method. For the full square domain, the parameters used are $L = 2\,\mathrm{mm}$, $H = 2\,\mathrm{mm}$, $F = 1\,\mathrm{N}$, $E = 1\,\mathrm{MPa}$, $\nu = 0.3$, $\alpha = 1$, $\tau = 0.9$, $\lambda^0 = \gamma^0 = 1$ (parameters in the optimisation process as detailed in Appendix A), $c = 0.1$, desired volume fraction 0.25, and a mesh of 80×80 isoparametric elements as in section 4.1.

7.3. *Bracket with two holes*

The final application example is based on Nguyen-Xuan et al. [38, 40], and aims at showing the full capabilities of the explicitly discretized boundary. This example is the bracket with two holes, in which the

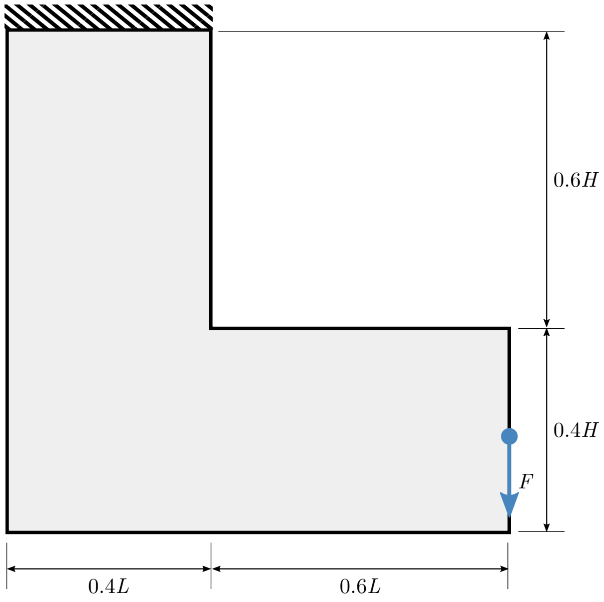


Figure 21: Schematic of the L-bracket

irregular hole boundaries are used for boundary conditions. The bracket is fixed on the left hole and two point loads are applied in the right hole.

The geometry of this example is obtained, as the L-bracket in Section 7.2, using the FNM capabilities. For this case in particular given the more complex geometry a modelling technique based on the R-function disjunction and conjunction is employed [41, 42]. This modelling technique allows us to easily capture the curved geometry and impose non-designable domains (such as the ring around the circles). For this example, the parameters used are $L = 3\,\text{mm}$, $H = 1\,\text{mm}$, $F = 1\,\text{N}$, $E = 1\,\text{MPa}$, $\nu = 0.3$, $\alpha = 1$, $\tau = 0.9$, $\lambda^0 = \gamma^0 = 1$ (parameters in the optimisation process as detailed in Appendix A), $c = 0.1$, desired volume fraction 0.2, and a mesh of 300 × 100 isoparametric elements as in section 4.1.

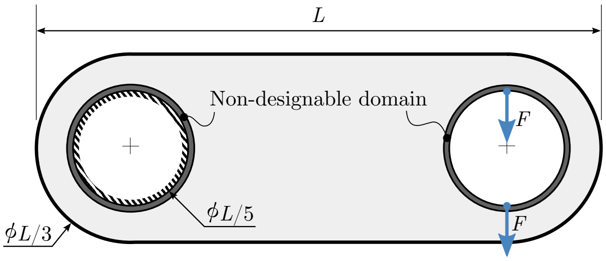


Figure 22: Schematic of the bracket with two holes

7.4. *Results*

The results of the optimisation algorithm can be seen in the form of compliance and volume constraint curves in Figures 23, 26 and 29, for the MBB beam, L-bracket and bracket with two holes examples, respectively.

Furthermore, selected LS curves are presented in Figures 24, 27 and 30 for the MBB beam, L-bracket and bracket with two holes examples, respectively.

he numerical models corresponding to the converged solution (250^{th} iteration) can be seen side-by-side to the respective zero-level set curves in Figure 25 for the MBB beam, Figure 28 for the L-bracket and Figure 31 for the bracket with two holes.

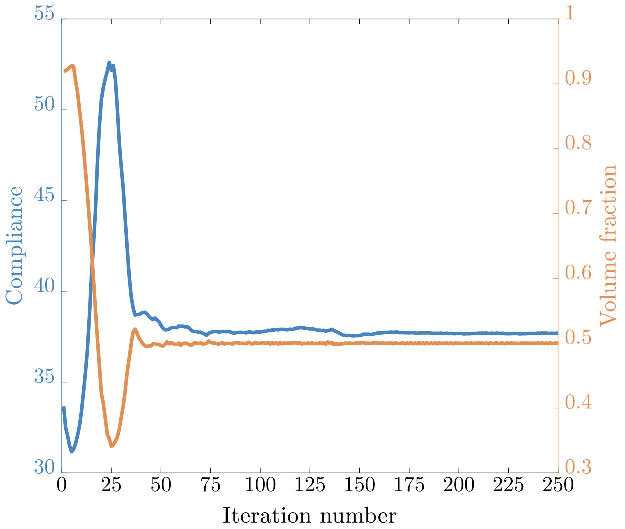
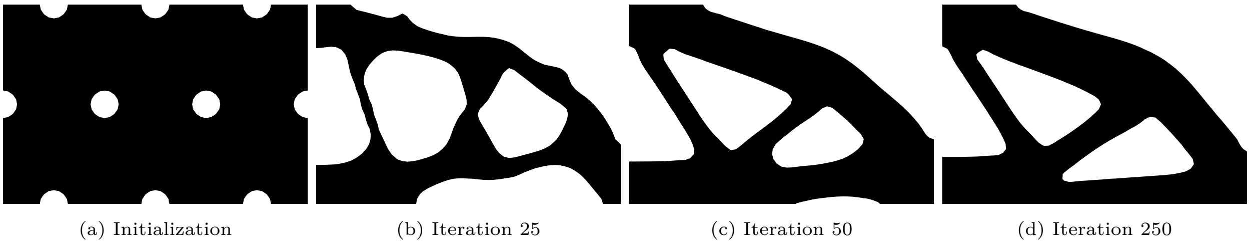


