

A stable discontinuity-enriched finite element method for 3-D problems containing weak and strong discontinuities

Jian Zhang, Sanne J. van den Boom, Fred van Keulen, Alejandro M. Aragón*

Department of Precision and Microsystems Engineering, Delft University of Technology, Delft, The Netherlands

Received 22 December 2018; received in revised form 29 April 2019; accepted 16 May 2019

Available online 24 May 2019

Highlights

- The Discontinuity-Enriched Finite Element Method is introduced to solve 3-D problems;
- A geometric engine is proposed to handle arbitrary (intersecting) discontinuities;
- Intersecting discontinuities are resolved by hierarchical enrichment functions;
- We show the methodology's convergence and ability to extract stress intensity factors;
- We demonstrate the method's stability with the use of a simple preconditioner.

Abstract

A new enriched finite element technique, named the Discontinuity-Enriched Finite Element Method (DE-FEM), was introduced recently for solving problems with both weak and strong discontinuities in 2-D. In this mesh-independent procedure, enriched degrees of freedom are added to new nodes collocated at the intersections between discontinuities and the sides of finite elements of the background mesh. In this work we extend DE-FEM to 3-D and describe in detail the implementation of a geometric engine capable of handling interactions between discontinuities and the background mesh. Several numerical examples in linear elastic fracture mechanics demonstrate the capability and performance of DE-FEM in handling discontinuities in a fully mesh-independent manner. We compare convergence properties and the ability to extract stress intensity factors with standard FEM. Most importantly, we show DE-FEM provides a stable formulation with regard to the condition number of the resulting system stiffness matrix.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Keywords: Enriched finite element methods; DE-FEM; Strong discontinuities; Fracture mechanics; GFEM; XFEM

1. Introduction

Enriched finite element methods have fundamentally changed the modeling of problems containing discontinuities. By means of enrichment functions that incorporate the required field jumps, these methods are able to completely decouple the geometrical description of discontinuities from the underlying finite element (FE) discretization, eliminating the need for creating *matching* or *discontinuity-conforming* meshes. The Discontinuity-Enriched Finite Element Method (DE-FEM), recently proposed by Aragón and Simone for 2-D problems [1], uses

* Corresponding author.

E-mail address: a.m.aragon@tudelft.nl (A.M. Aragón).

a single formulation to model problems containing both weak and strong discontinuities (C^{-1} -continuous field) by placing enriched degrees of freedom (DOFs) *only* to nodes created along discontinuities. This new versatile procedure, which has several advantages over other commonly used enriched formulations, is demonstrated in this manuscript for 3-D problems in elastostatics.

Standard FEM can be used to model problems with discontinuities, but it requires a matching discretization where the sides of elements align to material interfaces or cracks. When dealing with fracture problems, mesh adaptivity in the neighborhood of crack tips is usually needed to improve the FE approximation at the expense of a more complex mesh creation. Alternatively, the stress singularity can be captured by using special elements arranged concentrically at the crack tip, *e.g.*, quadrilateral finite elements where two nodes in parametric space map to the crack tip location in physical space [2], or quarter-point finite elements, where the mid-size nodes of quadratic Lagrange finite elements are moved closer to the crack tip [3]. Although these special meshes can capture the stress singularity, generating them is a time-consuming process. Creating matching meshes in general can be quite challenging depending on the morphology of the problem, particularly because of the strict conditions on mesh quality that are required—elements with bad aspect ratios could reduce the approximation accuracy [4]; moreover, robustness is still questionable, especially for 3-D FE meshes, as issues remain in generating meshes that correctly match boundaries [5]. Recently, numerical techniques such as Universal Meshes [6] and the Conforming to Interface Structured Adaptive Mesh Refinement (CISAMR) [7] have been proposed to modify meshes locally to discontinuities. These methods can therefore guarantee adequate discretizations for the analysis by ensuring elements with proper aspect ratios—which are crucial for recovering accurate gradient fields—while avoiding the need for global remeshing.

