Mathematics and Computers in Simulation 218 (2024) 266–291

Contents lists available at ScienceDirect

Mathematics and Computers in Simulation

journal homepage: www.elsevier.com/locate/matcom

Original articles

Mathematical perspective on XFEM implementation for models involving contribution on interfaces

M.T. Cao-Rial [a,b], C. Moreno [c], P. Quintela [a,d,*]

[a] CITMAga, Campus Vida, 15782 Santiago de Compostela, Spain
[b] Dept. Matemáticas, Universidade da Coruña, Spain
[c] Dept. de Estadística, Investigación Operativa y Cálculo Numérico Universidad Nacional de Educación a Distancia, Spain
[d] Dept. de Matemática Aplicada, Universidade de Santiago de Compostela, Spain

ARTICLE INFO

Keywords:
Extended finite element method
Mathematical tools for numerical implementation
Integration over interfaces
Rayleigh waves

ABSTRACT

Models involving interfaces with discontinuities or even singularities of some fields across them are very frequent in real life problems modelling. In the last decades, the use of the eXtended Finite Element Method (XFEM) instead of the traditional FEM has become more and more popular, mainly because of two advantages: the mesh of the domain can be independent of the interface position, therefore avoiding remeshing, and it allows to enrich an area with specific shape functions fitted to the particular properties (singularities, discontinuities) of the expected solution, obtaining more accurate results with less computational efforts. Nevertheless, a critical point of XFEM is its implementation since it varies from one problem to another, due to the different kind (and number) of degrees of freedom on each node. A diligent organization of nodes, degrees of freedom and enrichment functions is fundamental to achieve an efficient implementation. Our aim in this paper is to provide a common reference framework for the implementation of XFEM from a mathematical point of view, providing the readers with a set of tools that will allow them to apply it to any kind of problem. To this aim, we present a detailed description of XFEM implementation, with special emphasis on the terms that involve integration over interfaces. The proposed tools are presented in a general context, and as an example, we will apply them to a problem of solids mechanics. In particular, we will contextualize the procedure on a Rayleigh waves propagation problem in a cracked structure considering a Signorini contact condition on the crack sides.

1. Introduction

When the solution of a problem involves jumps or singularities, for an accurate approximation of the actual solution, the standard FEM is very expensive, computationally speaking. It usually needs very refined meshes, and even re-meshing from time to time if the position of the singularities or defects evolves. Numerical simulation of cracks and effective contact between their lips, or of free boundary problems with surface tension at the interface, are good examples where the choice of FEM is debatable because of its high computational cost.

The eXtended Finite Element Method (XFEM) introduces local enrichments to the standard shape functions to replicate the discontinuities or the singular behaviour of the solution. Examples are the contact problems, the propagation of cracks or the movement of interfaces in two-phase problems involving free boundaries (see [1–3]). Such local enrichments allow a more accurate

* Corresponding author at: CITMAga, Campus Vida, 15782 Santiago de Compostela, Spain.
E-mail addresses: teresa.cao@udc.es (M.T. Cao-Rial), cmoreno@ccia.uned.es (C. Moreno), peregrina.quintela@usc.es (P. Quintela).

https://doi.org/10.1016/j.matcom.2023.11.012
Received 8 May 2023; Received in revised form 20 July 2023; Accepted 9 November 2023
Available online 18 November 2023

approximation of non-smooth solutions without the need of creating a new mesh with every evolution of the interface and also avoiding mesh refining to accurately capture the singularities where present. So, XFEM offers an obvious computational advantage in such cases. There are some references that deal with the implementation of the XFEM in various contexts of solid and fluid mechanics (see [4–9]), or most recently [10] for COMSOL Multiphysics or Abaqus/Explicit [11], but none of them tackles the implementation of terms that deal with jumps on the interface.

There is already a wide variety of literature concerning the effectiveness of solving this type of problems with XFEM methodology. For instance, an enriched FEM technique for thermo-mechanical contact problems is employed in [12], and in [13], in which a formulation allowing crack lips contact in terms of a penalty formulation for normal contact is presented. The XFEM formulation is also discussed in [14] for modelling crack growth with frictional contact, or in [15] for fatigue crack growth of interfacial cracks in bi-layered materials. Also, in fluid mechanics, the literature highlighting its advantages is extensive. Thus, in [16] different enrichment schemes and time-integration schemes within the XFEM for two-phase and free-surface flows are investigated. Also, in [17] an XFEM approach is presented in the case of a quasi-static Stokes n-phase flow. Problems of fluid–structure interaction are dealt with in [18–22], while in [23] a review of its application to fractured porous media is done. One can also find applications to acoustics [24], materials science (corrosion) [25] or environmental science [26].

In [27] the authors tackled the particularities of the XFEM method related to the integration on enriched elements in a general framework. In particular, they introduced an automatic partition technique for elements with enriched nodes, in the two- and three-dimensional cases. The proposed procedure avoids a casuistic analysis for each element crossed by the interface, and it facilitates obtaining new schemes for the numerical integration over them from the classical quadrature rules. However, and to our knowledge, nothing has been published concerning a detailed description of the implementation of the method when the model involves integration on the interface.

For this reason, in this paper we focus now on the general implementation of the XFEM, to allow the process to be as automatic as possible in aspects such as the contribution to the stiffness matrix of the elements crossed by the interface, or the assembling of the interface terms that usually appear when there is contact, friction, or a jump by surface tension. The different kinds of degrees of freedom that might appear on the same problem, the local character of the enrichments, which might apply to some nodes now, and to different ones later in time, if the interface were to evolve, has a big influence on the discrete matrices and even more when the model involves integration on the interface terms. Therefore, a good systematization of the implementation is a critical point to guarantee efficient and accurate results. In this work a general methodology will be presented, which can be applied to any model, with minor changes. To systematize the discretization process, we will present a series of mathematical tools that will allow us to easily define and establish the local to global mappings needed for an automatic implementation of the elementary matrices. Firstly, we will need to define the different kind of nodes, elements, and edge sets that depend on the type of problem and on the singularities of the expected solution. Then, we define some embeddings relating the elements of one set to the others. These embeddings will play a key role for computing and assembling the elementary matrices appearing in the mass, stiffness or jump on the interface terms. Also, all the local-to-global mappings needed for an automatic implementation of the elementary matrices will be detailed. This rigorous study of the particularities that the XFEM implementation presents compared with the classical FEM has never been approached (to the authors knowledge) and it establishes the key points to be taken into account when implementing XFEM to simulate many different problems.

The main feature of the XFEM is that the mesh is not adapted to the interface. In this way, the integration on the interface cannot be achieved through integration over element boundaries, so all the nodes in one element will contribute to the integral term, but, depending on the problem, not all their associated degrees of freedom will. Therefore, a key point will be the selection of the degrees of freedom that actually contribute to the elementary matrices to be computed. To this end, the proposed methodology will use the embedding operators described in this work which will allow us to select some specific degrees of freedom on each element, computed automatically, and it can be applied and/or generalized to any XFEM enrichment. The proposed methodology is therefore applicable to any discretization algorithm, and leaves the user freedom to transform the proposed operators to their corresponding formulation in classes, structures, arrays, etc. for being used in any programming language.

The outline of the paper is as follows: Firstly, in Section 2 we present the schematics of the implementation process, from the physical model to the various parts of the XFEM programme that will be detailed in the following sections. Then, in Section 3 we will present the example model that will serve as a practical use case for the XFEM implementation methodology showed in this paper. It is a plain strain problem for the contact between the lips of a crack. But again, it is just an example, since the methodology is presented in such way that might be applied to 3D problems, or to many other problems involving contributions on the interface, like the movement of bubbles in a fluid or phase change problems with moving boundary among others. Then, in Section 4 we will introduce the basic mathematical elements that will vertebrate the implementation procedure. In particular, the notations for the nodes, elements, and edge sets will be presented there as well as the embeddings that relate them all. Next, in Section 5 we will review the XFEM approximation and enrichment functions considered for the framework problem. Two different kind of enrichments will be considered, so that any generalization to more kinds would be straightforward. In Section 6 we will briefly recall some results concerning the integration on enriched elements and the resulting quadrature formulas. In Section 7 we will present the discretization of the differential operators involved. In Section 8 we will propose coherent local-to-global mappings in order to achieve an efficient matrices assembling as well as the integration procedure over the interface. Finally, in Section 9 some numerical results and the performance of XFEM over FEM will be shown, and in Section 10 the main results of this work will be summarized.

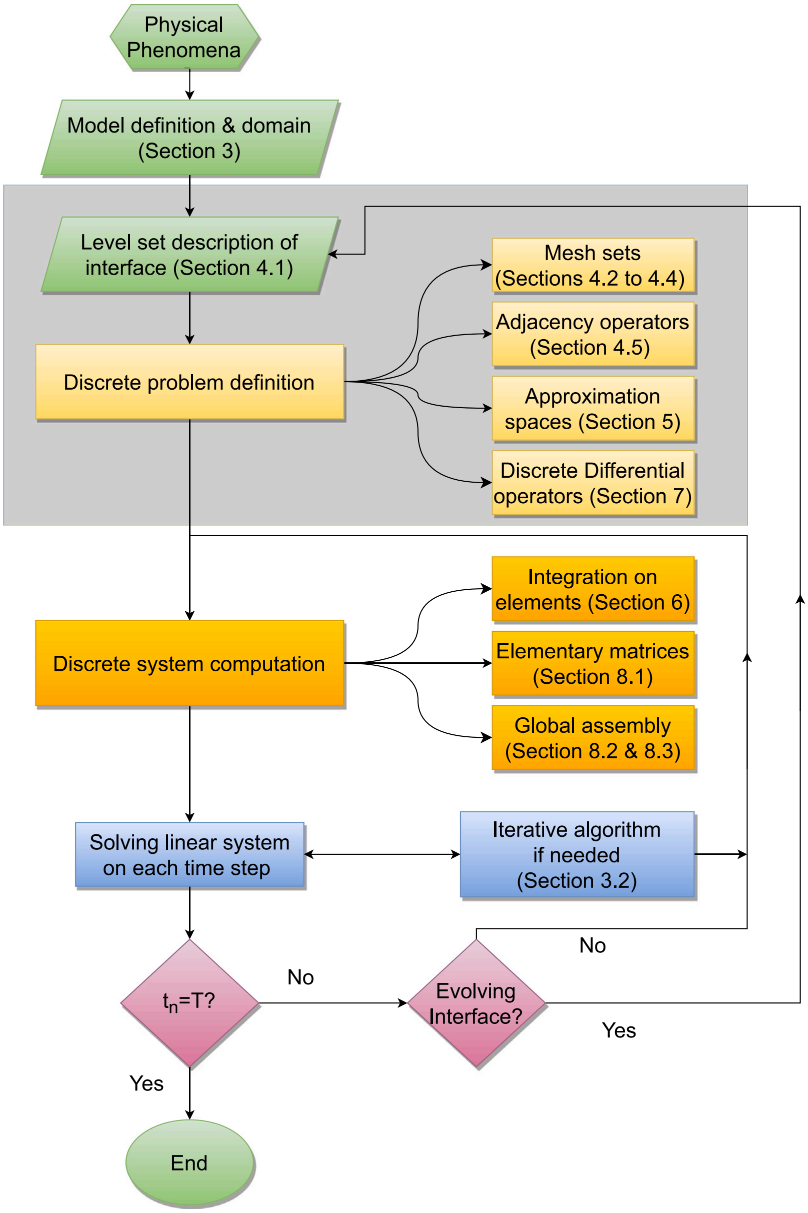
Physical Phenomena
Model definition & domain (Section 3)
Level set description of interface (Section 4.1)
Discrete problem definition
Mesh sets (Sections 4.2 to 4.4)
Adjacency operators (Section 4.5)
Approximation spaces (Section 5)
Discrete Differential operators (Section 7)
Discrete system computation
Integration on elements (Section 6)
Elementary matrices (Section 8.1)
Global assembly (Section 8.2 & 8.3)
Solving linear system on each time step
Iterative algorithm if needed (Section 3.2)
t_n=T?
No
Evolving Interface?
No
Yes
Yes
End

Fig. 1. Flow chart of a general XFEM implementation.

2. Implementation overview

In this section we will give an overview of the main steps that one needs to consider when creating an XFEM code to solve a mathematical model. This rough description will be detailed on the next sections, especially those that differ most from a standard FEM implementation. To illustrate the process, a flow chart of the proposed implementation procedure is presented in Fig. 1 where each box gives the information on which section will describe the corresponding step. The different stages are grouped by colour as follows:

- Green steps are related to the modelling, and are preliminary to the numerical simulation. Defining the physical problem, mathematical model, variational formulation, domain on which it is posed, and interface description by level sets are the first steps to consider.
- In yellow, the discretization steps are represented. These correspond to the determination of the approximation spaces, FEM approximation, type and number of enrichment functions needed, mesh of the domain, adjacency operators, and discrete form of the differential operators on the variational formulation.

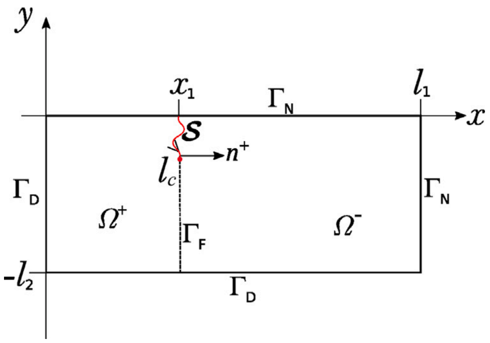

Fig. 2. Domain decomposition given by the extended interface.

- Then, the steps related to the computation of the system matrices are represented in orange. This includes integration on elements (main focus on integration on enriched elements and on the interface), elementary matrices computation and assembly.
- Then, depending on the kind of problem, we might need to iterate in time, or even use an iterative algorithm to solve a nonlinear system on each time step. This being the case of our practical use case, there is a subsection devoted to such algorithm.
- Finally, a decision is made whether the interface evolves or not. That being the case, the mesh sets and adjacency operators might need to be updated, together with the enriched parts of the discrete matrices. The criteria for the interface evolution will depend on the kind of problem (growing crack(s), moving bubble, moving boundary for phase change problems, etc.) so we will not enter into details here. The relevant part of this decision is that, depending on the result, one goes back to the interface description by level sets and redefine all the necessary discrete elements, like the adjacency operators (Section 4) and the enriched part of the elementary matrices (Section 8).

Under the proposed methodology, if the interface does not evolve, the steps over a grey background will not be recomputed. And even if the interface evolves, there is no need to recompute everything, but only those parts related to the enriched nodes, which are only a few compared with the total number of nodes on a typical finite elements mesh.

3. Mathematical model

As a thread of the implementation methodology presented in this document, and just as a practical use case, we will consider the mathematical model associated with the behaviour of a Rayleigh wave on a plate with a crack. This model was studied and solved with standard FEM in [28] and was chosen to be the thread in this work because it involves mass and stiffness matrices, external and body forces and, above all, interface terms due to contact conditions between the crack lips as well as a singularity on the crack tip. Therefore, it serves well to the purpose of presenting the methodology on a framework that can be easily transferred to other problems involving some kind of discontinuity in the interface, and need the calculation of integrals on it, or that present singularities in some area of their domain. Since we are only interested in this model as a use case that provides us with a discrete variational formulation general enough to facilitate the understanding of the main challenges arising on XFEM implementation, we will omit the details of the obtention of such formulation and the algorithm used to solve it, since these details can be consulted in [28] and they do not constitute the aim of this work.

We consider a three dimensional plate that occupies the region $(0,l_1)\times(-l_2,0)\times(-l_3,l_3)$ of the space XYZ, being l_i, $1\le i\le 3$ positive numbers. We assume that there exists a crack near the surface $y\equiv 0$ that is independent of z and that there is no initial gap between its lips. In the following we assume that the applied forces and boundary conditions are compatible with a plane strain hypothesis which allows us to solve the problem on the vertical middle surface. Therefore, all the variables only depend on the spatial variables x, y, and the time variable, t. The plate middle section corresponds to $\Omega=(0,l_1)\times(-l_2,0)$. Its boundary, Γ, splits into two disjoint parts: $\Gamma_D=[x\equiv 0]\cup[y\equiv -l_2]$ where the displacements are imposed and $\Gamma_N=[y\equiv 0]\cup[x\equiv l_1]$, where the body is free, or traction forces are applied.