Figure 23: Compliance and volume fraction over the iterations for the MBB example.

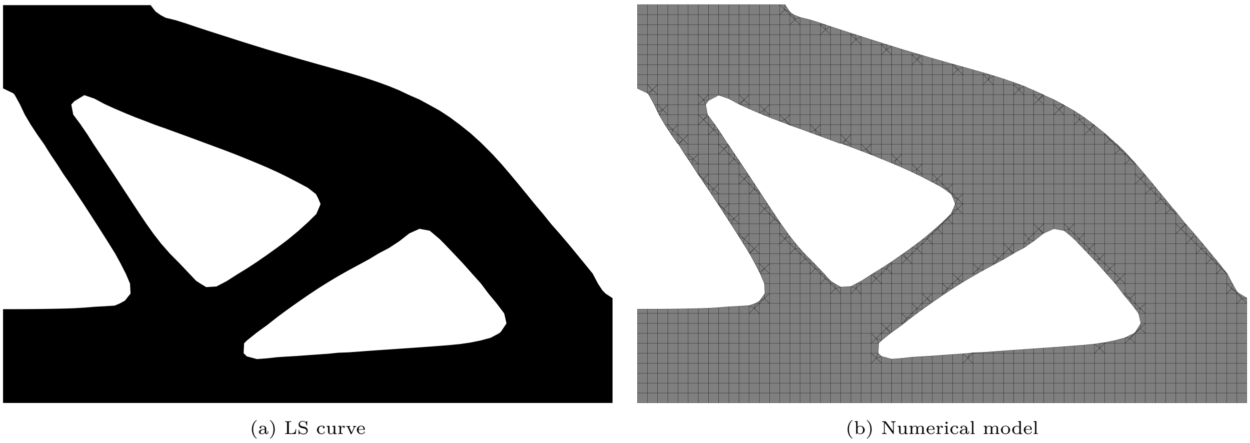


(a) Initialization (b) Iteration 25 (c) Iteration 50 (d) Iteration 250

Figure 24: LS curves from selected time iterations of the MBB beam example

7.5. *Analysis*

Figures 23, 26 and 29 show that the three application examples converged without complications; furthermore, Figures 24, 27 and 30 show that the obtained converged topology is in accordance with results reported in the literature for the same problems [30, 38].



(a) LS curve

(b) Numerical model

Figure 25: Comparison of the converged LS curve and numerical model for the MBB beam example

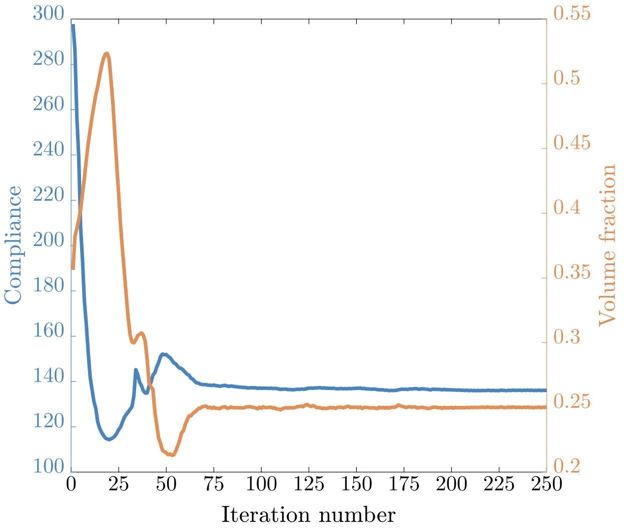
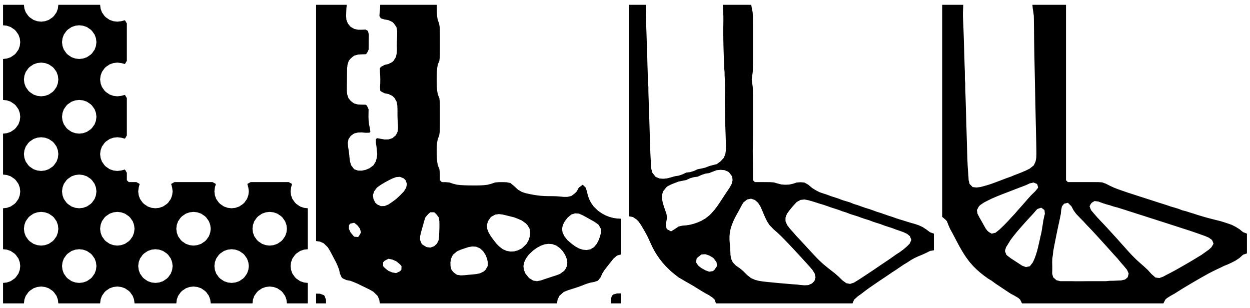