The eXtended/Generalized Finite Element Method (X/GFEM) [8–10] solves the aforementioned shortcomings in an elegant manner by enriching the approximation space with functions that can reproduce the discontinuities [11,12]. In X/GFEM, the discretization is then decoupled from discontinuities by detecting cut elements, and by adding enriched DOFs to the standard mesh nodes of cut elements so that discontinuities can be resolved by the formulation. Although X/GFEM provides great flexibility in the choice of underlying FE discretizations, many properties that are usually taken for granted in standard FEM are lost, including the physical meaning of DOFs (unless shifting is used [13]), the straightforward imposition of essential (Dirichlet) boundary conditions [14,15], and most importantly, the stability—here understood as the condition number of the resulting system matrix, which in standard FEM scales with mesh size h as $\mathcal{O}(h^{-2})$. Indeed, several enrichment functions could result in an unstable formulation where the condition number of the system matrix grows much faster than that of standard FEM. Pursuing Stable GFEM (SGFEM) that is devoid of this issue has been the focus of recent research in the field [16], with SGFEM proposed for weak discontinuities in [17,18] and for strong discontinuities in [19–21]. Moreover, depending on the type of enrichment functions used, special techniques are needed in X/GFEM to prevent the loss of accuracy in *blending elements*, *i.e.*, non-cut elements that share enriched DOFs [22–24]. Finally, the computer implementation of X/GFEM is also more involved than that of standard FEM [25] since a variable number of DOFs per mesh node is required, and quadrature rules used depend on the type of enrichment functions—for fracture problems, for instance, elements with singular enrichments require special integration techniques [26].

Within the realm of enriched formulations, the Discontinuity-Enriched Finite Element Method presents a new paradigm for the mesh-independent analysis of weak and strong discontinuities [1]. DE-FEM can be seen as a technique that combines X/GFEM's most salient mesh-independent property—by decoupling the FE discretization from the problem's complex geometric features—while keeping the attractive properties of standard FEM. Indeed, in DE-FEM enrichment functions vanish at mesh nodes by construction, and as a consequence DOFs associated to the latter keep their physical interpretation, *e.g.*, displacement at the node location. In addition, essential boundary conditions can be prescribed directly as in standard FEM—both standard and enriched DOFs [27]. The computer implementation is also straightforward since the new enriched nodes are added to the same data structure used to store the original mesh nodes, and their corresponding enriched DOFs are retrieved from the solution vector in the same way as standard DOFs. Because DE-FEM was designed for simplicity in the formulation (and thus in the implementation), enrichment functions that capture the stress singularities along crack fronts are not used (although they may also be added). As a result, convergence rates for singular problems are not optimal.

In the absence of strong discontinuities, DE-FEM simplifies to the Interface-enriched Generalized Finite Element Method (IGFEM) [28] or to the Hierarchical Interface-enriched Finite Element Method (HIFEM) [29], which were devised for weak discontinuities alone [30–36]. HIFEM builds on IGFEM to resolve multiple weak discontinuities

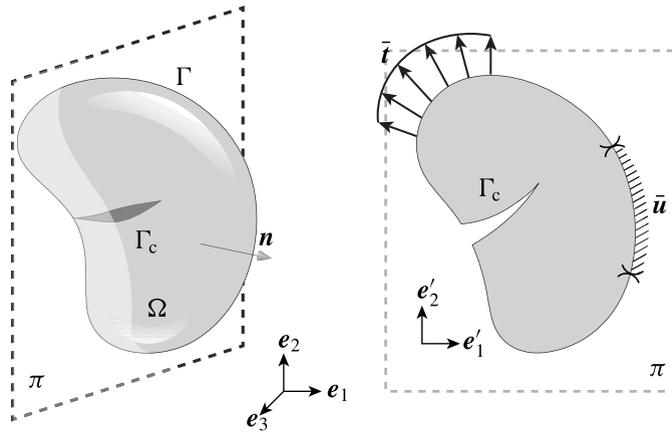


Fig. 1. A 3-D body Ω containing a traction-free crack Γ_c (left); Prescribed Dirichlet and Neumann boundary conditions, \bar{u} on Γ_u and \bar{t} on Γ_t , respectively, are visualized on a plane π that slices the domain with local coordinates $\{e'_i\}$ (right).

within a single element—and even intersecting discontinuities—by means of a hierarchical construction of enrichment functions. Given the comprehensive literature on the study of weak discontinuities with IGFEM/HIFEM, here we devote ourselves to problems containing strong discontinuities and to their interactions with weak discontinuities.