Let $[0,T]$ be the time interval of interest. Let $\boldsymbol{u}(x,y,t)$ denote the displacement field and u_α its components, $\alpha\in\{1,2\}$. The components of the linearized strain tensor, $\varepsilon_{\alpha\beta}(\boldsymbol{u})$, are related to the stress tensor, $\sigma_{\alpha\beta}(\boldsymbol{u})$, through Hooke's constitutive law for elastic materials.

The crack restricted to Ω will be denoted by $\mathcal{S}$. It is assumed to be an oriented line such that given $s\in[0,l_c]$, with l_c a positive number, $l_c\ll l_2$, then $(x(s),-s)$ is a point on the crack, and $(x(l_c),-l_c)$ is the crack tip. To obtain later the variational formulation, it is convenient to split the domain Ω into two subdomains, by extending $\mathcal{S}$ with enough regularity, as can be seen in Fig. 2, in such a way that $\mathcal{S}$ is part of their boundary. For that, we consider a prolongation of $\mathcal{S}$ in the direction of the tangent vector to $\mathcal{S}$ at the crack tip. It will be denoted as Γ_F. We denote by Ω^+ (resp. Ω^-) the domain whose points lay to the right (resp. left) of the curve $\mathcal{S}$, when walking along it in the direction of the positive orientation of the crack. With this decomposition the fictitious boundary is $\Gamma_F = (\partial\Omega^+ \cap \partial\Omega^- \cap \Omega) \setminus \mathcal{S}$, and natural transmission interface conditions are assumed on it. Let us emphasize that Γ_F is only needed for the definition of the approximation spaces and the rigorous obtention of the discrete variational formulation, but it plays no role on the numerical resolution of the problem. Note that in other use cases, such as models with phase changes like gas bubbles in fluids, the definition of the $\Omega^\pm$ subdomains may not require the incorporation of fictitious boundaries. Let us also denote by $\boldsymbol{u}^+$ (resp. $\boldsymbol{u}^-$) the vector field associated to the displacements restricted to Ω^+ (resp. Ω^-), which, in the following, for the sake of brevity we will denote by $\boldsymbol{u}^{+(-)}$. We apply the same criteria to $\boldsymbol{\sigma}^{+(-)}$. Finally, the vector $\boldsymbol{n}^{+(-)}$ denotes the unit outward normal vector to $\Omega^{+(-)}$. If there is no room for confusion, we denote $\boldsymbol{n}^+$ as simply $\boldsymbol{n}$. Under plane strain assumptions, the behaviour of the plate is governed by:

$$\rho_0\ddot{\boldsymbol{u}} - \mathrm{div}\boldsymbol{\sigma} = \boldsymbol{f} \quad \text{in } \Omega^+ \cup \Omega^-, \tag{1}$$

$$\boldsymbol{\sigma n} = \boldsymbol{g} \quad \text{on } \Gamma_N, \tag{2}$$

$$\boldsymbol{u} = \boldsymbol{u}^D \quad \text{on } \Gamma_D, \tag{3}$$

$$[\sigma_n] = 0;\ \sigma_n \le 0;\ \boldsymbol{\sigma}_T = \boldsymbol{0} \quad \text{on } \mathcal{S}, \tag{4}$$

$$\sigma_n[u_n] = 0;\ [u_n] \le 0 \quad \text{on } \mathcal{S}, \tag{5}$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}) \quad \text{in } \Omega^+ \cup \Omega^-, \tag{6}$$

$$\dot{\boldsymbol{u}}(\boldsymbol{x}, 0) = \boldsymbol{v}_0(\boldsymbol{x}) \quad \text{in } \Omega^+ \cup \Omega^-, \tag{7}$$

$$[\boldsymbol{u}] = \boldsymbol{0};\ [\boldsymbol{\sigma n}] = \boldsymbol{0} \quad \text{on } \Gamma_F, \tag{8}$$

with $\boldsymbol{x} = (x, y)$, and being $[\boldsymbol{u}] = \boldsymbol{u}^+ - \boldsymbol{u}^-$, $[\boldsymbol{\sigma n}] = \boldsymbol{\sigma}^+\boldsymbol{n}^+ + \boldsymbol{\sigma}^-\boldsymbol{n}^-$, and $[u_n]$ the jump of the normal component of $\boldsymbol{u}$, this is, $[u_n] = \boldsymbol{u}^+ \cdot \boldsymbol{n}^+ + \boldsymbol{u}^- \cdot \boldsymbol{n}^-$. Also $\sigma_n = \boldsymbol{\sigma n} \cdot \boldsymbol{n}$ and $\boldsymbol{\sigma}_T = \boldsymbol{\sigma n} - \sigma_n\boldsymbol{n}$. For the boundary Γ_N to be free, we take $\boldsymbol{g} = \boldsymbol{0}$ without loss of generality.

3.1. Discrete variational formulation

In order to numerically solve problem (1)–(8), we follow the work in [28] where the interested reader can find the suitable variational framework, the treatment of the contact condition as well as an algorithm that combines a penalization method with a generalized Newton method to solve a fixed point problem using finite elements compatible with the crack position. In this work, for the sake of brevity, we only present the discrete variational equation that needs to be solved at each iteration j of the algorithm, which results from the application of the previous techniques, so we can focus on the implementation of the different integral terms when XFEM is used.

Let us consider a discretization of the domain Ω by n_{he} elements. For the sake of simplicity, we consider that the elements are triangles and the standard shape functions are piecewise linear. The extension to other types of elements of shape functions is not difficult. Let

$$\mathcal{T}_h = \{T^k : k = 1, \dots, n_{he}\}, \tag{9}$$

be the set of all elements in the mesh and

$$\mathcal{P}_h = \{\boldsymbol{x}_I : I = 1, \dots, n_{hn}\},$$

the set of all nodes of $\mathcal{T}_h$, being n_{hn} its cardinality. In the following, if there is no confusion, we will refer to the nodes $\boldsymbol{x}_I$ just by their index I.

The mesh $\mathcal{T}_h$ induces a discretization of the interface, $\mathcal{S}$, obtained by linearly approximating the interface on each element crossed by it. So, $\mathcal{S}$ is approximated by a polygonal line, $\mathcal{S}_h$,

$$\mathcal{S}_h = \{e_i : i = 1, \dots, n_{hf}\},$$

being n_{hf} its cardinality. In Fig. 3 the actual interface and its linearization are shown.

We will see in Section 4.1 how this linearization of the interface is obtained.

To get the discrete variational formulation of (1)–(8) at each time step, we consider the following spaces:

$$W_h = \{\boldsymbol{v}_h = (\boldsymbol{v}_h^+, \boldsymbol{v}_h^-) \in [C^0(\bar{\Omega}^+)]^2 \times [C^0(\bar{\Omega}^-)]^2\}$$

$$V_h = \{\boldsymbol{v}_h \in W_h / \boldsymbol{v}_{h|T^{k\pm}} \in [P_1(T^{k\pm})]^2, \forall T^k \in \mathcal{T}_h, \boldsymbol{v}_h = \boldsymbol{0} \quad \text{on} \quad \Gamma_D,\ [\boldsymbol{v}_h] = \boldsymbol{0} \text{ on } \Gamma_F\},$$

where $T^{k+} = T^k \cap \Omega^+$ and $T^{k-} = T^k \cap \Omega^-$, and

$$P_h = \{q_h \in L^\infty(\mathcal{S}_h); q_{h|e_i} \in P_0(e_i),\ 1 \le i \le n_{hf}\}.$$

Notice that this definition of W_h and V_h allows for a discontinuity of $\boldsymbol{v}_h$ along $\mathcal{S} \cup \Gamma_F$.

We also denote by $\mathcal{S}_j^c$ the subset of edges in $\mathcal{S}_h$ where effective contact happens at iteration j of the iterative algorithm solving the nonlinearity.

Now, let $B : V_h \to P_h$ be the operator defined by

$$B(\boldsymbol{v}_h)_{|e_i} = [v_{hn}]_{|e_i} \quad \text{for all edge } e_i \text{ in } \mathcal{S}_h. \tag{10}$$

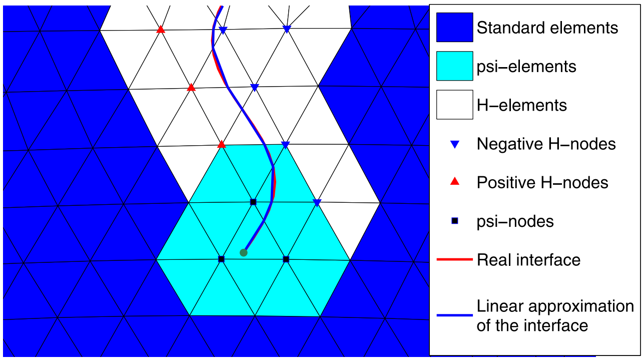

Fig. 3. Near the crack mesh and linearized interface.

Then, we consider a penalty-duality algorithm introduced in [29], which is a generalization of the Uzawa's method for the augmented Lagrangian, combined with a generalized Newton's method to accelerate the rate of convergence. The details of the algorithm can be consulted by the interested reader in, for example, [28,30,31] where the increased rate of convergence is shown. This algorithm is robust, and no oscillations or loss of convergence have been observed, achieving convergence very fast in less than ten iterations on each time step.

The discrete variational formulation can be obtained by following the same procedure as in [28] where the mesh was assumed to be compatible with the crack, this is, $T^k = T^{k+}$ or $T^k = T^{k-}$. Therefore, at each time step, the system to be solved at each iteration, j, comes from the following equation:

$$\int_{\Omega} \rho_0 \ddot{\boldsymbol{u}}_h^j \cdot \boldsymbol{v}_h \, d\boldsymbol{x} + \int_{\Omega} \boldsymbol{\sigma}_h^j : \boldsymbol{\varepsilon}_h(\boldsymbol{v}_h) \, d\boldsymbol{x} + \frac{1}{\epsilon} \int_{\mathcal{S}_{j-1}^c} B(\boldsymbol{u}_h^j) B(\boldsymbol{v}_h) \, d\Gamma = \int_{\Omega} \boldsymbol{f}_h \cdot \boldsymbol{v}_h \, d\boldsymbol{x} + \int_{\Gamma_N} \boldsymbol{g}_h \cdot \boldsymbol{v}_h \, d\Gamma, \tag{11}$$

for all $\boldsymbol{v}_h \in V_h$, where ϵ is a small positive parameter, $\boldsymbol{\varepsilon}_h(\boldsymbol{v}_h)$ is the linearized strain tensor associated to $\boldsymbol{v}_{h|T^k}$ related to $\boldsymbol{\sigma}_h(\boldsymbol{v}_h)$ through Hooke's law and $\mathcal{S}_{j-1}^c$ is the subset of edges in $\mathcal{S}_h$ where effective contact happens at iteration $j-1$. It is in this set, $\mathcal{S}_{j-1}^c$, where the jump of the displacement needs to be zero, so this will constitute the stopping criteria of the algorithm (along with a maximum number of iterations of 20 which is never reached). The algorithm will stop when $B(\boldsymbol{u}_h^j) < 1.e - 10$ on $\mathcal{S}_{j-1}^c$.

Remark 1. Notice that this variational formulation involves a term that requires integration on the interface, which is the main aim of this work. Furthermore, it also includes the usual terms appearing on the variational formulation of a wide range of problems, such as the acceleration term (leading to a mass matrix), internal stress (stiffness matrix) and loading terms. This is the reason why this model has been chosen as the framework to present the proposed methodology, because it can be easily transferred to any other problem involving this kind of terms or similar ones.

This variational framework takes into account the discontinuity across the interface, but to capture the solution singularity near the crack tip with this approximation we would need a very refined mesh. This will be avoided by adding to the discrete space new shape functions that present the same kind of singularity. The resulting enriched discrete space will not be V_h, but a new V_h^X, and the variational formulation in (11) will also hold. We deal with it in Section 5.

Another advantage of this enriched approximation is that the mesh does not need to be compatible with the crack, so in the case of the crack growing, there is no need for re-meshing, saving the additional computational cost that it would mean on each time step.

4. Sets of mesh nodes and elements. Adjacency

In this section we introduce the mathematical elements that will vertebrate a general XFEM implementation procedure and the relations between them.

As the model to be solved might have different singularities, the kind and number of enrichments might vary from one problem to another. Throughout this paper, we will set ourselves on the case that two different kinds are needed, being this the case for our example. The generalization to other problems with more (or less) enrichments needed would be easy to perform following the same methodology and operators presented here. To this end, Fig. 6 plays a key role in order to define the necessary embeddings and discard the negligible ones. Our practical use case presents a type I discontinuity on the displacement and a type II discontinuity on the stress, so we consider two kind of enrichment functions: H is a discontinuous function through $\mathcal{S}_h$, and a set, Ψ, of four functions Ψ_l, $l = 1, \dots, 4$, which reproduce the singular behaviour of the stress near the crack tip.

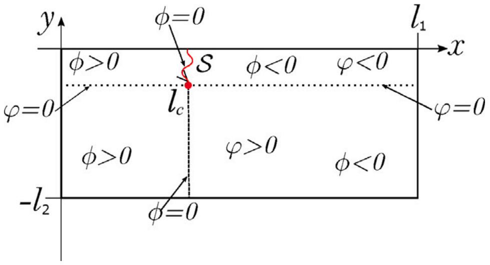

Fig. 4. Example case: representation of the crack through level sets.

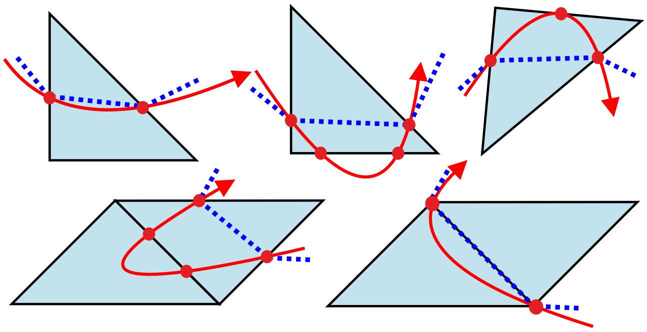

Fig. 5. Linear approximation of the interface within each element.

4.1. Level sets representation

Let us assume that the interface is given by level sets. In the model example of this paper, being the interface a crack near the surface of the plate, only two types of level set functions (see Fig. 4) are needed to describe it. Let us consider ϕ, defined in Ω_h, and φ, defined in $(-l_2, 0)$, such that:

$$\begin{cases} \phi(\boldsymbol{x}) > 0, & \text{if } \boldsymbol{x} \in \Omega_h^+, \qquad \varphi(y) < 0, \quad \text{if } y \in (-l_c, 0], \\ \phi(\boldsymbol{x}) = 0, & \text{if } \boldsymbol{x} \in \mathcal{S}_h \cup \Gamma_F, \qquad \varphi(y) = 0, \quad \text{if } y = -l_c, \\ \phi(\boldsymbol{x}) < 0, & \text{if } \boldsymbol{x} \in \Omega_h^-, \qquad \varphi(y) > 0, \quad \text{if } y \in (-l_2, -l_c). \end{cases}$$

The zero level set of ϕ corresponds to the interface when $\varphi < 0$ too. Besides, the point $\boldsymbol{x} \in \Omega_h$ verifying $\phi(\boldsymbol{x}) = 0$ and $\varphi(\boldsymbol{x}) = 0$ represents the interface ending point.