Figure 26: Compliance and volume fraction over the iterations for the L-bracket example

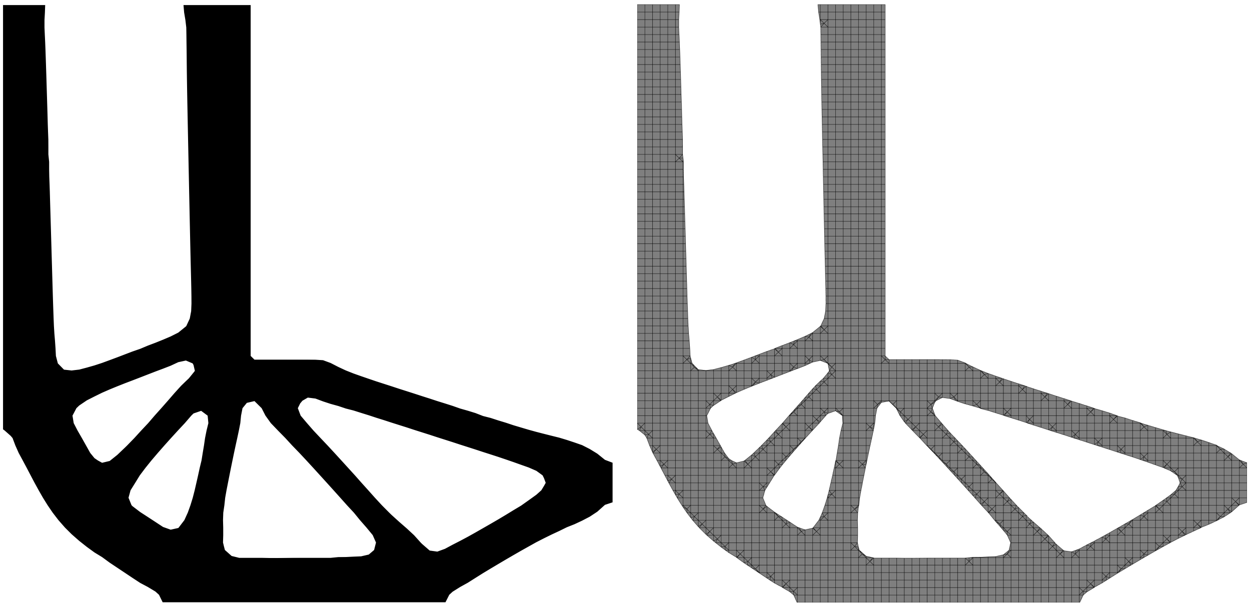
It is important to note that the bracket with two holes application example reports a slower topology evolution in comparison to the other two examples, which is due to the finer mesh used. A finer mesh is used in this example so as to be able to limit the non-designable domain (rings around the two holes) to a relatively small size, which is defined as being the length of two elements.

As mentioned in Sections 7.2 and 7.3, the domains for the L-bracket and bracket with two holes examples were obtained using the FNM capabilities. The numerical model is defined as a square or rectangular domain from which sections are subtracted. This, technically, allows for any domain to be represented (so long as the user implements the required domain subtractions or additions) with the proposed method using a starting uniform quadrilateral mesh; and also, for the imposition of boundary conditions on curvilinear and irregular boundaries using the floating nodes (as is the case with the bracket with two holes example).



(a) Initialization (b) Iteration 25 (c) Iteration 50 (d) Iteration 250

Figure 27: LS curves from selected time iterations of the L-bracket example



(a) LS curve (b) Numerical model

Figure 28: Comparison of the converged LS curve and numerical model for the L-bracket example

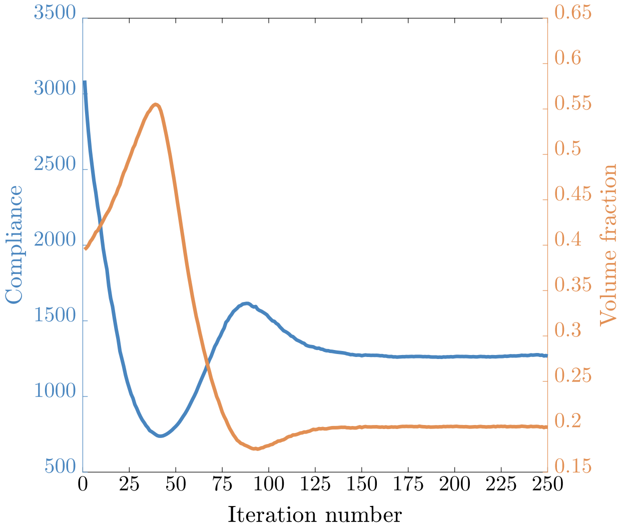
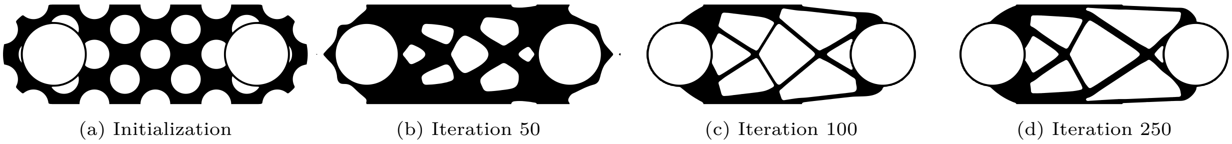


Figure 29: Compliance and volume fraction over the iterations for the bracket with two holes example.