DE-FEM was first proposed for analyzing 2-D problems in elastostatics. In this paper we extend DE-FEM to 3-D, and we discuss in detail its computer implementation in a displacement-based finite element code. In particular, we discuss algorithmic considerations for a 3-D geometric engine that handles the interactions between the background (original) mesh and material interfaces and/or cracks. By means of a discontinuous patch test, we demonstrate DE-FEM can generate two independent kinematic fields as long as proper care is taken in the construction of a *conforming integration mesh*. A convergence study then demonstrates the accuracy and convergence rates of the method, which is then used to extract stress intensity factors (SIFs). Most importantly, we show that DE-FEM is a stable formulation, *i.e.*, the condition number of the stiffness matrix grows at roughly the same rate as that of standard FEM. Finally, we showcase DE-FEM in the challenging problem of intersecting weak and strong discontinuities, which is handled effortlessly by means of the hierarchical implementation *à la* HIFEM.

2. Formulation

Consider the elastostatics boundary value problem for a cracked body $\Omega \subset \mathbb{R}^3$ with closure $\bar{\Omega}$ and boundary $\partial\Omega \equiv \Gamma = \bar{\Omega} \setminus \Omega$. The body is subjected to a non-homogeneous essential boundary condition (BC) $\bar{u} : \Gamma_u \rightarrow \mathbb{R}^3$ and to a traction BC $\bar{t} : \Gamma_t \rightarrow \mathbb{R}^3$, the latter having a zero value along a crack $\Gamma_c \subset \Gamma_t$, as shown in Fig. 1. The abstract form of the weak (variational) formulation is: Given the displacement field $u = v + \tilde{u}$, find $v \in \mathcal{V}_0(\Omega)$ such that

$$B(v, w) = L(w) - B(\tilde{u}, w) \quad \forall w \in \mathcal{V}_0, \quad (1)$$

where $L(\cdot)$ and $B(\cdot)$ are the linear and bilinear forms, respectively, \tilde{u} is a vector-valued function that satisfies $\tilde{u}|_{\Gamma_u} = \bar{u}$, and \mathcal{V}_0 is the vector-valued function space with components in $\mathcal{H}_0^1(\Omega)$ (Sobolev space that satisfies homogeneous essential boundary conditions on Γ_u). For elastostatics, the linear and bilinear forms are given, respectively, by

$$L(w) = \int_{\Omega} b \cdot w \, d\Omega + \int_{\Gamma_t} \bar{t} \cdot w \, d\Gamma \quad (2)$$

and

$$B(v, w) = \int_{\Omega} \sigma(v) : \epsilon(w) \, d\Omega. \quad (3)$$

In (2), $b : \Omega \rightarrow \mathbb{R}^3$ is the body force, and in (3) the stress tensor $\sigma : \bar{\Omega} \rightarrow \mathbb{R}^3 \times \mathbb{R}^3$ obeys Hooke's law on the linearized strain, *i.e.*, $\sigma = C\epsilon$, with $\epsilon(v) = \frac{1}{2}(\nabla v + \nabla v^T)$.