The discontinuous enrichment function H is defined as

$$H(\boldsymbol{x}) = \tilde{H}(\phi(\boldsymbol{x})), \tag{12}$$

being $\tilde{H}(\varsigma)$ the Heaviside-like function

$$\tilde{H}(\varsigma) = \begin{cases} 1, & \text{if } \varsigma \geq 0, \\ -1, & \text{if } \varsigma < 0. \end{cases}$$

Recall that the element edges might not be aligned with the interface and let us denote by $\{N_I : I \in \mathcal{P}_h\}$ the set of the standard shape functions associated to the nodes of the partition. In practice, the values of the level set functions are computed only on the nodes, so $\phi_I = \phi(\boldsymbol{x}_I)$, $\varphi_I = \varphi(\boldsymbol{x}_I)$, with $I = 1, \ldots, n_{hn}$. Therefore, new level-set functions, still denoted by ϕ and φ, can be defined as the approximation of ϕ and φ using the standard shape functions $N_I(\boldsymbol{x})$. So, at any point $\boldsymbol{x} \in \Omega_h$, this new level set functions are given by $\phi(\boldsymbol{x}) = \sum_I N_I(\boldsymbol{x})\phi_I$ (respect. with φ). In this way, the interface is approximated by a straight segment on each element as shown in Fig. 5. Consequently, the finer the mesh the more accurate the approximation of the interface. More details on the interface approximation can be found in [27].

4.2. Node sets

We classify the mesh nodes regarding the kind of shape functions associated to each one (see Fig. 3):

- $\mathcal{P}_\psi$ the subset of $\mathcal{P}_h$ that contains all the nodes enriched with the singular functions Ψ. We will refer to these nodes as *psi-nodes*. The total number of nodes in $\mathcal{P}_\psi$ is denoted by $n_{\psi n}$,

- $\mathcal{P}_H$ the subset of $\mathcal{P}_h$ that contains all the nodes enriched with the Heaviside type functions H. We will refer to these nodes as *H-nodes*. The total number of nodes in $\mathcal{P}_H$ is denoted by n_{Hn}.
- $\mathcal{P}_S$ the subset of $\mathcal{P}_h$ that contains all non-enriched nodes. We will refer to these nodes as *standard nodes*. The total number of nodes in $\mathcal{P}_S$ is denoted by n_{Sn}.

Therefore, $\mathcal{P}_h$ is the union of the three disjoint subsets $\mathcal{P}_S$, $\mathcal{P}_H$ and $\mathcal{P}_\psi$ and $n_{hn} = n_{\psi n}+n_{Hn}+n_{Sn}$. When referring to the numbering of a node with respect to any of the previous sets $\mathcal{P}_L$, $L \in \{h, H, \psi, S\}$, the notation I_L will be used. As we said before, from now on, $\mathcal{P}_L$ will also denote the sets of local indices corresponding to the nodes of the set $\mathcal{P}_L$.

Let us consider the embeddings $n_L : \mathcal{P}_L \hookrightarrow \mathcal{P}_h$, that correlate the nodes local numbering on $\mathcal{P}_L$, $L \in \{S, H, \psi\}$ with their respective global one on $\mathcal{P}_h$. We also consider the mappings n_L^{-1}, $L \in \{S, H, \psi\}$ defined by

$$n_L^{-1}(I_h) = \begin{cases} I_L & \text{if } I_h = n_L(I_L), \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

It is clear that this mapping is a left inverse of n_L, since

$$n_L^{-1}(n_L(I_L)) = I_L, \quad \forall I_L \in \{1, \dots, n_{Ln}\},\ L \in \{S, H, \psi\}.$$

Note that a fictitious node must be added to the sets $\mathcal{P}_L$, $L \in \{S, H, \psi\}$, so that n_L^{-1} is well defined. Such node will be labelled by 0_L.

The relations between the different sets and embeddings are graphically shown in the diagram shown in Fig. 6.

4.3. Element sets

Taking into account the specific characteristics of the nodes considered, a classification of the elements of the mesh in three disjoint sets is obtained (see Fig. 3):

- $\mathcal{T}_\psi$ contains all elements with at least one node belonging to $\mathcal{P}_\psi$. We will refer to these elements as *psi-elements*. The total number of elements in $\mathcal{T}_\psi$ is denoted by $n_{\psi e}$,
- $\mathcal{T}_H$ contains all elements with at least one node belonging to $\mathcal{P}_H$ and none belonging to $\mathcal{P}_\psi$. We will refer to these elements as *H-elements*. The total number of elements in $\mathcal{T}_H$ is denoted by n_{He}.
- $\mathcal{T}_S$ contains all elements with only standard nodes, this is, without any nodes belonging to $\mathcal{P}_\psi \cup \mathcal{P}_H$. We will refer to these elements as *standard elements*. The total number of elements in $\mathcal{T}_S$ is denoted by $n_{Se} = n_{he} - (n_{\psi e} + n_{He})$.

Therefore, $\mathcal{T}_h$ is the union of the three disjoint subsets $\mathcal{T}_S$, $\mathcal{T}_H$ and $\mathcal{T}_\psi$.

Similarly to what we did with the node sets, we consider the inclusions $T_L : \mathcal{T}_L \hookrightarrow \mathcal{T}_h$, correlating the element sets for $L \in \{S,\ H,\ \psi\}$.

We can see in the diagram in Fig. 6 how the applications and sets defined above relate to each other.

4.4. Edge sets

The previous classification of elements induces a similar classification of edges in the polygonal interface, $\mathcal{S}_h$, that approximates the real one depending on which element the edge belongs to. Indeed, we consider the subsets $\mathcal{S}_H$ and $\mathcal{S}_\psi$ that contain the edges of $\mathcal{S}_h$ that cross elements in $\mathcal{T}_H$ or $\mathcal{T}_\psi$, respectively. Because of their definition $\mathcal{S}_H$ and $\mathcal{S}_\psi$ are disjoint sets such that $\mathcal{S}_h = \mathcal{S}_H \cup \mathcal{S}_\psi$. Also notice that, due to the linear approximation of the interface, each edge can only belong to one element in $\mathcal{T}_h$. Similarly to what we did for nodes and elements, in order to integrate on the interface $\mathcal{S}_h$, we need to relate the interface edges with the elements they cross. This is why the applications D, D_{HH} and $D_{\psi\psi}$, are defined, assigning to each interface edge the corresponding mesh element, according to the diagram in Fig. 6.

4.5. Adjacency operators

The final step to describe a mesh oriented to the XFEM implementation is to establish the relations between edges, nodes and elements. The adjacency element-node is defined by operators, $C_{LM} : \mathcal{T}_L \to \mathcal{P}_M^3$, $L,\ M \in \{h, S, H, \psi\}$ giving for each element in $\mathcal{T}_L$ the set of nodes belonging to this element with their numbering on the set $\mathcal{P}_M$. After fixing indexation on $\mathcal{T}_L$ and $\mathcal{P}_M$, the adjacency operator can be represented by a matrix, denoted by $[C_{LM}] \in \mathcal{M}_{3\times n_{Le}}$, being $[C_{LM}]_{ik}$ the local numbering on $\mathcal{P}_M$ of the ith node of the element $T^k \in \mathcal{T}_L$, $1 \le k \le n_{Le}$, $1 \le i \le 3$, or 0 if the node does not belong to $\mathcal{P}_M$.

The commutative diagram on Fig. 6 shows how the applications and sets defined above relate to each other.

If we denote by $\boldsymbol{n}_L^{-1}$, $L \in \{S, H, \psi\}$ the vector-valued function that acts like $n_L^{-1} : \mathcal{P}_h \to \mathcal{P}_L$ defined in (13) on each component, we can define the previous adjacency operators by

$$C_{LM} = \boldsymbol{n}_M^{-1} C_{hh} T_L, \quad L, M \in \{S, H, \psi\}. \tag{14}$$

These operators will be used to build the local-to-global allocation arrays that are essential to the mass and stiffness matrices assembling. Even though there are more possibilities, only five of them are relevant to the implementation and are shown in the diagram in Fig. 6. Note that all the information relative to standard degrees of freedom that might be needed for the assembling is provided by both $[C_{hh}]$ and $[C_{SS}]$ while the information related to *psi-nodes* and *H-nodes* is provided by $[C_{HH}]$, $[C_{\psi H}]$ and $[C_{\psi\psi}]$.

$$\begin{tikzcd}
 & \mathcal{P}^3_\psi & & \mathcal{P}^3_S \\
\mathcal{P}^3_H & & \mathcal{P}^3_h \arrow[ll, "n_H^{-1}"'] \arrow[ul, "n_\psi^{-1}"'] \arrow[ur, "n_S^{-1}"] & \\
 & \mathcal{T}_\psi \arrow[u, dotted, "C_{\psi\psi}"] \arrow[ul, dotted, "C_{\psi H}"] \arrow[dr, "T_\psi"] & & \mathcal{T}_S \arrow[uu, "C_{SS}"'] \arrow[dl, "T_S"] \\
\mathcal{T}_H \arrow[uu, "C_{HH}"] \arrow[rr, "T_H"] & & \mathcal{T}_h \arrow[uu, "C_{hh}"'] & \\
 & \mathcal{S}_\psi \arrow[uu, dotted, "D_{\psi\psi}"'] \arrow[dr, "S_{\psi\psi}"] & & \\
\mathcal{S}_H \arrow[uu, "D_{HH}"] \arrow[rr, "S_{HH}"'] & & \mathcal{S}_h \arrow[uu, "D"'] &
\end{tikzcd}$$

Fig. 6. Relevant adjacency operators.

Remark 2. Note that although the sets of elements $\mathcal{T}_S$, $\mathcal{T}_H$ and $\mathcal{T}_\psi$ are disjoint, not all the nodes on one element need to be on the same nodes set. An element on $\mathcal{T}_\psi$ might have some nodes on $\mathcal{P}_\psi$ and the rest on $\mathcal{P}_H$ or $\mathcal{P}_S$ (see Fig. 3).

Remark 3. Notice that if the interface were to evolve, we will not need remeshing, but updating these sets and embeddings. One node that was previously enriched might not be enriched on the next time step and vice versa. A careful definition of the adjacency operators and the local-to-global allocation mappings is essential for a flawless implementation.

5. XFEM approximation

The key point of the extended finite element method is to add to the standard FEM basis some particular shape functions verifying the basic concept of partition of unit and displaying specific properties of the expected solution (see [1,2] or [3] for instance). As stated before, the standard shape functions considered here are piecewise linear (Lagrange P^1). Besides, for representing a crack near the surface we need two types of enrichment functions: the first one that accounts for the possible discontinuity of the displacement when crossing the interface, H, and the second type that is well adapted to the expected singularity in the stresses near the tip of the crack; in this example a set of four functions Ψ_l, $l = 1, \dots, 4$, affecting the nodes on $\mathcal{P}_\psi$, are considered (see [7] or [32]). Therefore, the approximation of a function $\boldsymbol{u} : \boldsymbol{x} \in \Omega \to \boldsymbol{u}(\boldsymbol{x}) \in \mathbb{R}^2$, is considered as

$$\boldsymbol{u}_h(\boldsymbol{x}) = \sum_{I_h=1}^{n_{hn}} N_{I_h}(\boldsymbol{x})\boldsymbol{u}_{I_h} + \sum_{I_H=1}^{n_{Hn}} N_{n_H(I_H)}(\boldsymbol{x})H(\boldsymbol{x})\boldsymbol{c}_{I_H} + \sum_{I_\psi=1}^{n_{\psi n}} N_{n_\psi(I_\psi)}(\boldsymbol{x}) \sum_{l=1}^{4} \Psi_l(\boldsymbol{x})\boldsymbol{b}^l_{I_\psi}, \tag{15}$$

where $\boldsymbol{u}_{I_h}, \boldsymbol{c}_{I_H}, \boldsymbol{b}^l_{I_\psi} \in \mathbb{R}^2$.

In [27] the interested reader can find the graphical representation of the one- and two-dimensional standard and Heaviside shape functions. We remark that this choice for H is not unique and other options can be found in [3]. Here we are dealing with two different types of enrichment, having some nodes enriched with only one function (H), some nodes enriched with 4 functions (Ψ_l, in this case $1 \le l \le 4$) and of course, most nodes not enriched at all. The procedure can be easily generalized to more (or less) enrichment types and any other number of enrichment functions, depending on the expected characteristics of the solution of each particular problem.

Regarding the enrichment functions near the crack tip, Ψ_l are given by (see [33])

$$\Psi_1(\boldsymbol{x}) = \sqrt{r}\sin\frac{\theta}{2}; \qquad \Psi_2(\boldsymbol{x}) = \sqrt{r}\cos\frac{\theta}{2}; \qquad \Psi_3(\boldsymbol{x}) = \sqrt{r}\sin\frac{\theta}{2}\sin\theta; \qquad \Psi_4(\boldsymbol{x}) = \sqrt{r}\cos\frac{\theta}{2}\sin\theta, \tag{16}$$

where $(r,\theta) \in (0,+\infty) \times [-\pi,\pi]$ are the polar coordinates of point $\boldsymbol{x} = (x,y)$ with origin on the crack tip, and polar axis tangent to the crack at its tip so that the crack lips correspond to $\theta = \pm\pi$.

Remark 4. Notice that the limit of $\Psi_1(\boldsymbol{x}) = \sqrt{r(\boldsymbol{x})}\sin\frac{\theta(\boldsymbol{x})}{2}$, when $\boldsymbol{x}$ approaches the crack from positive angles, i.e., $\theta \to \pi$, is $\sqrt{r(\boldsymbol{x})}\sin\frac{\pi}{2} = \sqrt{r(\boldsymbol{x})}$, while from negative angles ($\theta \to -\pi$) is $\sqrt{r(\boldsymbol{x})}\sin\frac{-\pi}{2} = -\sqrt{r(\boldsymbol{x})}$. This fact makes it necessary to keep the alignment of the segment $\theta = \pi$ with the crack edges belonging to $\mathcal{S}_h$ in those elements adjacent to the one containing the crack tip (whose nodes are enriched with the near-tip functions) by taking into account a change of reference system on each element or alternative near-tip functions. This procedures can be consulted in many works, see for example [14,34–36]. For the sake of simplicity, in this work we consider that the crack is straight near the tip, i.e. on the elements of $\mathcal{P}_\psi$. Notice that this is not a restriction since it could be achieved simply by refining the mesh near the tip.

The graphical representation of the functions (16) can be found in [27]. The compact support of these singular enrichment is guaranteed when these functions are multiplied by the standard shape functions.

Finally, with the shape functions introduced in (15), we can write the approximation space V_h as

$$V_h =< N_{I_h}(\boldsymbol{x}) >_{I_h\in\{1,\ldots,n_{hn}\}} \oplus < N_{n_H(I_H)}(\boldsymbol{x})H(\boldsymbol{x}) >_{I_H\in\{1,\ldots,n_{Hn}\}}, \tag{17}$$

and introduce a new space V_h^X as

$$V_h^X = < N_{I_h}(\boldsymbol{x}) >_{I_h\in\{1,\ldots,n_{hn}\}} \oplus < N_{n_H(I_H)}(\boldsymbol{x})H(\boldsymbol{x}) >_{I_H\in\{1,\ldots,n_{Hn}\}} \oplus \cup_{l=1}^{4} \left(< N_{n_\psi(I_\psi)}(\boldsymbol{x})\Psi_l(\boldsymbol{x}) >_{I_\psi\in\{1,\ldots,n_{\psi n}\}} \right). \tag{18}$$

6. Integration on enriched elements

For numerical integration on elements crossed by the interface, the use of classical quadrature formulas is not recommended. To avoid loss of precision, it is recommended that these elements be partitioned beforehand so that the functions to be integrated are continuous for each of them. Such partitioning is only considered for the purpose of achieving a good numerical integration, and it does not affect the number of degrees of freedom of the discrete problem.

There are several works that consider strategies for integration over cut elements. For example in [37] the authors use a Schwarz–Christoffel conformal mapping to integrate without splitting into subelements, in [38] the authors propose a quadrature scheme for arbitrarily shaped convex or concave volumes, or in [39] where they use DG mortaring to deal with nonconforming overlapping meshes. Other authors introduce new transformation methods for the numerical integration of singular functions e.g. [40,41] for 2D and 3D respectively. In this section, just for the sake of completeness, we will briefly recall some results from [27] that allow us to integrate over enriched elements regardless of the quadrature formula used on the non-enriched elements, maintain the same order on the enriched ones, and it is easily adaptable into the automatic processing proposed in the rest of the paper.