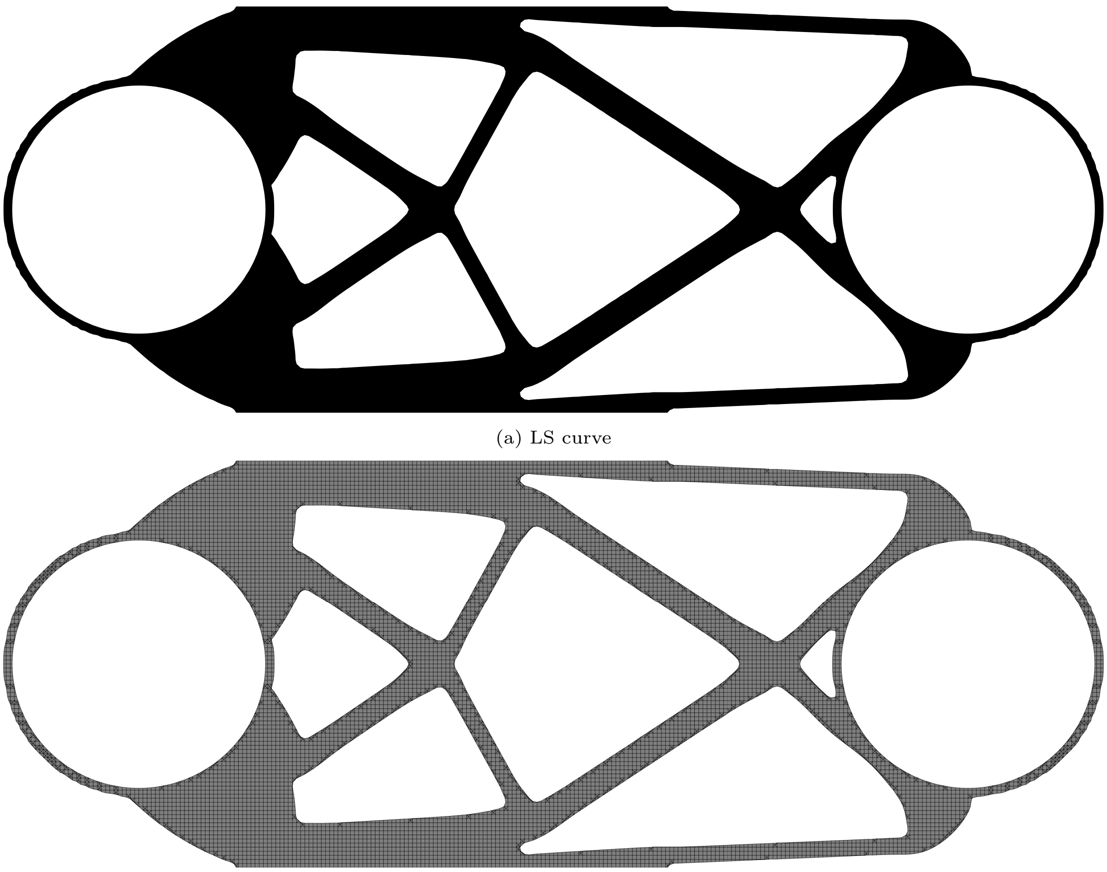

(a) Initialization

(b) Iteration 50

(c) Iteration 100

(d) Iteration 250

Figure 30: LS curves from selected time iterations of the bracket with two holes example



(a) LS curve

(b) Numerical model

Figure 31: Comparison of the converged LS curve and numerical model for the bracket with two holes example

8. Conclusion

This paper proposes the first methodology in the literature that couples the level set method and the floating node method to seamlessly integrate the implicit boundary tracking and movement with an explicitly discretized numerical model.

The explicit boundary representation achieved in our formulation gives direct access to boundary nodes as the boundary evolves, and therefore leads to the following advantages over methods employing fully-implicit boundary representation:

- direct calculation of physical quantities at the boundary of the domain (such as displacement, strain/stress and velocities). Also, the accuracy of these quantities is consistently higher over a wide range of mesh densities (Figure 12 demonstrates this);
- geometrically-accurate translation of the level-set field into the numerical model without any error-inducing sharp edges or blurred regions (as shown in Figures 13 and 18);
- ability to impose boundary conditions directly on the evolving boundary;
- simpler and more efficient level-set reinitialization (as can be seen in Figure 10); and
- reduction of the number of elements required at each iteration, thus resulting in a more efficient computation in terms of use of resources, and comparable in terms of total computation time (as reported in Figure 19).

The methodology proposed relies on the floating node method for the boundary discretization. Therefore, when compared with literature on explicit boundary representation, our method stands out as providing:

- easier and more streamlined implementation in generic FE packages through user-element functionality, since the original mesh and element connectivities are not altered; and
- a mesh partitioning scheme that does not require remeshing or other standard meshing operations; thus, in principle, it is a more efficient methodology.