In order to solve the finite-dimensional form of (1), the domain is discretized into finite elements e_i (simplexes in particular) such that $\cup_i \bar{e}_i = \bar{\Omega}^h \approx \bar{\Omega}$. Following a Bubnov-Galerkin approach, both the trial solution and the weight function are chosen from the DE-FEM space:

$$\mathcal{S}_e^h = \left\{ \mathbf{v}^h(\mathbf{x}) \mid \mathbf{v}^h(\mathbf{x}) = \underbrace{\sum_{i \in \iota_h} N_i(\mathbf{x}) \mathbf{U}_i}_{\text{standard FEM}} + \underbrace{\sum_{i \in \iota_w} \psi_i(\mathbf{x}) \boldsymbol{\alpha}_i}_{\text{weak}} + \underbrace{\sum_{i \in \iota_s} \chi_i(\mathbf{x}) \boldsymbol{\beta}_i}_{\text{strong}} \right\}, \quad \mathbf{U}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i \in \mathbb{R}^3, \quad (4)$$

where the standard FE space (first term) is augmented by an enriched space with special enrichment functions that can capture the jump in the gradient (weak term) and the primal (strong term) fields. In the standard FEM term, ι_h is the index set containing all original mesh nodes; N_i denotes the i th Lagrange shape function and \mathbf{U}_i are the corresponding standard DOFs. In the enriched term, ι_w and ι_s are index sets of enriched nodes created along discontinuities. For weak discontinuities, these nodes are associated with enrichment functions ψ_i and corresponding enriched DOFs $\boldsymbol{\alpha}_i$. In addition to these, strong discontinuities add a new set of nodes associated with enrichment functions χ_i and enriched DOFs $\boldsymbol{\beta}_i$.

In order to have an arbitrary number of discontinuities interact with a single element, we follow the work of Soghrati [29]. The procedure is schematically shown in Fig. 2, where an element e_i is split by two discontinuities. The first discontinuity creates a first level of integration subdomains, and these in turn become the parent elements for a second discontinuity. These relationships are stored in an ordered tree data structure, which is used during numerical quadrature to build the enrichment functions in a hierarchical manner. Therefore, for the e th cut element, the approximation $\mathbf{u}^h \in \mathcal{S}_e^h$ can be written as

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i \in \iota_h} N_i(\mathbf{x}) \mathbf{U}_i + \sum_{k \in h} \sum_{i \in \iota_w} \psi_{ki}(\mathbf{x}) \boldsymbol{\alpha}_{ki} + \sum_{k \in h_s} \sum_{i \in \iota_s} \chi_{ki}(\mathbf{x}) \boldsymbol{\beta}_{ki}, \quad (5)$$

where $h = \{1, \dots, D\}$ is the index set used to represent D discontinuities (both weak and strong) that interact with the element, and $h_s \subseteq h$ only includes the indices for strong discontinuities. The formulation in uncut elements remains the same as in standard FEM, i.e., only the first term in (5) is used.

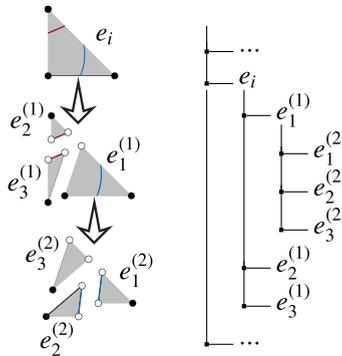


Fig. 2. Schematic representation of the hierarchical creation of integration subdomains for an element e_i split by two discontinuities ($D = 2$). The hierarchical information is stored in an ordered tree that is also used for constructing enrichment functions during the numerical quadrature of local element arrays.

For the sake of simplicity, we denote by $\boldsymbol{\varphi} \equiv [N_1 \dots \psi_1 \dots \chi_1 \dots] = [N \ \boldsymbol{\psi} \ \boldsymbol{\chi}]$ a vector that stacks shape and enrichment functions that are non-zero on the e th element and by $\mathbf{B} \equiv \boldsymbol{\partial} \boldsymbol{\varphi} = [\boldsymbol{\partial} N \ \boldsymbol{\partial} \boldsymbol{\psi} \ \boldsymbol{\partial} \boldsymbol{\chi}]$ the strain–displacement matrix, which is obtained by applying to $\boldsymbol{\varphi}$ the differential operator $\boldsymbol{\partial}$:

$$\boldsymbol{\partial} \equiv \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}^T. \quad (6)$$