6.1. Integration of discontinuous functions

Let $T^k \in \mathcal{T}_h$ be an element crossed by the interface, $\mathcal{S}_h$. Let $\Xi^k = \{T^e : e = 1,\ldots,n^k\}$ a partition of T^k verifying that $\mathcal{S}_h \cap T^k \subset \cup_{e=1}^{n^k} \partial T^e$. Note that with this condition, T^{k+} and T^{k-} defined in Section 3.1, can be expressed as the union of some of these subelements and each subelement belongs either to Ω^+ or to Ω^-. Besides, this condition ensures that the interface on T^k can be described as union of edges of some subelements. For each $e = 1,\ldots,n^k$, we define an affine map $\mathcal{N}^e : T^k \to T^e$ by assigning to each point $\boldsymbol{x} \in T^k$, a point $\boldsymbol{x}' \in T^e$ having the same barycentric coordinates with respect to T^e than $\boldsymbol{x}$ with respect to T^k. We will denote by J^e the Jacobian of the affine map $\mathcal{N}^e$. Besides, we define the partition matrix $[\mathcal{N}^e]$, of $T^e \in \Xi^k$ with respect to T^k as the matrix whose components are the barycentric coordinates of the vertices of T^e with respect to the shape functions $N_{V_i^k}$ associated to the nodes of T^k. Notice that this matrix is the one associated to $\mathcal{N}^e$. A procedure to build these partition matrices in a automatic way, independently of the elements casuistic, was detailed in [27] both for the two and three-dimensional cases. Let V_i^k be the vertices of T^k and V_i^e, $1 \le e \le n^k$ the vertices of subelement T^e for $1 \le i \le 3$. We denote by $N_{V_i^k}$ and $N_{V_i^e}$, the standard shape functions associated to the elements T^k and T^e respectively. Given a point $\boldsymbol{x} \in T^e \subset T^k$, let us denote by

$$\left[N_S^k\right](\boldsymbol{x}) = \left[N_{V_1^k}(\boldsymbol{x}),\ N_{V_2^k}(\boldsymbol{x}),\ N_{V_3^k}(\boldsymbol{x})\right], \tag{19}$$

the matrix belonging to $\mathcal{M}_{1\times 3}$ whose components are the standard shape functions associated to the vertices of T^k, $N_{V_i^k}(\boldsymbol{x})$, and $[N_S^e](\boldsymbol{x})$ for the one with components $N_{V_i^e}(\boldsymbol{x})$, $1 \le i \le 3$.

Proposition 1. *Given $f \in L^1(T^k)$, and a n_q-points integration rule*

$$Q_{n_q}(f) = \sum_{p=1}^{n_q} \omega_p f(\eta_p, \xi_p), \tag{20}$$

with $[\omega_p, (\eta_p, \xi_p)]_{p=1,\ldots,n_q}$ as weights and quadrature points on the reference element, the integral

$$\int_{T^k} f(\boldsymbol{x}) d\boldsymbol{x}$$

on any element T^k intersected by the interface with associated partition Ξ^k, can be approximated by

$$Q_{n_q}(f) = |J^k| \sum_{e=1}^{n^k} \sum_{p=1}^{n_q} \omega_p^e f(\eta_p^e, \xi_p^e), \tag{21}$$

where J^k is the Jacobian of the affine transformation from the reference element to T^k, the weights ω_p^e and the coordinates of the integration points (η_p^e, ξ_p^e), are computed as

$$\omega_p^e = |J^e|\omega_p, \quad P^e = [V^k][\mathcal{N}^e][N_{n_q}], \tag{22}$$

being $[N_{n_q}]$ *the matrix of barycentric coordinates of the integration points on the reference element* T*, this is*

$$[N_{n_q}] = [[N_S^k]^t(\eta_1, \xi_1)...[N_S^k]^t(\eta_{n_q}, \xi_{n_q})],$$

and $[\mathcal{N}^e]$ *the corresponding partition matrix.*

6.2. Integration of singular functions

In order to integrate over partially cut elements, one must take into account not only the discontinuity due to the interface but also the singularity near the crack tip. That is why special quadrature rules are also recommended to capture the singularity more accurately. A summary of the different techniques can be found in [40,41] or [3].

6.3. Integration on the interface

When the variational formulation of the problem of interest involves some integral on the interface, the integrands will typically be jumps on the interface coming from displacements, gradients, pressure, or any other field of interest. The line integral (for 2-D problems) poses not additional difficulties in terms of quadrature, nor does it require any partitioning (assuming that the interface has been linearly approximated on each element), so interface integration was not treated in [27]. However, the jump terms add special matrices to the matrix formulation of the problem, aside from the typical mass, damping or stiffness matrices, so the construction and assembling of this jump matrices will be detailed in Section 8.

7. Discrete differential operators

Now that we know how to integrate any function f over an element partially or totally cut by the interface, in this section we focus on how the XFEM formulation affects the discrete expression of variational formulation (11) in comparison with the standard FEM approximation. Since the domain (and boundary) integrals are computed as the sum of elementary (edges) ones, we will initially focus our attention on the expression of the main differential operators when restricted to an element; afterwards, in Section 8.2 we will propose coherent local-to-global mappings in order to achieve an efficient assembling avoiding a casuistic analysis about the number of nodes of each type belonging to the element. The procedure will be presented for the 2-D case, and it can be easily generalized to 3-D cases with minor changes, take for example that each standard node would have three associated degrees of freedom instead of two.

Given $\boldsymbol{u}_h \in V_h^X$, we denote by $\{u_h\}$ the column vector composed by all scalar coefficients in expression (15), so

$$\{u_h\} = \left\{ \begin{array}{c} \{\boldsymbol{u}_{I_h}\}_{I_h \in \{1,\ldots,n_{hn}\}} \\ \hline \{\boldsymbol{c}_{I_H}\}_{I_H \in \{1,\ldots,n_{Hn}\}} \\ \hline \{\boldsymbol{b}_{I_\psi}\}_{I_\psi \in \{1,\ldots,n_{\psi n}\}} \end{array} \right\}, \tag{23}$$

is a vector in $\mathbb{R}^{n_{gdf}}$ whose components are the global degrees of freedom of $\boldsymbol{u}_h$. In particular, $n_{gdf} = 2(n_{hn}+n_{Hn}+4n_{\psi n})$. Furthermore, the column vectors $\{\boldsymbol{u}_I\}$ and $\{\boldsymbol{c}_I\}$ belong both to $\mathbb{R}^2$ and represent the standard and H degrees of freedom respectively, while the column vector $\{\boldsymbol{b}_I\} \in \mathbb{R}^8$ comprises 4 vectors $\{\boldsymbol{b}_I^l\} \in \mathbb{R}^2$ organized as

$$\{\boldsymbol{b}_I\} = \left\{ \begin{array}{c} \{\boldsymbol{b}_I^1\} \\ \hline \vdots \\ \hline \{\boldsymbol{b}_I^4\} \end{array} \right\},$$

and represents the degrees of freedom associated to the Ψ enrichment at each ψ-node.

Since we will be working on the element level, given any element $T^k \in \mathcal{T}_h$, the first step is to extract from the global degrees of freedom vector only those associated to T^k. Actually, among the elements $T^k \in \mathcal{T}_h$ we can distinguish four cases, elements on $\mathcal{T}_S$, elements on $\mathcal{T}_H$, elements on $\mathcal{T}_\psi$ with some node on $\mathcal{P}_H$ and elements on $\mathcal{T}_\psi$ without nodes on $\mathcal{P}_H$. Even though unrealistic, for the simplicity of the exposition and to automate the process, here and in the following section we will consider that the nodes on T^k are enriched with all the enrichment functions. Therefore, let us denote by $n_{edf} = 36$ the number of degrees of freedom per element. If V_i^k represents the ith vertex of the kth element, $\{\boldsymbol{u}_{V_i^k}\} := \{\boldsymbol{u}_{[C_{hh}]_{ik}}\}$, $\{\boldsymbol{c}_{V_i^k}\} := \{\boldsymbol{c}_{n_H^{-1}([C_{hh}]_{ik})}\}$ and $\{\boldsymbol{b}_{V_i^k}\} := \{\boldsymbol{b}_{n_\psi^{-1}([C_{hh}]_{ik})}\}$, for $1 \le i \le 3$, will represent the standard, H and ψ element degrees of freedom associated to vertex V_i^k. As we said before, $N_{V_i^k}$ denotes the standard shape function associated to node V_i^k, this is, $N_{V_i^k} := N_{[C_{hh}]_{ik}}$. Note that $n_L^{-1}([C_{hh}]_{ik})$, $L \in \{H, \psi\}$ might be zero, so we will just assume that $\boldsymbol{c}_0 = \{\boldsymbol{0}\} \in \mathbb{R}^2$, and $\boldsymbol{b}_0 = \{\boldsymbol{0}\} \in \mathbb{R}^8$.

We consider the matrix $[A^k] \in \mathcal{M}_{n_{edf} \times n_{gdf}}$ such that

$$\{u_h^k\} = [A^k]\{u_h\}, \tag{24}$$

being $\{u_h^k\} \in \mathbb{R}^{n_{edf}}$ the vector whose components are the element degrees of freedom. This matrix $[A^k]$ is closely related to the assembling procedure that will be described on Section 8.2, so we will not enter into details here. The disposition of the local degrees of freedom on $\{u_h^k\}$ will be different to the global one, since they will be grouped by components as follows:

$$\{u_h^k\} = \left\{\begin{array}{c} \{u_1^k\} \\ \hline \{u_2^k\} \\ \hline \{c_1^k\} \\ \hline \{c_2^k\} \\ \hline \{b_1^k\} \\ \hline \{b_2^k\} \end{array}\right\}, \tag{25}$$

where $\{u_s^k\} \in \mathcal{M}_{3\times 1}$, with $1 \le s \le 2$ represents the standard degrees of freedom associated to the sth component of $\boldsymbol{u}_h$ on all the vertices, this is,

$$\{u_s^k\} = \left\{\begin{array}{c} \left(u^k_{V_1^k}\right)_s \\ \hline \left(u^k_{V_2^k}\right)_s \\ \hline \left(u^k_{V_3^k}\right)_s \end{array}\right\}, \tag{26}$$

and $\{c_s^k\}$ and $\{b_s^k\}$ are organized similarly.

Remark 5. Clearly, for many elements some vector $\{\boldsymbol{c}_{V_i^k}\}$, $\{\boldsymbol{b}_{V_i^k}\}$ or even both might be null as well as the respective rows on $[A^k]$, and the corresponding columns on the elementary matrices are unnecessary and will not be assembled. The choice of considering separate cases is left to the programmer.

From (15) one can easily deduce that the expression of $\boldsymbol{u}_h$ when restricted to T^k, which is noted by $\boldsymbol{u}_{h|T^k}$, is given by

$$\boldsymbol{u}_{h|T^k}(\boldsymbol{x}) = \sum_{i=1}^{3} N_{V_i^k}(\boldsymbol{x}) \left(\boldsymbol{u}_{V_i^k} + H(\boldsymbol{x})\boldsymbol{c}_{V_i^k} + \sum_{l=1}^{4} \Psi_l(\boldsymbol{x})\boldsymbol{b}^l_{V_i^k} \right), \boldsymbol{x} \in T^k. \tag{27}$$

Our aim is to find a matrix $[\hat{N}^k]$ such that

$$\boldsymbol{u}_{h|T^k}(\boldsymbol{x}) = [\hat{N}^k](\boldsymbol{x})\{u_h^k\}, \ \boldsymbol{x} \in T^k. \tag{28}$$

Let us recall the definition of $\left[N_S^k\right]$ given by (19) and let us denote by $\left[N_H^k\right] = H\left[N_S^k\right]$ and $[N_\psi^k] \in \mathcal{M}_{1\times 12}$ the matrix whose components are the standard shape functions multiplied by the Ψ_l functions, this is

$$[N_\psi^k] = \left[N_{V_1^k}[\Psi] | N_{V_2^k}[\Psi] | N_{V_3^k}[\Psi]\right],$$

with

$$[\Psi] = \left[\Psi_1, \ \Psi_2, \ \dots, \ \Psi_4\right].$$

Then considering the matrix $[\hat{N}^k]$ belonging to $\mathcal{M}_{2\times n_{edf}}$ and defined as

$$[\hat{N}^k] = \left[\mathcal{I}\left([N_S^k]\right) \mid \mathcal{I}\left([N_H^k]\right) \mid \mathcal{I}\left([N_\psi^k]\right)\right], \tag{29}$$

being $\mathcal{I} : \mathcal{M}_{p\times q} \to \mathcal{M}_{2p\times 2q}$ the pattern function defined by

$$\mathcal{I}([\boldsymbol{v}]) = \left[\begin{array}{cc} [\boldsymbol{v}] & 0 \\ 0 & [\boldsymbol{v}] \end{array}\right], \quad [\boldsymbol{v}] \in \mathcal{M}_{p\times q},$$

where p and q are natural numbers, it verifies relation (28).

In order to obtain the components of the gradient of the function $\boldsymbol{u}_{h|T^k}(\boldsymbol{x})$ defined by (27), we follow a similar procedure. Note that since the interface might introduce singularities on $\boldsymbol{u}_h$ or its derivatives, we only define the gradient components on points not belonging to the interface $\mathcal{S}$. When integrating on an element containing these singularities, a division into subelements will be necessary and some boundary terms might appear. We will deal with this issue on the following sections.

We denote by $[\nabla N_S^k]$ the 2×3 matrix whose columns are the gradient of the standard shape functions, which is given by

$$[\nabla N_S^k] = \left[\nabla N_{V_1^k} | \nabla N_{V_2^k} | \nabla N_{V_3^k}\right] = \left[\begin{array}{ccc} \partial_1 N_{V_1^k} & \partial_1 N_{V_2^k} & \partial_1 N_{V_3^k} \\ \partial_2 N_{V_1^k} & \partial_2 N_{V_2^k} & \partial_2 N_{V_3^k} \end{array}\right].$$

Similarly, $[\nabla^X N_H^k] = H[\nabla N_S^k]$ on $\Omega^+ \cup \Omega^-$, and

$$[\nabla N_\psi^k] = \left[\nabla \left[N_{V_1^k}[\Psi]\right] | \cdots | \nabla \left[N_{V_3^k}[\Psi]\right]\right], \tag{30}$$

which is easily obtained on $\Omega^+ \cup \Omega^-$ by using the chain rule as products of $N_{V_i^k}$, $[\Psi]$ and their gradients matrix, $[\nabla\Psi]$, given by

$$[\nabla\Psi] = \begin{bmatrix} \partial_1\Psi_1 & \partial_1\Psi_2 & \partial_1\Psi_3 & \partial_1\Psi_4 \\ \partial_2\Psi_1 & \partial_2\Psi_2 & \partial_2\Psi_3 & \partial_2\Psi_4 \end{bmatrix}, \tag{31}$$

which is well defined except at the tip point.