In the current work, we chose to showcase the technology with an application to structural topology optimisation where the following features are advantageous over current methods:

- retaining an accurate numerical model of the domain in every iteration (as depicted in Figure 18);
- directly computing sensitivities at the boundary, resulting in an improvement of the optimum solution attained, specially for coarser meshes (as can be seen in Figure 17); and

- performing the optimisation iterations without the need to modify the initial mesh, thus leaving the input file unchanged.

In summary, the proposed formulation constitutes a new, more accurate and more efficient approach for modelling evolving boundaries. It has the potential to create a step change in areas including structural design and optimisation, fluid-structure interaction, additive manufacturing and multi-phase flow.

9. Acknowledgement

The funding from the Engineering and Physical Sciences Research Council (EPSRC) is gratefully acknowledged.

A. Topology optimisation

A.1. Introduction

This section provides a brief review in topology optimisation. The readers are referred to [8, 31, 32] for more comprehensive expositions of this topic, as this section will contain only the necessary theory for the application of the proposed method to topology optimisation in Section 2.2 and Section 6.

The topology optimisation method described herein is only capable of determining what is the best design evolution given the current boundary topology; the ability of generating new voids was not considered for simplicity.

Topology optimisation relies on the static mechanic analysis for the computation of the design evolution. The method computes sensitivities in order to determine what is the best direction for the boundary to move, in order to improve some objective function.

For a static analysis, consider box D as a general domain from which one can obtain a sub-domain with volume Ω and a void region accounting for the remaining volume (Figure 32). The static equilibrium of the body with volume Ω contained in box D, with tractions $\mathbf{t}$ acting on Γ_t, and with displacements imposed as supports along Γ_u, leads to the displacement vector $\mathbf{u}$. The body is considered elastic and in the plane-stress domain. The strain tensor $\boldsymbol{\epsilon}$ is related to the displacements through the cartesian differential operator $\mathcal{L}_x$. The stress tensor $\boldsymbol{\sigma}$ is related to the strains through a constitutive relation of the form $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}$, where $\mathbf{D}$ is the elastic constitutive tensor.

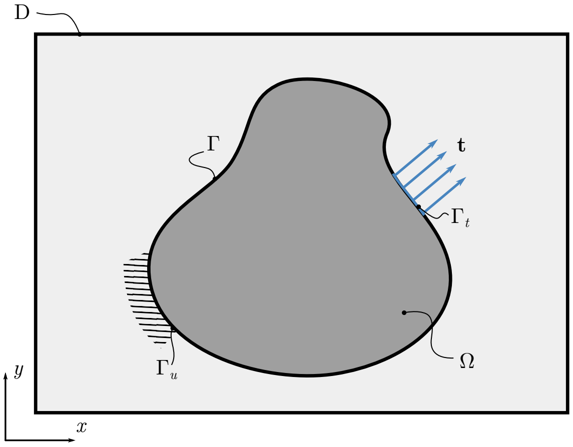


Figure 32: Elastic domain Ω as part of D with the corresponding boundary conditions

A.2. Sensitivity

The optimisation process consists of minimising the mean compliance functional when subjected to a volume constraint. Thus, it is possible to write the following:

$$\begin{aligned} \text{minimise} \quad & J(\mathbf{u}, \Omega) = \int_\Omega \boldsymbol{\epsilon}^{\mathrm{T}}(\mathbf{u})\boldsymbol{\sigma}(\mathbf{u})\,\mathrm{d}\Omega\,, \\ \text{subject to} \quad & \int_\Omega \mathrm{d}\Omega \leq \Omega_{\max}\,. \end{aligned} \tag{23}$$

To implement the constrained optimisation problem, an augmented Lagrangian approach was chosen, similar to Xing et al. [23]. Therefore, the Lagrangian of the problem can be written as

$$\mathcal{L} = J + \lambda \left(\int_\Omega \mathrm{d}\Omega - \Omega_{\max} \right) + \frac{\gamma}{2} \left(\int_\Omega \mathrm{d}\Omega - \Omega_{\max} \right)^2 , \tag{24}$$

where λ is a Lagrange multiplier, and γ is a penalty term; both are updated every iteration i according to:

$$\begin{aligned} \gamma^i &= \tau\gamma^{i-1}\,, \\ \lambda^i &= \lambda^{i-1} + \frac{\gamma^i}{2} \left(\int_\Omega \mathrm{d}\Omega - \Omega_{\max} \right) , \end{aligned} \tag{25}$$

with τ a real number in $]0, 1[$. The movement of the boundary can be simplified to consider only the normal direction [8, 23]. Hence, the sensitivity (often called shape derivative in this context) of the Lagrangian to a change in the moving boundary Γ in the arbitrary normal direction $\boldsymbol{\theta}$ is

$$\dot{\mathcal{L}} = \int_\Gamma \left(\bar{\lambda} - \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon} \right) \boldsymbol{\theta} \cdot \mathbf{n}\,\mathrm{d}\Gamma\,, \tag{26}$$

where $\bar{\lambda}$ is a modified Lagrange multiplier defined as

$$\bar{\lambda} = \max\left(0, \lambda^i\right) , \tag{27}$$

to satisfy the Karush-Kuhn-Tucker conditions.