The element local stiffness matrix and the nodal load vector are given, respectively, by

$$k_e = \int_e \mathbf{B}^\top \mathbf{C} \mathbf{B} \, de \quad \text{and} \quad \mathbf{f}_e = \int_e \boldsymbol{\varphi}^\top \mathbf{b} \, de + \int_{\partial e} \boldsymbol{\varphi}^\top \bar{\mathbf{t}} \, d\partial e, \tag{7}$$

where \mathbf{C} is the constitutive matrix given by

$$\mathbf{C} = \begin{bmatrix} \lambda \mathbb{1} + \mu \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mu \mathbf{I} \end{bmatrix}, \tag{8}$$

λ and μ are the Lamé constants, and $\mathbb{1}$ and \mathbf{I} are the unity and identity matrices, respectively. Details about the construction of k_e and \mathbf{f}_e , together with pseudo-code for constructing the enrichment functions in a hierarchical manner, are given later in Section 4.

The finite-dimensional form obtained by taking into account all finite elements

$$\sum_{e \in \Omega^h} B_e(\mathbf{v}^h, \mathbf{w}^h) = \sum_{e \in \Omega^h} L_e(\mathbf{w}^h) - \sum_{e \in \Omega^h} B_e(\tilde{\mathbf{u}}^h, \mathbf{w}^h) \quad \forall \mathbf{w}^h \in \mathcal{S}_e^h, \tag{9}$$

leads to an augmented system of linear equations

$$\underbrace{\begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{u\alpha} & \mathbf{K}_{u\beta} \\ \mathbf{K}_{u\alpha}^\top & \mathbf{K}_{\alpha\alpha} & \mathbf{K}_{\alpha\beta} \\ \mathbf{K}_{u\beta}^\top & \mathbf{K}_{\alpha\beta}^\top & \mathbf{K}_{\beta\beta} \end{bmatrix}}_K \underbrace{\begin{bmatrix} \mathbf{U} \\ \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}}_F = \underbrace{\begin{bmatrix} \mathbf{F}_u \\ \mathbf{F}_\alpha \\ \mathbf{F}_\beta \end{bmatrix}}_F. \tag{10}$$

In (10) we made explicit the partition of global arrays into standard and enriched parts. Aragón and Simone [1] discuss the formulation in more detail for 2-D elastostatics, considering also the case of cohesive tractions acting on cracks.

3. Geometric engine

The complexity of mesh generation in standard FEM is transferred in DE-FEM to a geometric engine that takes care of the interaction between a background mesh (usually structured) and the discontinuities. The engine needs to be designed for handling complex scenarios with multiple discontinuities, as shown for instance in Fig. 3, and to handle not only cracks (strong discontinuities) but also material interfaces (weak discontinuities).

The operations of the geometric engine are summarized in the flow chart of Fig. 4. The engine takes as input the background mesh and the set of discontinuities. In this work, FE models are discretized using linear tetrahedra, cracks are described explicitly through planar regions (analogous to line segments in 2-D), and material interfaces are described implicitly through level set functions.

The engine first loops over discontinuities and then loops over mesh elements for each discontinuity. Tetrahedral elements cut by the first discontinuity are detected. The spatial locations of new enriched nodes is then determined by performing intersection tests between the discontinuity and element sides. At these locations we create the corresponding number of enriched nodes depending on the type of discontinuity; one enriched node per intersection (associated with weak DOFs) is generated if the discontinuity is weak. If the discontinuity is strong, two enriched nodes are created per intersection (each associated with either weak or strong DOFs) unless the node is at the crack