Then, the matrix $[\nabla\hat{N}^k]$ built as

$$[\nabla\hat{N}^k] = \left[\mathcal{I}\left([\nabla N_S^k]\right)\ \mathcal{I}\left([\nabla N_H^k]\right)\ \mathcal{I}\left([\nabla N_\psi^k]\right)\right],$$

belongs to $\mathcal{M}_{4\times n_{edf}}$, is well defined on $\Omega^+ \cup \Omega^-$ and taking into account (27) it verifies:

$$\left\{\begin{array}{c} \nabla u_{1h|T^k}(\boldsymbol{x}) \\ \hline \nabla u_{2h|T^k}(\boldsymbol{x}) \end{array}\right\} = [\nabla\hat{N}^k](\boldsymbol{x})\{u_h^k\},\ \boldsymbol{x} \in T^k \setminus \bar{S}. \tag{32}$$

From the gradient vector defined on (32) we can easily obtain the components of any other operator involving the partial derivatives, like for example the strain vector

$$\{\varepsilon(\boldsymbol{u}_h)(\boldsymbol{x})\}_{|T^k} = [\mathcal{D}_2][\nabla\hat{N}^k](\boldsymbol{x})\{u_h^k\},\ \boldsymbol{x} \in T^k \setminus \bar{S}, \tag{33}$$

being

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{Bmatrix} \text{ and } [\mathcal{D}_2] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}. \tag{34}$$

Remark 6. In order to compute the derivatives with respect to x and y of the Ψ shape functions we need to use the chain rule and the Jacobian matrix $[Q]$

$$[Q] = \begin{bmatrix} \cos\theta & -\frac{\sin\theta}{r} \\ \sin\theta & \frac{\cos\theta}{r} \end{bmatrix}, \quad \text{so that} \quad [\nabla_{xy}\Psi] = [Q][\nabla_{r\theta}\Psi].$$

Remark 7. In the following, to simplify the notation, we will omit the operator $\mathcal{I}$ on the matrix representations if no mistake is possible, so

$$\left[\mathcal{I}\left([N_S^k]\right)\ \mathcal{I}\left([N_H^k]\right)\ \mathcal{I}\left(\left[N_\psi^k\right]\right)\right]$$

will be denoted by

$$\left[N_S^k\ N_H^k\ N_\psi^k\right].$$

8. Elemental matrices and assembling

One of the main difficulties of the practical implementation of the extended finite elements is the variable number of degrees of freedom per node and element. Even for elements belonging to the same subset $\mathcal{T}_M$, $M \in \{H, \psi\}$, their nodes do not necessarily belong to the same set of nodes as has been pointed out in Remark 2. Therefore, there will be standard elements with 6 degrees of freedom, or an element with all nodes enriched with the $\psi-$functions, in which case the element will have 30 degrees of freedom associated, as well as many combinations in between. In particular, on a two-dimensional domain and for a mesh of triangular elements almost any even number between 6 and 30 might occur depending on the particular element. The matrix representation presented on Section 7 for a function $\boldsymbol{u}_h$ restricted to T^k was (see (28), (29) and Remark 7):

$$\boldsymbol{u}_{h|T^k} = \left[\hat{N}^k\right]\{u_h^k\} = \left[N_S^k\ N_H^k\ N_\psi^k\right]\{u_h^k\}.$$

Let us note also that the matrix formulation on elements belonging to $\mathcal{T}_S$ or $\mathcal{T}_H$ is slightly different, needing only $[N_S^k]$ or $[N_S^k\ N_H^k]$, respectively. There are several possible combinations, but in practice we only need to distinguish between the three element sets presented at the beginning of this paper, $\mathcal{T}_h$, $\mathcal{T}_H$ and $\mathcal{T}_\psi$. We will present here just the more complex case, since the other two are similar. The elementary matrices will be built by blocks and for each element at most three arrays will be used to relate the local ordering of its degrees of freedom with their respective global ordering.

8.1. Elemental matrices

In this section we will study the computation of the elementary stiffness and mass matrices in the more general case, assuming that the element belongs to $\mathcal{T}_\psi$, so it could have the three kinds of degrees of freedom. In order to do so, we will use the matrix formulations (28) and (32) obtained in Section 7.

We will see first the computation of the mass matrix, related to terms like $\int_\Omega \boldsymbol{u}_h \cdot \boldsymbol{v}_h \, d\boldsymbol{x}$.

As usual on the FEM, because of the properties of the shape functions N_I, the domain integral can be obtained more easily as the sum of the elementary integrals,

$$\int_\Omega \boldsymbol{u}_h \cdot \boldsymbol{v}_h \, d\boldsymbol{x} = \sum_{T^k \in \mathcal{T}_h} \int_{T^k} \boldsymbol{u}_{h|T^k} \cdot \boldsymbol{v}_{h|T^k} \, d\boldsymbol{x}. \tag{35}$$

With the matrix formulation (28), the elementary integral on T^k can be expressed as:

$$\int_{T^k} \boldsymbol{u}_{h|T^k} \cdot \boldsymbol{v}_{h|T^k} \, d\boldsymbol{x} = \int_{T^k} \left[\hat{N}^k\right] \{u_h^k\} \cdot \left[\hat{N}^k\right] \{v_h^k\} \, d\boldsymbol{x} = \int_{T^k} \{v_h^k\}^t \left[\hat{N}^k\right]^t \left[\hat{N}^k\right] \{u_h^k\} d\boldsymbol{x} = \{v_h^k\}^t \left(\int_{T^k} \left[\hat{N}^k\right]^t \left[\hat{N}^k\right] \, d\boldsymbol{x} \right) \{u_h^k\},$$

and therefore, the elementary mass matrix, $[M^k]$, with dimension $n_{edf} \times n_{edf}$ is defined by

$$[M^k] = \int_{T^k} \left[\hat{N}^k\right]^t \left[\hat{N}^k\right] d\boldsymbol{x}, \tag{36}$$

so

$$\int_{T^k} \boldsymbol{u}_{h|T^k} \cdot \boldsymbol{v}_{h|T^k} \, d\boldsymbol{x} = \{v_h^k\}^t [M^k] \{u_h^k\}. \tag{37}$$

Similarly, the stiffness matrix, related to terms like $\int_{\Omega^\pm} \nabla \boldsymbol{u}_h : \nabla \boldsymbol{v}_h \, d\boldsymbol{x}$, is computed on each element as

$$\int_{T^{k\pm}} \nabla \boldsymbol{u}_{h|T^{k\pm}} : \nabla \boldsymbol{v}_{h|T^{k\pm}} \, d\boldsymbol{x} = \int_{T^{k\pm}} \left[\nabla \hat{N}^k\right] \{u_h^k\} \cdot \left[\nabla \hat{N}^k\right] \{v_h^k\} \, d\boldsymbol{x} = \{v_h^k\}^t \left(\int_{T^{k\pm}} \left[\nabla \hat{N}^k\right]^t \left[\nabla \hat{N}^k\right] \, d\boldsymbol{x} \right) \{u_h^k\},$$

being $T^{k\pm} = T^k \cap \Omega^\pm$. The elementary stiffness matrix is defined by $[K^k] = [K^{k+}] + [K^{k-}]$, being

$$[K^{k\pm}] = \int_{T^{k\pm}} \left[\nabla \hat{N}^k\right]^t \left[\nabla \hat{N}^k\right] d\boldsymbol{x}. \tag{38}$$

Remark 8. Note that given the particular form of $[\hat{N}^k]$ and $[\nabla \hat{N}^k]$, the elementary matrices can be built by blocks, so for example (remember again Remark 7)

$$[M^k] = \begin{bmatrix} \left[N_S^k\right]^t \left[N_S^k\right] & \left[N_S^k\right]^t \left[N_H^k\right] & \left[N_S^k\right]^t \left[N_\psi^k\right] \\ \left[N_H^k\right]^t \left[N_S^k\right] & \left[N_H^k\right]^t \left[N_H^k\right] & \left[N_H^k\right]^t \left[N_\psi^k\right] \\ \left[N_\psi^k\right]^t \left[N_S^k\right] & \left[N_\psi^k\right]^t \left[N_H^k\right] & \left[N_\psi^k\right]^t \left[N_\psi^k\right] \end{bmatrix}, \tag{39}$$

where $[M^k] \in \mathcal{M}_{36\times36}$, and similarly for the elementary stiffness matrix. Also noticeable is that the non-diagonal blocks involving the Ψ functions are not square. For example, $\left[N_\psi^k\right]^t \left[N_S^k\right] \in \mathcal{M}_{24\times6}$. Actually, the whole elementary matrix is never built, the several blocks are computed and assembled separately. That is why three destination arrays will be used, one for each kind of degrees of freedom, which are combined to assemble the different blocks.

Any other integral involving divergence or strain deformation tensor can be obtained similarly just by using the appropriate matrices and patterns.

8.2. Computing the assembling matrices

In order to pass from the global to the local degrees of freedom on the previous section, we used the matrix $[A^k] \in \mathcal{M}_{n_{edf} \times n_{gdf}}$ verifying (24).

In this section we will see how this matrix can be used also to assemble the elementary matrices just computed. The property that will allow us to do it is that

$$[A^k][A^k]^t = Id(n_{edf}), \tag{40}$$

so, if we consider expression (36) we can actually write

$$\sum_k \{v_h^k\}^t[M^k]\{u_h^k\} = \sum_k \{v_h^k\}^t[A^k][A^k]^t[M^k][A^k][A^k]^t\{u_h^k\} = \{v_h\}^t\left(\sum_k [A^k]^t[M^k][A^k]\right)\{u_h\}.$$

Defining the global mass matrix, $[M]$, as

$$[M] = \sum_k [A^k]^t[M^k][A^k] \in \mathcal{M}_{n_{gdf}\times n_{gdf}},$$

we can rewrite from (35) and (37)

$$\int_\Omega \boldsymbol{u}_h \cdot \boldsymbol{v}_h \, d\boldsymbol{x} = \{v_h\}^t[M]\{u_h\}.$$

Note also that the assembling matrix for element k can be built by blocks,

$$[A^k] = \begin{bmatrix} A_S^k \\ A_H^k \\ A_\psi^k \end{bmatrix},$$

where the matrices $[A_L^k]$, $L \in \{S,H\}$ belong to $\mathcal{M}_{6\times n_{gdf}}$ and $[A_\psi^k] \in \mathcal{M}_{24\times n_{gdf}}$, so, thanks to (39), one can assemble the blocks of the elementary matrices separately as

$$[M] = \sum_k \sum_{L,M} [A_L^k]^t[N_L^k]^t[N_M^k][A_M^k] \in \mathcal{M}_{n_{gdf}\times n_{gdf}},\ T^k \in \mathcal{T}_h,\ L, M \in \{S,H,\psi\}.$$

The stiffness matrix can be computed in a similar way from the elementary stiffness matrices and the assembling matrix.

For the practical implementation point of view, there are two key tools when defining the matrix components:

- The adjacency node-element operator $[C_{hh}]$, defined on Section 4, giving for each element the set of nodes belonging to this element with their numbering on $\mathcal{P}_h$.
- The transforming index functions p, p_H and p_ψ, assigning to each elementary degree of freedom a position on the global ordering. Such ordering is a programmer's choice. We consider the following ones:

  $$\mathsf{p}(s,I_h) = s + (I_h - 1)2,\ \ \mathsf{p}_H(s,I_H) = \begin{cases} s + (I_H - 1 + n_{hn})2 & \text{if } I_H > 0, \\ 0 & \text{if } I_H = 0, \end{cases}$$

  while

  $$\mathsf{p}_\psi(s,l,I_\psi) = s + 2(l - 1 + 4(I_\psi - 1) + n_{hn} + n_{Hn})$$

  if $I_\psi > 0$ and $\mathsf{p}_\psi(s,l,I_\psi) = 0$ otherwise. In these expressions $1 \le s \le 2$ relates to the component, $l \in \{1,\dots,4\}$ relates to the function Ψ_l and $1 \le I_h \le n_{hn}$, $1 \le I_H \le n_{Hn}$, $1 \le I_\psi \le n_{\psi n}$ are the global numbering of the node in $\mathcal{P}_h$, $\mathcal{P}_H$ and $\mathcal{P}_\psi$, respectively. Therefore, for example, $\mathsf{p}(s,I)$ gives the global numbering of the standard degree of freedom associated to the sth component of the Ith node on the mesh and $\mathsf{p}_\psi(s,l,I_\psi)$ gives the global numbering of the degree of freedom associated to the sth component of the Ψ_l function of the I_ψ-th node on $\mathcal{P}_\psi$.

The composition of these functions will define the assembling matrix components. Since the standard degrees of freedom will occupy the first $2n_{hn}$ positions, the global numbering of the Heaviside degrees of freedom has to start with $2n_{hn}+1$ which corresponds to $\mathsf{p}_H(1,1)$, and the ψ ordering will start with $2(n_{hn}+n_{Hn})+1$, corresponding to $\mathsf{p}_\psi(1,1,1)$.

Now, given an index j in the local ordering of the element T^k, $1 \le j \le n_{edf}$, we need to recover the component, $s(j)$, local node, $i(j)$, global node, $I(i(j))$, and Ψ_l function, $l(j)$ (if that is the case), to which the jth elementary degree of freedom is associated to. In order to do that, we use the floor function $E[a]$ and the modulo operator $[a]_m$ defined as

$$E[a] = \max\{z \in \mathbb{Z}; z \le a\},\ [a]_m = a - mE\left[\frac{a}{m}\right],\ a \in \mathbb{R},\ m \in \mathbb{N}.$$

Considering the local ordering of the elementary degrees of freedom established on (25) we know that

$$s(j) = \begin{cases} E\left[\frac{j-1}{3}\right] + 1 & 1 \le j \le j^\star, \\ E\left[\frac{j-1-j^\star}{3}\right] + 1 & j^\star + 1 \le j \le 2j^\star, \\ E\left[\frac{j-1-2j^\star}{12}\right] + 1 & 2j^\star + 1 \le j \le n_{edf}, \end{cases} \tag{41}$$

with $j^\star = 6$, gives the displacement component associated to which the jth elementary degree of freedom contributes, while

$$i(j) = \begin{cases} [j-1]_3 + 1 & 1 \le j \le 2j^\star, \\ \left[E\left[\frac{j-1-2j^\star}{4}\right]\right]_3 + 1 & 2j^\star + 1 \le j \le n_{edf}, \end{cases} \tag{42}$$

gives the local number of the node (vertex) at the element k,

$$I(j,k)=\begin{cases}[C_{hh}]_{i(j)k} & 1\le j\le j^\star,\\ n_H^{-1}([C_{hh}]_{i(j)k}) & j^\star+1\le j\le 2j^\star,\\ n_\psi^{-1}([C_{hh}]_{i(j)k}) & 2j^\star+1\le j\le n_{edf},\end{cases}\tag{43}$$

gives the number of the global node in the mesh $\mathcal{T}_h$, and

$$l(j)=\begin{cases}0 & 1\le j\le 2j^\star,\\ \left[j-1-2j^\star\right]_4+1 & 2j^\star+1\le j\le n_{edf},\end{cases}\tag{44}$$

gives the Ψ_l function involved.

So, the assembling matrix is defined as:

$$[A_M^k]_{jq}=\delta_{q\xi_M^k(j)},\, M\in\{S,H,\psi\},$$

being

$$\begin{aligned}&\xi_S^k(j)=\mathsf{p}\left(s(j),I(j,k)\right),\ \xi_H^k(j)=\mathsf{p}_H\left(s(j+j^\star),I(j+j^\star,k)\right),\ 1\le j\le j^\star,\\ &\xi_\psi^k(j)=\mathsf{p}_\psi\left(s(j+2j^\star),l(j+2j^\star),I(j+2j^\star,k)\right),\ 1\le j\le 4j^\star.\end{aligned}\tag{45}$$

Note that the position of the non-null element on the jth row of $[A_M^k]$ is given by $\xi_M^k(j)$, so the implementation can be done by using the vectors ξ_M^k without actually building the assembling matrices.

The null positions on $[A_M^k]$, $M\in\{H,\psi\}$ will indicate which rows and/or columns from the elementary mass and stiffness matrices will not be assembled, since they correspond with non-enriched nodes on elements in $\mathcal{T}_H$ and $\mathcal{T}_\psi$.

8.3. Jump operator on the interface

Now, considering the iterative algorithm described in Section 3.1 for solving the nonlinearity arising by the contact between the lips of the crack, we will see the implementation of the integral term on the interface at each iteration:

$$\frac{1}{\epsilon}\int_{\mathcal{S}_{j-1}^c}B(\boldsymbol{u}_h^j)B(\boldsymbol{v}_h)\,d\Gamma,$$

arising in variational formulation (11). It deals with the contact condition between the crack faces, being $\mathcal{S}_{j-1}^c$ the subset of edges in $\mathcal{S}_h$ where effective contact happens at iteration $j-1$ of the iterative algorithm.