One direction that guarantees the descent of $\mathcal{L}$ is along $-\dot{\mathcal{L}}$; hence we can set

$$\boldsymbol{\theta} = -\left(\bar{\lambda} - \boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}\boldsymbol{\epsilon} \right) \mathbf{n}\,. \tag{28}$$

A.3. FEM discretization

The displacement vector $\mathbf{u}$ is related to the displacement DoF vector $\mathbf{q}_u$ using the matrix of linear shape functions for the displacements $\mathbf{N}_u$:

$$\mathbf{u} = \mathbf{N}_u \mathbf{q}_u\,. \tag{29}$$

The strain and stress tensor are defined respectively as

$$\begin{aligned}\boldsymbol{\epsilon} &= \mathcal{L}_x \mathbf{u} = \mathcal{L}_\xi \mathbf{J}^{-1} \mathbf{q}_u = \mathbf{B}\mathbf{q}_u \,, \\ \boldsymbol{\sigma} &= \mathbf{D}\boldsymbol{\epsilon} = \mathbf{D}\mathbf{B}\mathbf{q}_u \,,\end{aligned} \tag{30}$$

where $\mathbf{B} = \mathcal{L}_\xi \mathbf{J}^{-1}$; $\mathbf{J}$ is Jacobian matrix; and $\mathcal{L}_\xi$ is the differential operator in natural coordinates.

The FEM equations for the static mechanical analysis are written in matrix form as

$$\mathbf{K}_u \mathbf{q}_u = \mathbf{f}_u \,, \tag{31}$$

in which the stiffness matrix and force vector are given respectively by

$$\begin{aligned}\mathbf{K}_u &= \int_\xi \int_\eta \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} \det(\mathbf{J}) \, \mathrm{d}\xi \, \mathrm{d}\eta \,, \\ \mathbf{f}_u &= \mathbf{f} \,,\end{aligned} \tag{32}$$

where $\mathbf{f}$ is the force vector corresponding to the applied load.

The sensitivities for the optimisation can also be discretized. Thus, combining Equations 26 and 28 yields the elemental sensitivity

$$v_{\mathrm{n}}^e = \mathbf{q}_u^{\mathrm{T}} \mathbf{K}_u^{\mathrm{e}} \mathbf{q}_u \,, \tag{33}$$

in which the superscript e indicates an elemental quantity. Equation 33 is then averaged at every node and normalized taking into account the area of each element. This leads to the nodal sensitivity vector, which for the j^{th} node yields:

$$\mathbf{v}_{\mathrm{n}}(j) = \frac{1}{n^e} \sum_{i=1}^{n^e} \frac{v_{\mathrm{n}i}^e}{A_i^e} - \bar{\lambda} \,. \tag{34}$$

Alternatively, for the case in which the boundary is explicitly discretized in the numerical model, the nodal sensitivity vector at the boundary is given by

$$\mathbf{v}_{\mathrm{n}}(j) = \boldsymbol{\epsilon}^{\mathrm{T}}(j) \boldsymbol{\sigma}(j) - \bar{\lambda} \,. \tag{35}$$

References

[1] Z.-Q. Zhang, G. R. Liu, B. C. Khoo, A three dimensional immersed smoothed finite element method (3D IS-FEM) for fluid–structure interaction problems, Computational Mechanics 51 (2) (2013) 129–150. doi:10.1007/s00466-012-0710-1. URL http://link.springer.com/10.1007/s00466-012-0710-1

[2] A.-K. Tornberg, B. Engquist, A finite element based level-set method for multiphase flow applications, Computing and Visualization in Science 3 (1-2) (2000) 93–101. doi:10.1007/s007910050056. URL http://link.springer.com/10.1007/s007910050056

[3] R. Comminal, M. P. Serdeczny, D. B. Pedersen, J. Spangenberg, Numerical modeling of the strand deposition flow in extrusion-based additive manufacturing, Additive Manufacturing 20 (2018) 68–76. doi:10.1016/J.ADDMA.2017.12.013.
URL https://www.sciencedirect.com/science/article/pii/S2214860417305079?via{%}3Dihub

[4] H. Bikas, P. Stavropoulos, G. Chryssolouris, Additive manufacturing methods and modelling approaches: a critical review, The International Journal of Advanced Manufacturing Technology 83 (1-4) (2016) 389–405. doi:10.1007/s00170-015-7576-2.
URL http://link.springer.com/10.1007/s00170-015-7576-2

[5] J. Chessa, P. Smolinski, T. Belytschko, The extended finite element method (XFEM) for solidification problems, International Journal for Numerical Methods in Engineering 53 (8) (2002) 1959–1977. doi:10.1002/nme.386.
URL http://doi.wiley.com/10.1002/nme.386

[6] B. Chen, S. Pinho, N. De Carvalho, P. Baiz, T. Tay, A floating node method for the modelling of discontinuities in composites, Engineering Fracture Mechanics 127 (2014) 104–134. doi:10.1016/J.ENGFRACMECH.2014.05.018.
URL https://www.sciencedirect.com/science/article/pii/S0013794414001829

[7] N. De Carvalho, B. Chen, S. Pinho, J. Ratcliffe, P. Baiz, T. Tay, Modeling delamination migration in cross-ply tape laminates, Composites Part A: Applied Science and Manufacturing 71 (2015) 192–203. doi:10.1016/J.COMPOSITESA.2015.01.021.
URL https://www.sciencedirect.com/science/article/pii/S1359835X15000329