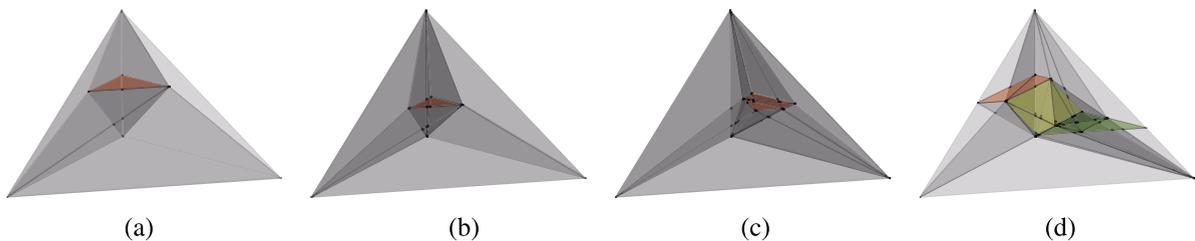


Fig. 3. Configurations considered when designing the geometric engine include (a) partially cut, (b) internal, (c) irregular, and (d) multiple discontinuities.

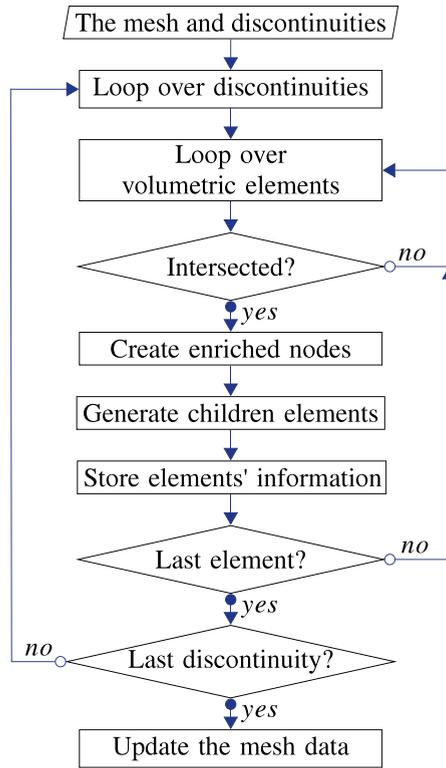


Fig. 4. Geometric engine’s procedure for handling models with multiple discontinuities hierarchically.

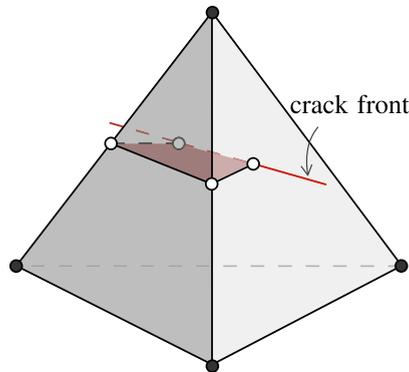


Fig. 5. Nodes along the crack front are generated when a linear tetrahedral element is split by a crack partially.

front (associated with weak enriched DOFs only). These enriched nodes, together with those of the original mesh, are later used to create tetrahedral integration elements.

It is worth noting that the engine described here would not be efficient when dealing with a great number of discontinuities. Conducting intersection tests for every element-discontinuity pair leads to the so called *all-pairs weakness*, which results in an algorithm of complexity $\mathcal{O}(DE)$ for E mesh elements and D discontinuities. The efficiency of the interaction can be greatly improved by performing heuristics conducted on bounding boxes. Furthermore, space partitioning techniques can speedup finding elements intersected by discontinuities. Among these one could use octrees, k -d trees, R^* -trees, and grids [37]. Finally, these techniques could be supplemented