Let us recall the expression in (27) that gives the displacement of a point $\boldsymbol{x}\in T_k\in\mathcal{T}_h$. If a point $\boldsymbol{x}_0\in T^k$ was to be on the crack, H would not be defined in $\boldsymbol{x}_0$, nor would be $\Psi_1(\boldsymbol{x}_0)=\sqrt{r(\boldsymbol{x}_0)}\sin(\pm\pi/2)$, since both are discontinuous on $\theta=\pm\pi$ (see (16). Therefore, the XFEM discretization of the jump $B(\boldsymbol{u}_h)=[\boldsymbol{u}_h\cdot\boldsymbol{n}]$ at a point $\boldsymbol{x}_0\in T^k\cap\mathcal{S}_h$ on the crack becomes

$$[\boldsymbol{u}_h(\boldsymbol{x}_0)\cdot\boldsymbol{n}]=\sum_{i=1}^{3}2N_{V_i^k}(\boldsymbol{x}_0)\boldsymbol{n}\cdot\left(\boldsymbol{c}_{V_i^k}+\sqrt{r(\boldsymbol{x}_0)}\boldsymbol{b}^1_{V_i^k}\right).$$

In order to obtain the global operator $[S]$ such that for every $\boldsymbol{x}\in\mathcal{S}_h$

$$[\boldsymbol{u}_h(\boldsymbol{x})\cdot\boldsymbol{n}]=[S](\boldsymbol{x})\{u_h\},\tag{46}$$

for each edge $e_m\in\mathcal{S}_h$, and at each point $\boldsymbol{x}\in e_m$, we consider the elementary matrix, $[S^m](\boldsymbol{x})\in\mathcal{M}_{1\times 6}$, given by the following vectors:

$$[S^m](\boldsymbol{x})=\left[\begin{array}{cc}[N_S^{D(m)}](\boldsymbol{x})n_1 & [N_S^{D(m)}](\boldsymbol{x})n_2\end{array}\right],$$

being $\boldsymbol{x}$ an interior point of the mth edge, e_m, $\boldsymbol{n}^+=(n_1,n_2)$ and D the application defined in Section 4.4 that assigns to the mth interface edge, the corresponding mesh element, T^k, $k=D(m)$, to which e_m belongs to, according to the diagram in Fig. 6. Therefore, for $m=1,\dots,n_{hf}$ we get

$$[\boldsymbol{u}_h(\boldsymbol{x})\cdot\boldsymbol{n}]_{|e_m}=2[S^m](\boldsymbol{x})(\{\boldsymbol{c}^{D(m)}\}+\sqrt{r(\boldsymbol{x})}\{\boldsymbol{b}^{1,D(m)}\}),$$

for every $e_m\in\mathcal{S}_h$ being

$$\{\boldsymbol{c}^{D(m)}\}=\left\{\begin{array}{c}\{c_1^{D(m)}\}\\ \hline \{c_2^{D(m)}\}\end{array}\right\}\text{ and }\{\boldsymbol{b}^{1,D(m)}\}=\left\{\begin{array}{c}\{b_1^{1,D(m)}\}\\ \hline \{b_2^{1,D(m)}\}\end{array}\right\},$$

the vectors analogous to those defined in (25), whose components are the local degrees of freedom associated to the H and Ψ_1 enrichment functions for the nodes of the element, $D(m)$, to which e_m belongs to. Notice that the components of the vector $\{\boldsymbol{c}^{D(m)}\}$ correspond to the components with indices $\mathsf{p}_H(:,n_H^{-1}(C_{hh}(:,D(m))))$ of the global nodal vector $\boldsymbol{u}_h$. Besides, the components of $\{\boldsymbol{b}^{1,D(m)}\}$

correspond to the components with indices $\mathsf{p}_\psi(:,1,n_\psi^{-1}(C_{hh}(:,D(m))))$ of the global nodal vector $\boldsymbol{u}_h$. From now on, in order to simplify the notation, we will omit the dependence on $\boldsymbol{x}$.

Let

$$\zeta_j^m = \begin{cases} \mathsf{p}_H(s(j), n_H^{-1}([C_{hh}]_{i(j),D(m)})), & \text{if } S_{HH}^{-1}(m) \neq 0, \\ \mathsf{p}_\psi(s(j), 1, n_\psi^{-1}([C_{hh}]_{i(j),D(m)})), & \text{if } S_{\psi\psi}^{-1}(m) \neq 0, \end{cases}$$

with $j = 1,\dots,6$ and S_{HH}, $S_{\psi\psi}$ the adjacency operators defined in Section 4.5. Then, the global jump matrix $[\mathbf{S}] \in \mathcal{M}_{n_{hf}\times n_{gdf}}$ has as components

$$[S]_{m\zeta_j^m} = \begin{cases} [S^m]_j, & \text{if } S_{HH}^{-1}(m) \neq 0, \\ \sqrt{r(\boldsymbol{x})}[S^m]_j, & \text{if } S_{\psi\psi}^{-1}(m) \neq 0, \end{cases}$$

for every $j = 1,\dots,6$.

Iterative algorithm and interface integral computation. Let us briefly recall the main steps of the iterative algorithm used to compute $(\boldsymbol{u}_h, \boldsymbol{\sigma}_h)$ at time t. A detailed explanation of such algorithm can be found in [28].

- Given $(\boldsymbol{u}_h^{j-1}, \boldsymbol{\sigma}_h^{j-1})$ and S_{j-1}^c, compute $(\boldsymbol{u}_h^j, \boldsymbol{\sigma}_h^j)$ verifying (11) for all $\boldsymbol{v}_h \in V_h^X$.
- Known $(\boldsymbol{u}_h^j, \boldsymbol{\sigma}_h^j)$, compute $B(\boldsymbol{u}_h^j)$ defined by (37).
- Compute S_j^c, the subset of edges in S_h where effective contact happens at iteration j, by using the condition $B(\boldsymbol{u}_h^j) \geq 0$.

Since the integral term

$$\int_{S_{j-1}^c} B(\boldsymbol{u}_h^j) B(\boldsymbol{v}_h)\, d\Gamma$$

might be computed on a different subset of edges at each iteration and not necessarily on the whole linearized interface S_h, in practice we do not use the whole global jump matrix $[S]$. Instead, on each iteration we only use some rows $[S]_m \in \mathcal{M}_{1\times n_{gdf}}$, $m \in E_{j-1} = \{m/e_m \in S_{j-1}^c\}$, so that the integral term on the interface can be computed as

$$\int_{S_{j-1}^c} B(\boldsymbol{u}_h^j) B(\boldsymbol{v}_h)\, d\Gamma = \sum_{m\in E_{j-1}} \int_{e_m} B(\boldsymbol{u}_h^j) B(\boldsymbol{v}_h)\, d\Gamma = \sum_{m\in E_{j-1}} \int_{e_m} \{v_h\}^t [S]_m^t [S]_m \{u_h^j\}\, d\Gamma.$$

Remark 9. The integration on the interface has been exemplified with a term involving jumps of the displacement. Of course, the same can be done if we need to integrate other discontinuous fields, like pressure, or gradient terms on the interface. The particularities of the global jump operator $[S]$ will depend on the enrichment functions considered and their derivatives (whether they are continuous across the interface or not). For example, taking the practical use case as a model, and using the discrete representation of the gradient given in (32), a similar global operator for the jump of a gradient field can be obtained.

9. Numerical results

In this section we will provide some numerical results obtained for the practical use case and compare them with its resolution with FEM. Also, there will be a section devoted to compare computational times for the implementation of both methods with several meshes and geometries.

9.1. Simulation of Rayleigh waves propagation problem

In this section we show some numerical results obtained with the XFEM, as well as a comparison with the results obtained by using classical FEM to solve the same problem. In order to solve with FEM, the mesh needs to be adapted to the crack, so the elements crossed by the crack have to be divided, but except for those elements, the meshes used with both methods match. The same splitting technique used for the integration in XFEM is used to compute these new elements needed to adapt the mesh to the interface for the FEM simulations. In Fig. 7 an example of both meshes is shown.

A comparison of the performance for both methods by using academic tests can be seen at [27], where solutions obtained for fractures Mode I (opening) and Mode II (in-plane shear) are shown. We will focus here on the contact problem that arises when a Rayleigh wave propagates on the surface of a plate.

Therefore, we will compare the solutions obtained for problem (1)–(8) solved by using XFEM and FEM. In both cases the time discretization is performed by using an implicit method from the Newmark's family (see [28] or [42]), taking $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$, for which the method is implicit and unconditionally stable. We consider a domain $\Omega = [0.7854, 0.8114] \times [-0.02, 0]\,\mathrm{m}^2$ with a crack given by the segment $\{0.7942\} \times [-0.0025, 0]\,\mathrm{m}$. The material parameters considered are $\rho = 2700\,\mathrm{kg/m}^3$, $E = 7.4100\mathrm{E}+10\,\mathrm{N/m}^2$ and $\nu = 0.3302$. The Rayleigh wave has velocity $3.e+3\,\mathrm{m/s}$ and wavelength $3.e-3\,\mathrm{m}$. The characteristic mesh size is considered small enough so that more than ten elements fit on a wavelength. In particular, we are considering meshes whose characteristic size is in the order of $2.e-5\,\mathrm{m}^2$. Finally, the time step considered is $\Delta t = 1.e-8\,\mathrm{s}$.

Two graphs corresponding to the displacements experienced by the surface of the plate in two different instants are presented in Figs. 8 and 9, before the wave reaches the crack and once it has been overcome it. In each graph, three lines are shown, corresponding

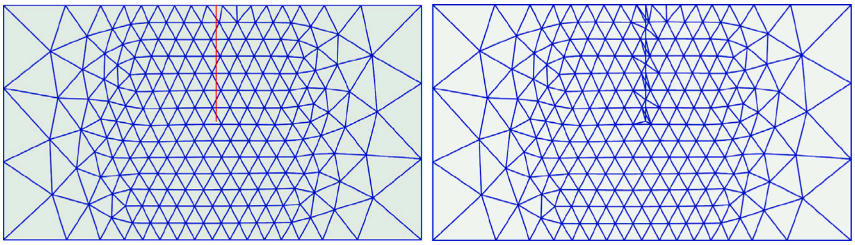

Fig. 7. Zoom of the meshes for geometry 1 for XFEM computations and adapted mesh for FEM.

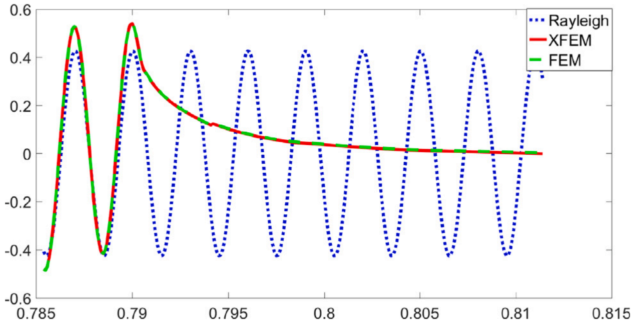

Fig. 8. Displacements on the plate surface at time $t = 1.6\text{E}{-}6\,\text{s}$ computed with XFEM and FEM.

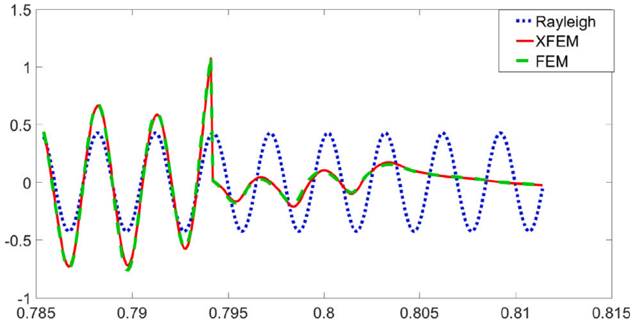

Fig. 9. Displacements on the plate surface at time $t = 6.\text{E}{-}6\,\text{s}$ computed with XFEM and FEM.

to the displacements calculated with XFEM, FEM, and those corresponding to the theoretical expression of the Rayleigh wave obtained in [28]. The displacement associated with a theoretical Rayleigh wave always vibrating in a plate without a crack is represented in dashed–dotted blue line, while in dashed red line the wave calculated by the XFEM method is shown and in solid green line the wave calculated by the FEM method. As can be seen, both approximations are practically equal, so the XFEM methodology to perform the approximation is validated.

Fig. 10 shows the difference, in absolute value, between the Von Mises norm for stresses computed with XFEM and FEM, for each integration point of the mesh, in two time instants, before and after the wave reaches the crack. Indeed, the figure on the left corresponds with $t = 1.6\text{E}{-}6\,\text{s}$ and the one on the right with $t = 6.\text{E}{-}6\,\text{s}$. It can be seen that the greatest difference in stress occurs near the tip, as expected, as it is the area of influence of the functions of enrichment.

For the stress measurement we take four particular points on the plate: one very close to the crack tip, $X1 = (0.79417, -0.00245)$, a remote point located after the crack, $X2 = (0.8091, -0.004728)$ and two remote points, located before the crack, $X3 = (0.7888, -0.004569)$ and $X4 = (0.7869, -0.000484)$, the latter also near the surface of the plate.

As can be seen on the left graph on Fig. 11, at $X1$, which is the closest to the crack tip, is where there are greater differences between the stresses computed with XFEM and the stresses computed with FEM. However, at $X4$ (right graph on Fig. 12), which is a point near the surface, where the predominant tensions are those of the Rayleigh wave, the results are practically identical with both methods.

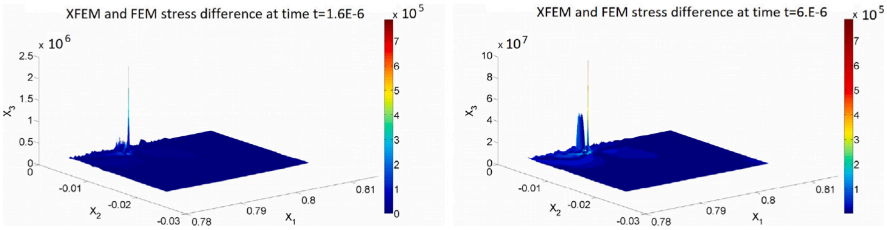

Fig. 10. Difference between Von Mises norm for stress computed with XFEM and FEM at time $t = 1.6\text{E}{-6}\,\text{s}$ and $t = 6.\text{E}{-6}\,\text{s}$.

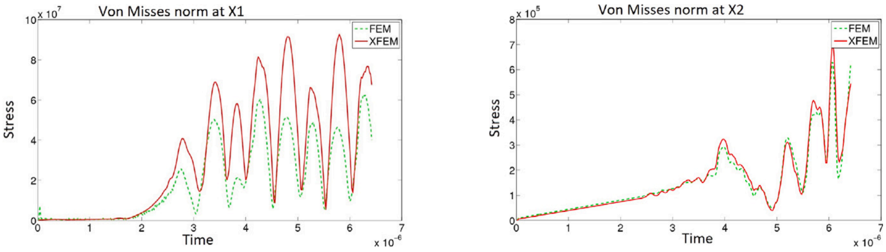

Fig. 11. Von Mises norm for stress at $X1$ and $X2$ computed with XFEM and FEM.

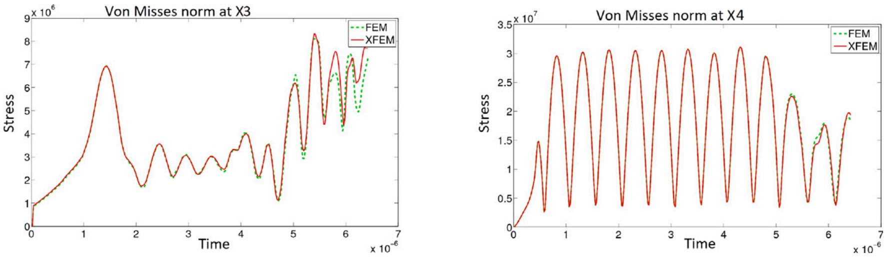

Fig. 12. Von Mises norm for stress at $X3$ and $X4$ computed with XFEM and FEM.

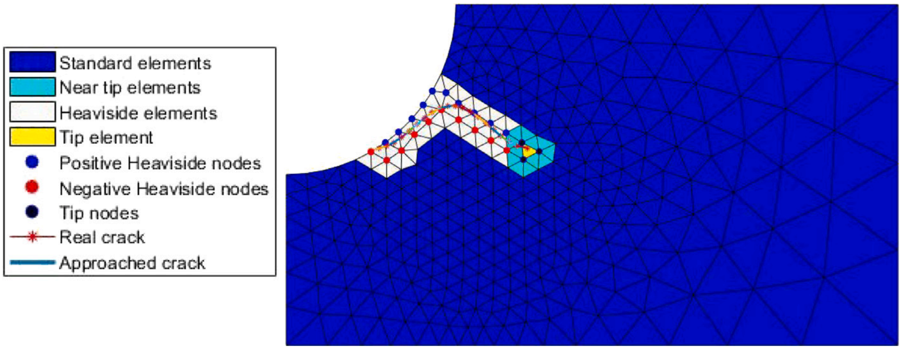

Fig. 13. Coarse mesh for geometry 2.