[8] G. Allaire, F. Jouve, A. M. Toader, Structural optimization using sensitivity analysis and a level-set method, Journal of Computational Physics 194 (1) (2004) 363–393. doi:10.1016/j.jcp.2003.09.032.
URL https://www.sciencedirect.com/science/article/pii/S002199910300487X?via{%}3Dihub

[9] J. Luo, Z. Luo, L. Chen, L. Tong, M. Y. Wang, A semi-implicit level set method for structural shape and topology optimization, Journal of Computational Physics 227 (11) (2008) 5561–5581. doi:10.1016/J.JCP.2008.02.003.
URL https://www.sciencedirect.com/science/article/pii/S0021999108000995

[10] G. Allaire, C. Dapogny, P. Frey, Shape optimization with a level set based mesh evolution method, Computer Methods in Applied Mechanics and Engineering 282 (2014) 22–53. doi:10.1016/J.CMA.2014.08.028.
URL https://www.sciencedirect.com/science/article/pii/S0045782514003077

[11] A. N. Christiansen, M. Nobel-Jørgensen, N. Aage, O. Sigmund, J. A. Bærentzen, Topology optimization using an explicit interface representation, Structural and Multidisciplinary Optimization 49 (3) (2014) 387–399. doi:10.1007/s00158-013-0983-9.
URL http://link.springer.com/10.1007/s00158-013-0983-9

[12] J. Stanford, T. Fries, A higher-order conformal decomposition finite element method for plane B-rep geometries, Computers & Structures 214 (2019) 15–27. doi:10.1016/J.COMPSTRUC.2018.12.006.
URL https://www.sciencedirect.com/science/article/pii/S0045794918311271?via{%}3Dihub

[13] M. J. Borden, C. V. Verhoosel, M. A. Scott, T. J. Hughes, C. M. Landis, A phase-field description of dynamic brittle fracture, Computer Methods in Applied Mechanics and Engineering 217-220 (2012) 77–95. doi:10.1016/J.CMA.2012.01.008.
URL https://www.sciencedirect.com/science/article/pii/S0045782512000199

[14] S. Riehl, P. Steinmann, On structural shape optimization using an embedding domain discretization technique, International Journal for Numerical Methods in Engineering 109 (9) (2017) 1315–1343. doi:10.1002/nme.5326.
URL http://doi.wiley.com/10.1002/nme.5326

[15] M. Y. Wang, X. Wang, "Color" level sets: a multi-phase method for structural topology optimization with multiple materials, Computer Methods in Applied Mechanics and Engineering 193 (6-8) (2004) 469–496. doi:10.1016/J.CMA.2003.10.008.

URL https://www.sciencedirect.com/science/article/pii/S0045782503005644

[16] R. Picelli, S. Townsend, C. Brampton, J. Norato, H. Kim, Stress-based shape and topology optimization with the level set method, Computer Methods in Applied Mechanics and Engineering 329 (2018) 1–23. doi:10.1016/J.CMA.2017.09.001.
URL https://www.sciencedirect.com/science/article/pii/S0045782517304395

[17] E. Burman, D. Elfverson, P. Hansbo, M. G. Larson, K. Larsson, Shape optimization using the cut finite element method, Computer Methods in Applied Mechanics and Engineering 328 (2018) 242–261. doi:10.1016/J.CMA.2017.09.005.
URL https://www.sciencedirect.com/science/article/pii/S0045782516316073

[18] Q. Xia, M. Y. Wang, T. Shi, A level set method for shape and topology optimization of both structure and support of continuum structures, Computer Methods in Applied Mechanics and Engineering 272 (2014) 340–353. doi:10.1016/J.CMA.2014.01.014.
URL https://www.sciencedirect.com/science/article/pii/S0045782514000279

[19] V. J. Challis, A discrete level-set topology optimization code written in Matlab, Structural and Multidisciplinary Optimization 41 (3) (2010) 453–464. doi:10.1007/s00158-009-0430-0.
URL http://link.springer.com/10.1007/s00158-009-0430-0

[20] M. Yulin, W. Xiaoming, A level set method for structural topology optimization and its applications, Advances in Engineering Software 35 (7) (2004) 415–441. doi:10.1016/J.ADVENGSOFT.2004.06.004.
URL https://www.sciencedirect.com/science/article/pii/S0965997804000705

[21] M. Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, Computer Methods in Applied Mechanics and Engineering 192 (1-2) (2003) 227–246. doi:10.1016/S0045-7825(02)00559-5.
URL https://www.sciencedirect.com/science/article/pii/S0045782502005595

[22] N. P. van Dijk, K. Maute, M. Langelaar, F. van Keulen, Level-set methods for structural topology optimization: a review, Structural and Multidisciplinary Optimization 48 (3) (2013) 437–472. doi:10.1007/s00158-013-0912-y.
URL http://link.springer.com/10.1007/s00158-013-0912-y

[23] X. Xing, P. Wei, M. Y. Wang, A finite element-based level set method for structural optimization, International Journal for Numerical Methods in Engineering 82 (7) (2009) n/a–n/a. doi:10.1002/nme.2785.
URL http://doi.wiley.com/10.1002/nme.2785

[24] S. Valance, R. de Borst, J. Réthoré, M. Coret, A partition-of-unity-based finite element method for level sets, International Journal for Numerical Methods in Engineering 76 (10) (2008) 1513–1527. doi:10.1002/nme.2371.
URL http://doi.wiley.com/10.1002/nme.2371

[25] Q. Xia, T. Shi, Optimization of structures with thin-layer functional device on its surface through a level set based multiple-type boundary method, Computer Methods in Applied Mechanics and Engineering 311 (2016) 56–70. doi:10.1016/j.cma.2016.08.001.