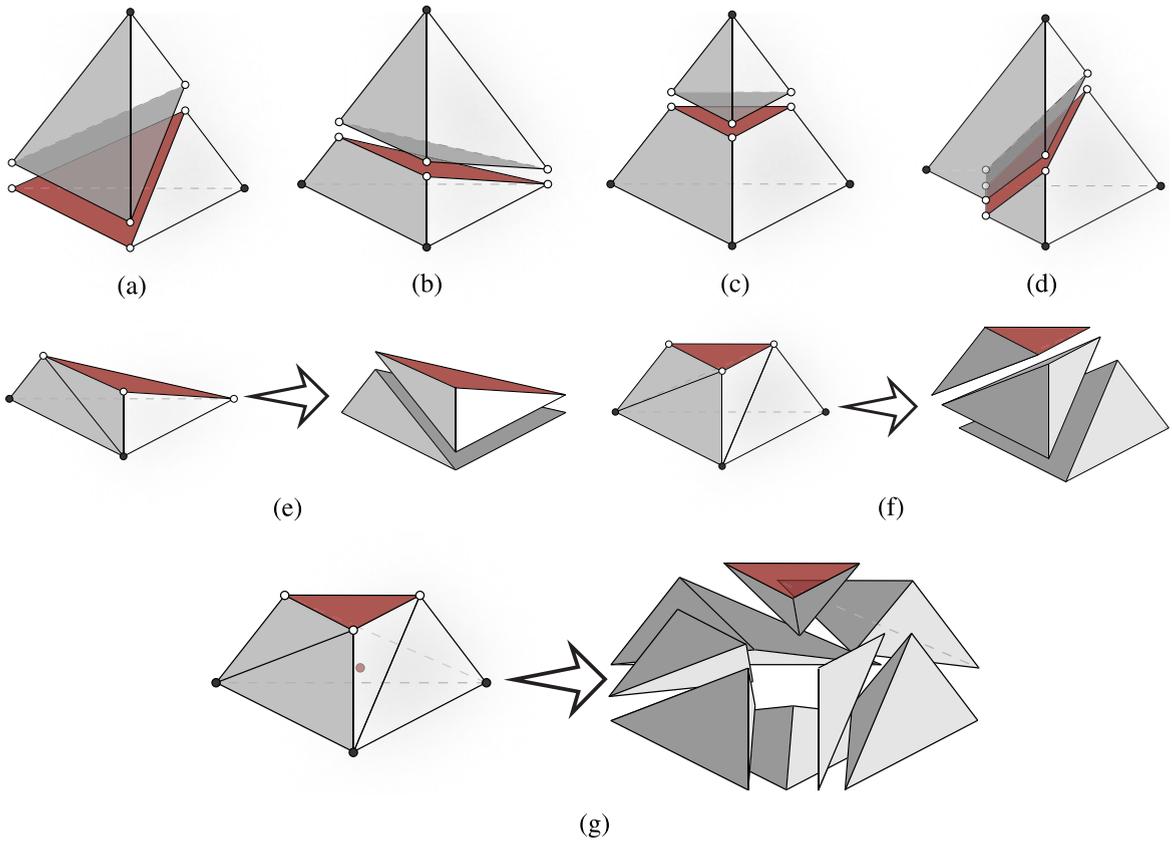


Fig. 6. Configurations that result from splitting completely a tetrahedron: (a) Two tetrahedra; (b) One tetrahedron and one pyramid; (c) One tetrahedron and one prism; and (d) Two prisms. One of possible situations of splitting (e) a pyramid into two tetrahedra and (f) a prism into three tetrahedra. (g) An untetrahedralizable case for subdividing a prism, which requires an extra vertex (called Steiner point) to create eight tetrahedra.

with fast marching methods and graph traversal algorithms to look only for neighboring elements of already cut elements.

A crucial task of the geometric engine is to determine the location of crack fronts. Fig. 5 shows that a planar region, which represents a 3-D crack, intersects with a tetrahedral element partially. In order to determine the location of nodes along the crack front, intersections tests are performed between the crack and all faces of the tetrahedron. Numerical precision is of major concern when dealing with floating point arithmetic, especially for determining whether a point lies on an edge or on a face. In this work we adopt a tolerance that depends on the mesh size h for finding intersections.

Other operations performed by the geometric engine include creating groups of nodes and elements that are associated with discontinuities (the latter are used for assigning the correct material properties), generating integration elements, and updating the original mesh data structure.