9.2. Computational times comparison

Regarding computational times, we will show next several tables displaying the times needed for several steps of the XFEM and FEM computations. In particular, we focus on the selection of enriched nodes and elements (bounding box time), the computation of quadrature nodes and weights for enriched elements (including the automatic computation of partition matrices), and the assembling of the system matrices and vectors.

For this study we consider two geometries for the domain and the interface. The first one, showed in Fig. 7 is a rectangle with a straight interface used for the previous simulations, and the second one, showed in Fig. 13, is a rectangle minus a circle on a

Table 1
Mesh description and computational times for Geometry 1.

Mesh	Geometry 1				
	1	2	3	4	5
N. Nodes XFEM (n_{hn})	234	860	3358	20 811	83 305
N. Nodes FEM	247	901	3445	20 902	83 487
N. Elts. XFEM	418	1625	6528	41 146	165 666
N. Elts. FEM	431	1683	6652	41 216	165 796
Bounding Box [s]	3.1035	3.8192	3.3177	8.2405	22.5336
Quadrature XFEM [s]	0.2138	0.2609	0.3646	1.8210	7.1894
$[M], [K]$ Assembly XFEM [s]	2.0190	2.08	2.5902	15.4181	628.3408
$[S]$ Assembly XFEM [s]	0.0055	0.0124	0.0129	0.0145	0.0231
$[M], [K]$ Assembly FEM [s]	0.1397	0.1721	0.7145	15.3611	586.5261

Table 2
Mesh description and computational times for Geometry 2.

Mesh	Geometry 2				
	1	2	3	4	5
N. Nodes XFEM (n_{hn})	323	1332	7950	31 497	125 718
N. Nodes FEM	366	1421	8177	31 948	126 621
N. Elts. XFEM	592	2556	15 648	62 496	250 436
N. Elts. FEM	635	2645	15 875	62 947	251 339
Bounding Box [s]	6.6718	7.3037	7.5514	8.6688	12.0418
Quadrature XFEM [s]	0.1025	0.3005	0.8314	2.6685	10.1301
$[M], [K]$ Assembly XFEM [s]	2.0776	2.2252	4.4396	38.0478	1839.0314
$[S]$ Assembly XFEM [s]	0.0145	0.0142	0.0179	0.0212	0.0377
$[M], [K]$ Assembly FEM [s]	0.1577	0.2989	2.4857	44.8116	2101.6107

corner, and a curved interface. For both geometries five meshes will be considered and the computational times will be shown for all of them.

We can see in Tables 1 and 2 that the computational time for the selection of enriched nodes and elements (bounding box) is of the same order than the assembling time for very coarse meshes, but when we pass to very refined meshes, it is much faster by several orders of magnitude. Also notice the computational times for the computation of the quadrature points and weights on enriched elements, which includes the automatic partition of enriched elements, and it is in all cases at least one order of magnitude faster than the assembling.

Notice also the really small time that the assembling of the jump matrix, $[S]$, takes in any case. Since this matrix only involves enriched degrees of freedom, its assembling is really fast, and also gives an idea of the time that would take to modify the enriched part of the mass and stiffness matrices, $[M]$ and $[K]$, were the interface to evolve and the sets of enriched nodes and elements were to change from one time step to another. This is where one can see the computational advantages that XFEM provides, since instead of remeshing and recalculating and assembling again all the system matrices, with XFEM one only needs to recalculate the bounding box and assembling just the blocks involving enriched nodes. Let us take for example mesh 5 for Geometry 2, and let us think of an evolving interface. With FEM, on each time step one would need to not only reassemble matrices $[M]$ and $[K]$ on each time step $O(10^3)$, but also remesh the domain. With XFEM, one just needs to recompute the bounding box $O(10)$ and modify the enriched blocks of $[M]$ and $[K]$ which one can assume will be the same order of assembling $[S]$, let us say, $O(10^{-1})$ for all three matrices together. On each time step, XFEM would be two orders of magnitude faster than FEM, even disregarding remeshing time for the latter.

10. Conclusions

We presented here a mathematical perspective on XFEM implementation methodology that can be applied to any model and any programming language with minor changes even those models involving jump terms on the interface. In addition, the proposed methodology is generalizable independently of the numerical discretization and very efficient regarding computational times, which is extremely important above all when there are terms appearing inside iterative algorithms that require their repeated computation.

Such analysis has been vertebrated with a series of mathematical tools (sets, functions, embeddings, etc.) that define and relate the key features of the XFEM, like the enriched elements, its nodes, and corresponding degrees of freedom in a way that allows generalization to any model, even with different particularities. This complete procedure for the implementation of the XFEM for problems involving contributions on interfaces has been analysed in detail using as framework a mathematical model corresponding to Rayleigh waves propagation on a damaged two dimensional plate, considering Signorini contact conditions between the sides of the crack. Besides, suitable approximation spaces and discrete variational formulation of the problem have been proposed.

The necessary enrichment functions to replicate discontinuous displacement and singular stress are reviewed as well as the different types of nodes and elements that arise from such enrichment. Several embeddings relating these sets with each other are introduced. These embeddings play a key role in the rigorous exposition of the implementation methodology as well as in its systematization to be easily generalizable to other contexts.

The discrete expression of the displacements, their jump across the linearized interface, and the associated strain vector are detailed in terms of the global degrees of freedom vector. These are the key points where the novelties with respect to FEM implementation arise, since the XFEM involves several kinds of degrees of freedom (and shape functions) that FEM does not consider.

The computation of the necessary elementary matrices and their efficient assembling have been studied, by means of several local-to-global mappings. Special emphasis was made on the treatment of the interface integral term, showing that only some degrees of freedom associated with the enrichment functions are involved in the computation of the jump.

A comparison between the results of the numerical simulation using XFEM and FEM for the same problem was presented. It showed that both methods provide very similar results on displacements (which was expected) and they allow to detect defects by comparing the results with those of a non-damaged plate. Nevertheless, the results for stress were very similar for points away from the crack tip but quite different in its neighbourhood. In this case, the XFEM results approximate better the singularity of the stress in the crack tip, which FEM does not capture. Therefore, the use of XFEM is crucial for the calculation of the associated energy in this area, and then to assess adequately whether the crack will continue to grow or not.

Also, for the sake of completeness, a methodology of automatic element partition and integration over enriched elements is briefly reviewed.

The example case that has served as a guide in this paper to present the complete procedure includes the terms that usually appear in the different algorithms presented in the literature to discretize using XFEM problems that involve some kind of discontinuity in the interface, that need the calculation of integrals on it, or that present singularities in some area of its domain. Therefore, the proposed implementation procedure is useful in all of them and avoids not only the particular cases in the partition of an element crossed by the interface, but also the loss of precision in the numerical integration formulas used either on elements crossed by the interface, or on the interface itself.

We can summarize then the advantages of XFEM in two fronts: the first one, accuracy. To get the same accuracy one will need a much finer mesh for FEM than for XFEM. The second one, computational cost. Should the interface be to evolve from one step to the next, the need for remeshing and reassembling of FEM is much more expensive than the re-computation of the bounding box and modification of enriched blocks of the matrices in XFEM case. Therefore, if the re-computation of the bounding box and modification of enriched blocks of the matrices is well optimized, XFEM will be not only more accurate but also more computationally efficient that FEM. In this work we provide the reader the tools to systematize many aspects of the implementation, differentiating initial computations from iterative ones, as well as avoiding unnecessary decision loops, which leads to a more structured and efficient code, whatever programming language is used.

Funding

This work has been supported by FEDER, Xunta de Galicia, Spain funds under the ED431C 2017/60 and ED431C 2021/15 grants and by the Ministry of Science and Innovation, Spain through the Agencia Estatal de Investigación (PID2019-105615RB-I00/AEI / 10.13039/501100011033) and European Union Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie Grant Agreement No 823731 CONMECH.

CRediT authorship contribution statement

M.T. Cao-Rial: Study conceptualization, Methodology, Software, Writing – first draft. **C. Moreno:** Study conceptualization, Methodology, Software. **P. Quintela:** Study conceptualization, Methodology, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors have no relevant financial or non-financial interests to disclose.

Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Annexe A. List of notations

Given the amount of notations and indices on this paper, in order to make the reading easier we present in Tables A.3 and A.4 a compilation of the most used notations. In both tables, the index , $L \in \{H, S, \psi\}$ or in the case of adjacency matrices $L, M \in \{h, S, H, \Psi\}$.

Table A.3
Table of notations 1.

Symb.	Meaning
$.^e$	relative to a subelement in an element
$\cdot_j$	elementary degrees of freedom index
$\cdot_p$	quadrature index
$\cdot_s$	components index $1 \leq s \leq 2$
$\boldsymbol{a}_{ir}^k$	characteristic points on T^k on edge $\bar{ir}$
e_i	$i-$th edge in $\mathcal{S}_h$
$\boldsymbol{f}, \boldsymbol{g}$	volume and surface forces
$\bar{f}$	smoothing of a general function f over an element
$\bar{ir}$	edge in T^k with vertices V_i^k and V_r^k
$\boldsymbol{n}$	normal vector
$\boldsymbol{t}^k$	crack ending point in T^k
$n_{edf(gdf)}$	number of elementary (global) degrees of freedom
$n_{Le(n)}$	number of $L-$elements (nodes)
n_{he}, (n_{hn})	number of mesh elements (nodes)
n_{hf}	number of edges of the linearized interface
n_L, $\boldsymbol{n}_L$	scalar and vector embeddings between node sets
n^k	number of subtriangles in a partition of T^k
n_q	number of quadrature points
$r(c)$	radial distance to the crack tip
$\boldsymbol{u}$	displacement field
$\boldsymbol{u}_D$, $\boldsymbol{u}_0$, $\boldsymbol{v}_0$	Dirichlet and initial conditions resp.
$\boldsymbol{u}_{I_h}$, $\boldsymbol{c}_{I_H}$, $\boldsymbol{b}_{I_\psi}$	vectors of degrees of freedom
B	jump operator
$[C_{LM}]$	adjacency matrix
C_{LM}	adjacency operators element-nodes
D, D_{HH}, $D_{\Psi\Psi}$	adjacency operators edge-element
p, p_H, p_ψ	local-to-global index functions
H	discontinuous Heaviside-type function
I	node and its global index $1 \leq I \leq n_{hn}$

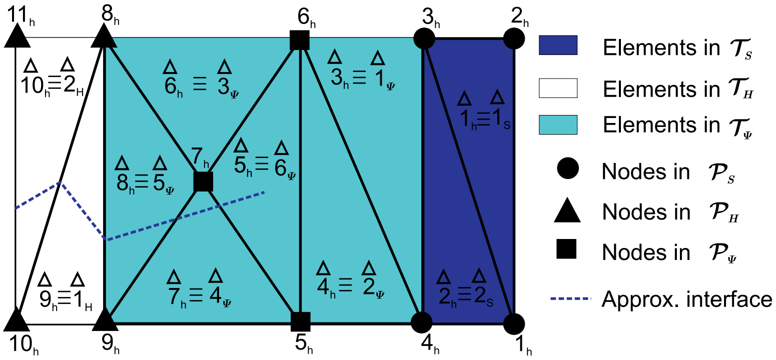

Fig. 14. Mesh example.

Annexe B. Mesh sets, operators and matrices on a simple example

In order to illustrate the definitions for node, element and edge sets, mappings and adjacency operators defined in Section 4 and assembling matrix defined in Section 8.2 we will use the following simple example. Let us consider the mesh shown in Fig. 14.

Node sets

Its nodes subsets and their respective images through the mappings n_L, $L \in \{S, H, \psi\}$ are:

$$\mathcal{P}_h = \{1_h, 2_h, 3_h, 4_h, 5_h, 6_h, 7_h, 8_h, 9_h, 10_h, 11_h\}, \ n_{hn} = 11,$$

$$\mathcal{P}_S = \{1_S, 2_S, 3_S, 4_S\}, \ n_S(\mathcal{P}_S) = \{1_h, 2_h, 3_h, 4_h\}, \ n_{Sn} = 4,$$

$$\mathcal{P}_H = \{1_H, 2_H, 3_H, 4_H\}, \ n_H(\mathcal{P}_H) = \{8_h, 9_h, 10_h, 11_h\}, \ n_{Hn} = 4,$$

$$\mathcal{P}_\psi = \{1_\psi, 2_\psi, 3_\psi\}, \ n_\psi(\mathcal{P}_\psi) = \{5_h, 6_h, 7_h\} \ n_{\psi n} = 3, .$$

Table A.4
Table of notations 2.

Symb.	Meaning
I_L	node index in the corresponding set
J^e	Jacobian of the affine map $\mathcal{N}^e$
J^k	Jacobian of the affine transformation from the ref. element
$N_I,\ I \in \mathcal{P}_h$	standard shape functions
$[A^k]$	assembling matrix.
$[M^k],\ [K^k]$	elementary mass and stiffness matrices
$[N_S^k],\ [N_S^e]$	standard shape functions matrix for T^k and T^e resp.
$[\mathcal{N}^e]$	partition matrices
$[N_{n_q}]$	matrix of baryc. coordinates of int. points on the ref. element
P^e	coordinates of integration points on subelement T^e
$\mathcal{P}_h$	set of mesh nodes
$\mathcal{P}_L$	set of $L-$nodes
$\mathcal{S},\ \mathcal{S}_h$	interface and linearized interface
$\mathcal{T}_L$	sets of standard, $H-$, and $\psi-$elements respect.
$\mathcal{T}_h$	mesh
Q_{n_q}	quadrature rule on the reference element
T^k	mesh element, $1 \leq k \leq n_{he}$
$T^{k\pm}$	mesh element intersection with $\Omega^{\pm}$
T^e	subelement in Ξ^k
S_j^c	subset of interface in effective contact at algorithm iteration j
V_h^X	enriched approximation space
$V_i^k,\ V_i^e$	element T^k (resp. T^e) vertices
$[V^k]$	matrix of coordinates of element T^k vertices
α_δ^{ir}	barycentric coordinates of characteristic points
β_i^k	barycentric coordinates of crack ending point
ϕ, φ	level set functions
$\boldsymbol{\varepsilon}$	linearized strain tensor
(η_p, ξ_p)	quadrature points
Γ_D	Dirichlet boundary
Γ_N	Neumann boundary
Γ_F	fictitious boundary with transmission conditions
Ψ	singular enrichment functions set
Ψ_l	$\Psi-$function with index $1 \leq l \leq 4$
Ω	domain
$\Omega^{\pm}$	domain division according to H sign
$\omega_p,\ \omega_p^e$	quadrature weights
$\boldsymbol{\sigma}$	stress tensor
Ξ^k	partition of mesh element T^k

Inverse mappings

Given the previous definitions, the left inverse mappings applied to each node on $\mathcal{P}_h$ give,

$$n_H^{-1} = \begin{cases} 0 & \text{if} \quad I_h \in \{1_h, 2_h, 3_h, 4_h, 5_h, 6_h, 7_h\}, \\ 1_H & \text{if} \quad I_h = 8_h, \\ 2_H & \text{if} \quad I_h = 9_h, \\ 3_H & \text{if} \quad I_h = 10_h, \\ 4_H & \text{if} \quad I_h = 11_h, \end{cases}$$

$$n_\psi^{-1} = \begin{cases} 0 & \text{if} \quad I_h \in \{1_h, 2_h, 3_h, 4_h, 8_h, 9_h, 10_h, 11_h\}, \\ 1_\psi & \text{if} \quad I_h = 5_h, \\ 2_\psi & \text{if} \quad I_h = 6_h, \\ 3_\psi & \text{if} \quad I_h = 7_h. \end{cases}$$