[26] S. H. Ha, S. Cho, Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh, Computers and Structures 86 (13-14) (2008) 1447–1455. doi:10.1016/j.compstruc.2007.05.025.

[27] S. H. Nguyen, H.-G. Kim, Level set based shape optimization using trimmed hexahedral meshes, Computer Methods in Applied Mechanics and Engineering 345 (2019) 555–583. doi:10.1016/J.CMA.2018.11.006.
URL https://www.sciencedirect.com/science/article/pii/S0045782518305619

[28] Y. Hahn, J. I. Cofer, Study of Parametric and Non-Parametric Optimization of a Rotor-Bearing System, in: Volume 7A: Structures and Dynamics, American Society of Mechanical Engineers, 2014. doi:10.1115/GT2014-25095.
URL https://asmedigitalcollection.asme.org/GT/proceedings/GT2014/45769/D{\unhbox\voidb@x\bgroup\accent127u\protect\penalty\@M\hskip\z@skip\egroup}sseldorf,Germany/236053

[29] S. Yamasaki, T. Nomura, A. Kawamoto, K. Sato, S. Nishiwaki, A level set-based topology optimization method targeting metallic waveguide design problems, International Journal for Numerical Methods in Engineering 87 (9) (2011) 844–868.

doi:10.1002/nme.3135.
URL http://doi.wiley.com/10.1002/nme.3135

[30] O. Sigmund, A 99 line topology optimization code written in Matlab, Structural and Multidisciplinary Optimization 21 (2) (2001) 120–127. doi:10.1007/s001580050176.
URL http://link.springer.com/10.1007/s001580050176

[31] O. Sigmund, K. Maute, Topology optimization approaches, Structural and Multidisciplinary Optimization 48 (6) (2013) 1031–1055. doi:10.1007/s00158-013-0978-6.
URL http://link.springer.com/10.1007/s00158-013-0978-6

[32] G. Allaire, F. Jouve, A.-M. Toader, A level-set method for shape optimization, Comptes Rendus Mathematique 334 (12) (2002) 1125–1130. doi:10.1016/S1631-073X(02)02412-3.
URL https://www.sciencedirect.com/science/article/pii/S1631073X02024123

[33] B. Chen, T. Tay, S. Pinho, V. Tan, Modelling the tensile failure of composites with the floating node method, Computer Methods in Applied Mechanics and Engineering 308 (2016) 414–442. doi:10.1016/J.CMA.2016.05.027.
URL https://www.sciencedirect.com/science/article/pii/S0045782516304388

[34] B. Chen, T. Tay, S. Pinho, V. Tan, Modelling delamination migration in angle-ply laminates, Composites Science and Technology 142 (2017) 145–155. doi:10.1016/J.COMPSCITECH.2017.02.010.
URL https://www.sciencedirect.com/science/article/pii/S0266353816308818

[35] E. Kocaman, B. Chen, S. Pinho, A polymorphic element formulation towards multiscale modelling of composite structures, Computer Methods in Applied Mechanics and Engineering 346 (2019) 359–387. doi:10.1016/J.CMA.2018.12.004.
URL https://www.sciencedirect.com/science/article/pii/S0045782518306029

[36] T.-P. Fries, T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications, International Journal for Numerical Methods in Engineering 84 (3) (2010) n/a–n/a. doi:10.1002/nme.2914.
URL http://doi.wiley.com/10.1002/nme.2914

[37] J.-H. Song, P. M. A. Areias, T. Belytschko, A method for dynamic crack and shear band propagation with phantom nodes, International Journal for Numerical Methods in Engineering 67 (6) (2006) 868–893. doi:10.1002/nme.1652.
URL http://doi.wiley.com/10.1002/nme.1652

[38] H. Nguyen-Xuan, A polytree-based adaptive polygonal finite element method for topology optimization, International Journal for Numerical Methods in Engineering 110 (10) (2017) 972–1000. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.5448, doi:10.1002/nme.5448.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.5448

[39] Q. Xia, T. Shi, S. Liu, M. Y. Wang, A level set solution to the stress-based structural shape and topology optimization, Computers and Structures 90-91 (1) (2012) 55–64. doi:10.1016/j.compstruc.2011.10.009.

[40] C. Talischi, G. H. Paulino, A. Pereira, I. F. Menezes, PolyTop: A Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes, Structural and Multidisciplinary Optimization 45 (3) (2012) 329–357. doi:10.1007/s00158-011-0696-x.

[41] S. Cai, W. Zhang, Stress constrained topology optimization with free-form design domains, Computer Methods in Applied Mechanics and Engineering 289 (2015) 267–290. doi:10.1016/j.cma.2015.02.012.

[42] V. Shapiro, Theory of R-functions and Applications: A Primer, Computer Science Technical Reports (1991).