Element sets

Now, the element sets and their respective images through the inclusions $T_L,\ L \in \{H, S, \psi\}$ are:

$$\mathcal{T}_h = \{\overset{\triangle}{1}_h, \overset{\triangle}{2}_h, \overset{\triangle}{3}_h, \overset{\triangle}{4}_h, \overset{\triangle}{5}_h, \overset{\triangle}{6}_h, \overset{\triangle}{7}_h, \overset{\triangle}{8}_h, \overset{\triangle}{9}_h, \overset{\triangle}{10}_h\},\ n_{he} = 10,\ n_{Se} = 2,\ n_{He} = 2,$$

$$\mathcal{T}_S = \{\overset{\triangle}{1}_S, \overset{\triangle}{2}_S\},\ T_S(\mathcal{T}_S) = \{\overset{\triangle}{1}_h, \overset{\triangle}{2}_h\},\ \mathcal{T}_H = \{\overset{\triangle}{1}_H, \overset{\triangle}{2}_H\}, \quad T_H(\mathcal{T}_H) = \{\overset{\triangle}{9}_h, \overset{\triangle}{10}_h\},$$

$$\mathcal{T}_\psi = \{\overset{\triangle}{1}_\psi, \overset{\triangle}{2}_\psi, \overset{\triangle}{3}_\psi, \overset{\triangle}{4}_\psi, \overset{\triangle}{5}_\psi, \overset{\triangle}{6}_\psi\},\ T_\psi(\mathcal{T}_\psi) = \{\overset{\triangle}{3}_h, \overset{\triangle}{4}_h, \overset{\triangle}{6}_h, \overset{\triangle}{7}_h, \overset{\triangle}{8}_h, \overset{\triangle}{5}_h\},\ n_{\psi e} = 6.$$

Adjacency operators

Next, we will see how the adjacency operators are defined in this case. The adjacency matrices $[C_{hh}]$, $[C_{HH}]$, $[C_{\psi H}]$ and $[C_{\psi\psi}]$ will be given by

$$[C_{hh}] = \begin{bmatrix} 1_h & 1_h & 3_h & 4_h & 6_h & 6_h & 7_h & 7_h & 8_h & 8_h \\ 2_h & 3_h & 6_h & 6_h & 7_h & 8_h & 9_h & 8_h & 10_h & 11_h \\ 3_h & 4_h & 4_h & 5_h & 5_h & 7_h & 5_h & 9_h & 9_h & 10_h \end{bmatrix}, \ [C_{HH}] = \begin{bmatrix} 1_H & 1_H \\ 3_H & 4_H \\ 2_H & 3_H \end{bmatrix},$$

$$[C_{\psi H}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1_H & 2_H & 1_H & 0 \\ 0 & 0 & 0 & 0 & 2_H & 0 \end{bmatrix}, \quad [C_{\psi\psi}] = \begin{bmatrix} 0 & 0 & 2_\psi & 3_\psi & 3_\psi & 2_\psi \\ 2_\psi & 2_\psi & 0 & 0 & 0 & 3_\psi \\ 0 & 1_\psi & 3_\psi & 1_\psi & 0 & 1_\psi \end{bmatrix}.$$

Assembling matrix

Finally, we will see how to build the assembling matrix for this example mesh.

The particular parameters in this case are: $n_{gdf} = 2(11 + 4 + 4 \times 3) = 54$, $n_{edf} = 36$. The global degrees of freedom are:

$$\{u_h\} = \begin{Bmatrix} \{\boldsymbol{u}_1\} \\ \vdots \\ \{\boldsymbol{u}_{11}\} \\ \{\boldsymbol{c}_1\} \\ \vdots \\ \{\boldsymbol{c}_4\} \\ \{\boldsymbol{b}_1\} \\ \vdots \\ \{\boldsymbol{b}_3\} \end{Bmatrix}. \tag{47}$$

We focus now on the sixth element on $\mathcal{T}_h$, which is the third element on $\mathcal{T}_\psi$, this is, $T_\psi\left(\overset{\triangle}{3}_\psi\right) = \overset{\triangle}{6}_h$. The set $\mathcal{P}_\psi$ consists of the nodes $\{5_h, 6_h, 7_h\}$ and the vertices of element $\overset{\triangle}{6}_h$ are $\{6_h, 8_h, 7_h\}$. Then,

$$[C_{hh}]_{:6} = \begin{pmatrix} 6_h \\ 8_h \\ 7_h \end{pmatrix}, \ [C_{\psi\psi}]_{:3} = \begin{pmatrix} 2_\psi \\ 0 \\ 3_\psi \end{pmatrix}, \ [C_{\psi H}]_{:3} = \begin{pmatrix} 0 \\ 1_H \\ 0 \end{pmatrix}.$$

Now if we consider the third elementary degree of freedom (first component associated to the standard shape function of the third vertex of the element $\overset{\triangle}{6}_h$, see (25)), $j = 3$, we get from (41)–(45):

$$s(j)=1, \ i(j)=3, \ l(j)=0, \ I(j,\overset{\triangle}{6}_h)=[C_{hh}]_{3,6}=7_h, \ \xi_S^{\overset{\triangle}{6}_h}(j)=\mathsf{p}(1,7) = 13.$$

Indeed,

$$\xi_S^{\overset{\triangle}{6}_h} = (11, 15, 13, 12, 16, 14)^t,$$

which corresponds to the contribution that the standard degrees of freedom of this element make to the vector of global degrees of freedom given in (47). Summarizing, it results the following assembling matrix:

$$[A_S^{\overset{\triangle}{6}_h}] = \begin{array}{c} \begin{array}{cccccccccc} 1 & \cdots & 11 & 12 & 13 & 14 & 15 & 16 & \cdots & n_{gdf} \end{array} \\ \left[\begin{array}{cccccccccc} 0 & \cdots & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \end{array}\right] \end{array}.$$

In a similar way one can build the matrices $[A_H^{\overset{\triangle}{6}_h}]$ and $[A_\psi^{\overset{\triangle}{6}_h}]$ by using

$$\xi_H^{\overset{\triangle}{6}_h} = (0, 23, 0, 0, 24, 0)^t,$$

and

$$\xi_\psi^{\overset{\triangle}{6}_h} = (39, 41, 43, 45, 0, 0, 0, 0, 47, 49, 51, 53, 40, 42, 44, 46, 0, 0, 0, 0, 48, 50, 52, 54)^t.$$

For example, in order to compute $\xi_H^6(2)$ we need $s(8) = 1$ and $I(8, 6) = n_H^{-1}([C_{hh}]_{2,6}) = n_h^{-1}(8_h) = 1_H$, so

$$\xi_H^6(2) = \mathsf{p}_H(s(8), I(8, 6)) = \mathsf{p}_H(1, 1_H) = 1 + 2n_{hn} = 23,$$

while

$$\xi_H^6(4) = \mathsf{p}_H(s(10), I(10,6)) = \mathsf{p}_H(2,0) = 0,$$

since $s(10) = 2$ and $I(10,6) = n_H^{-1}([C_{hh}]_{1,6}) = n_h^{-1}(6_h) = 0$. Also,

$$\xi_\psi^6(4) = \mathsf{p}_\psi(s(16), l(16), I(16,6)) = \mathsf{p}_\psi(1,4,2_\psi) = 1 + 2(4-1+4(2-1)+11+4) = 45,$$

since $s(16) = 1$, $l(16) = 4$ and $I(16,6) = n_\psi^{-1}([C_{hh}]_{1,6}) = n_\psi^{-1}(6_h) = 2_\psi$, and

$$\xi_\psi^6(5) = \mathsf{p}_\psi(s(17), l(17), I(17,6)) = \mathsf{p}_\psi(1,1,0) = 0,$$

since $s(17) = 1$, $l(17) = 1$ and $I(17,6) = n_\psi^{-1}([C_{hh}]_{2,6}) = n_\psi^{-1}(8_h) = 0$.

References

[1] A. Khoei, Extended Finite Element Method: Theory and Applications. Vol. 9781118457689, John Wiley & Sons, Ltd, 2015, pp. 1–565.

[2] T.P. Fries, The intrinsic XFEM for two-fluid flows, Internat. J. Numer. Methods Fluids 60 (4) (2009) 437–471.

[3] T.-P. Fries, T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications, Internat. J. Numer. Methods Engrg. 84 (3) (2010) 253–304.

[4] J. Chessa, H. Wang, T. Belytschko, On the construction of blending elements for local partition of unity enriched finite elements, Internat. J. Numer. Methods Engrg. 57 (7) (2003) 1015–1038.

[5] S. Bordas, P.V. Nguyen, C. Dunant, A. Guidoum, H. Nguyen-Dang, An extended finite element library, Internat. J. Numer. Methods Engrg. 71 (6) (2007) 703–732.

[6] I. Nistor, O. Pantalé, S. Caperaa, Numerical implementation of the extended finite element method for dynamic crack analysis, Adv. Eng. Softw. 39 (7) (2008) 573–587.

[7] N. Sukumar, J.-H. Prévost, Modeling quasi-static crack growth with the extended finite element method part I: Computer implementation, Int. J. Solids Struct. 40 (26) (2003) 7513–7537.

[8] E. Giner, N. Sukumar, J. Tarancón, F. Fuenmayor, An Abaqus implementation of the extended finite element method, Eng. Fract. Mech. 76 (3) (2009) 347–368.

[9] H. Talebi, M. Silani, S. Bordas, P. Kerfriden, T. Rabczuk, A computational library for multiscale modeling of material failure, Comput. Mech. 53 (5) (2014) 1047–1071.

[10] A. Jafari, P. Broumand, M. Vahab, N. Khalili, An eXtended finite element method implementation in COMSOL multiphysics: Solid mechanics, Finite Elem. Anal. Des. 202 (2022) Cited By :4.

[11] Y. Liang, Z. Gao, E.V. Iarve, An implementation of regularized extended finite element method in abaqus/explicit, in: AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2022, 2022.

[12] A. Khoei, B. Bahmani, Application of an enriched FEM technique in thermo-mechanical contact problems, Comput. Mech. 62 (5) (2018) 1127–1154.

[13] D. Mueller-Hoeppe, P. Wriggers, S. Loehnert, Crack face contact for a hexahedral-based XFEM formulation, Comput. Mech. 49 (6) (2012) 725–734.

[14] J. Dolbow, N. Moës, T. Belytschko, An extended finite element method for modeling crack growth with frictional contact, Comput. Methods Appl. Mech. Engrg. 190 (51) (2001) 6825–6846.

[15] S. Bhattacharya, I. Singh, B. Mishra, T. Bui, Fatigue crack growth simulations of interfacial cracks in bi-layered FGMs using XFEM, Comput. Mech. 52 (4) (2013) 799–814.

[16] H. Sauerland, T.-P. Fries, The extended finite element method for two-phase and free-surface flows: A systematic study, J. Comput. Phys. 230 (9) (2011) 3369–3390.

[17] S. Zlotnik, P. Díez, Hierarchical X-FEM for n-phase flow (n>2), Comput. Methods Appl. Mech. Engrg. 198 (30) (2009) 2329–2338.

[18] F. Alauzet, B. Fabrèges, M. Fernández, M. Landajuela, Nitsche-XFEM for the coupling of an incompressible fluid with immersed thin-walled structures, Comput. Methods Appl. Mech. Engrg. 301 (2016) 300–335.

[19] M.A. Fernández, F.M. Gerosa, An unfitted mesh semi-implicit coupling scheme for fluid-structure interaction with immersed solids, Internat. J. Numer. Methods Engrg. 122 (19) (2021) 5384–5408, Cited By :2.

[20] M. Martinolli, J. Biasetti, S. Zonca, L. Polverelli, C. Vergara, Extended finite element method for fluid-structure interaction in wave membrane blood pump, Int. J. Numer. Methods Biomed. Eng. 37 (7) (2021) Cited By :3.

[21] S. Zonca, C. Vergara, L. Formaggia, An unfitted formulation for the interaction of an incompressible fluid with a thick structure via an XFEM/DG approach, SIAM J. Sci. Comput. 40 (1) (2018) B59–B84.

[22] J. Brezina, P. Exner, Extended finite element method in mixed-hybrid model of singular groundwater flow, Math. Comput. Simulation 189 (2021) 207–236, MATCOM Special Issue: Modelling 2019: The 6th International Conference on Mathematical Modelling and Computational Methods in Applied Sciences and Engineering.

[23] B. Flemisch, A. Fumagalli, A. Scotti, A review of the XFEM-based approximation of flow in fractured porous media, in: SEMA SIMAI Springer Series, vol.12, 2016, pp. 47–76.

[24] S. Wu, O. Dazel, G. Legrain, A computational approach based on extended finite element method for thin porous layers in acoustic problems, Internat. J. Numer. Methods Engrg. 123 (18) (2022) 4209–4243.

[25] L. Bailly-Salins, L. Borrel, W. Jiang, B.W. Spencer, K. Shirvan, A. Couet, Modeling of high-temperature corrosion of Zirconium alloys using the extended finite element method (X-FEM), Corros. Sci. 189 (2021) Cited By :1.

[26] J.R. Craig, R. Gracie, Using the extended finite element method for simulation of transient well leakage in multilayer aquifers, Adv. Water Resour. 34 (9) (2011) 1207–1214, Cited By :4.

[27] M. Cao-Rial, C. Moreno, P. Quintela, A new methodology for element partition and integration procedures for XFEM, Finite Elem. Anal. Des. 113 (2016) 1–13.

[28] M.T. Cao, C. Moreno, P. Quintela, Simulation of Rayleigh waves in cracked plates, Math. Methods Appl. Sci. 30 (1) (2007) 15–42.

[29] A. Bermúdez, C. Moreno, Duality methods for solving variational inequalities, Comput. Math. Appl. 7 (1) (1981) 43–58.

[30] Á. Rodríguez-Arós, M. Cao-Rial, M. Sofonea, A contact model for piezoelectric beams, IFIP Adv. Inf. Commun. Technol. 494 (2016) 442–451.

[31] Á. Rodríguez-Arós, M. Cao-Rial, Mathematical and numerical analysis of an obstacle problem for elastic piezoelectric beams, Math. Mech. Solids. 23 (3) (2018) 262–278.

[32] N. Moës, A. Gravouil, T. Belytschko, Non-planar 3D crack growth by the extended finite element and level sets-part I: Mechanical model, Internat. J. Numer. Methods Engrg. 53 (11) (2002) 2549–2568.

[33] M.T. Cao-Rial, Problemas de contacto en elasticidad dinámica con XFEM (Ph.D. thesis), Universidade de Santiago de Compostela, 2011.

[34] T. Belytschko, T. Black, Elastic crack growth in finite elements with minimal remeshing, Internat. J. Numer. Methods Engrg. 45 (5) (1999) 601–620.

[35] J. Dolbow, N. Moës, T. Belytschko, Discontinuous enrichment in finite elements with a partition of unity method, Finite Elem. Anal. Des. 36 (3) (2000) 235–260.

[36] M. Fleming, Y.A. Chu, B. Moran, T. Belytschko, Enriched element-free Galerkin methods for crack tip fields, Internat. J. Numer. Methods Engrg. 40 (8) (1997) 1483–1504.

[37] S. Natarajan, D. Roy Mahapatra, S. Bordas, Integrating strong and weak discontinuities without integration subcells and example applications in an XFEM/GFEM framework, Internat. J. Numer. Methods Engrg. 83 (3) (2010) 269–294.

[38] Y. Sudhakar, W. Wall, Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods, Comput. Methods Appl. Mech. Engrg. 258 (2013) 39–54.

[39] L. Formaggia, C. Vergara, S. Zonca, Unfitted extended finite elements for composite grids, Comput. Math. Appl. 76 (4) (2018) 893–904.

[40] A. Cano, C. Moreno, A new method for numerical integration of singular functions on the plane, Numer. Algorithms 68 (3) (2015) 547–568.

[41] A. Cano, C. Moreno, Transformation methods for the numerical integration of three-dimensional singular functions, J. Sci. Comput. 71 (2) (2017) 571–593.

[42] T.J. Hughes, The Finite Element Method. Linear Static and Dynamic Finite Element Analysis, Prentice-Hall, 1987.