3.1. Creation of integration elements

Although it is not strictly necessary, creating an *integration tetrahedral mesh* is convenient for four reasons: (i) Lagrange shape functions in integration tetrahedra are used to construct the enrichment functions. Storing integration elements in an ordered tree data structure further facilitates the construction of enrichment functions at each level of hierarchy—the resulting recursive quadrature algorithm is general and its computer implementation is straightforward; (ii) By creating integration elements at either side of the discontinuities we ensure the enrichment functions are smooth, and thus they can be integrated with the minimum number of integration points (parent

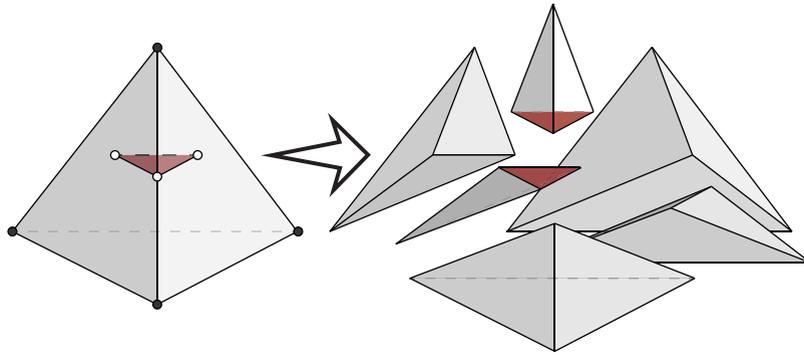


Fig. 7. Example of a partially split tetrahedron, where six new tetrahedra are created.

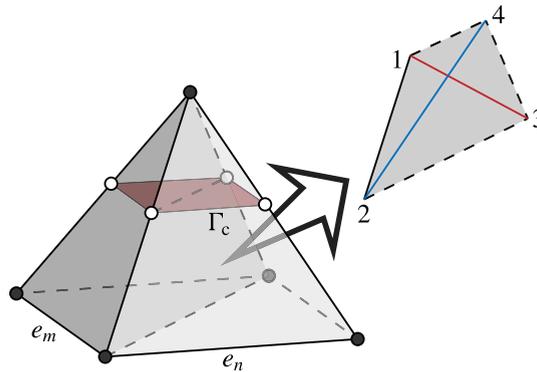


Fig. 8. Two contiguous elements e_m and e_n are split by a discontinuity Γ_c . By not taking proper care, it is possible that a *non-conforming integration mesh* below the discontinuity is created, whereby integration tetrahedra at one side of surface (1,2,3,4) have faces (1, 2, 4) and (2, 3, 4), and at the other side (1, 2, 3) and (1, 3, 4).

shape and enrichment functions are linear, and thus their corresponding derivatives are constant); (iii) Tetrahedral elements are the simplest 3-D elements to split, so they ease the handling of multiple discontinuities within a single parent mesh element; and (iv) Integration elements help in the post-processing stage so that results can be visualized correctly. It is worth noting that the creation of this *integration mesh* is a procedure bound locally to discontinuities, and as such it is completely different than performing remeshing.

Efficient algorithms for splitting a tetrahedron is one of our main concerns. There are four distinct configurations that result from completely splitting a tetrahedron with a planar region [34], as shown in Figs. 6(a)–6(d): (a) two tetrahedra; (b) a tetrahedron and a pyramid; (c) a tetrahedron and a prism; and (d) two prisms. A pyramid can be considered to be composed by two tetrahedra (Fig. 6(e)). Therefore, we can get three new tetrahedral elements in case (b). Normally, tetrahedralizing a prism (Fig. 6(f)) can produce three tetrahedra under the premise that any two diagonals of three quadrilateral sides share the same vertex. However, in the case that the diagonals of the body are already prescribed (for example by the fact that neighboring elements have been tetrahedralized), it might be impossible to create new tetrahedra directly. This situation is alleviated by adding an extra vertex (called Steiner point) inside the prism (see Fig. 6(g)). Then, this prism can be split into eight tetrahedra that share this vertex.

While fully splitting a tetrahedron into several tetrahedra can be accomplished by using basic splitting rules [38], the complexity of the geometric operations increases rapidly when dealing with partial splits (see Fig. 7) or when a crack resides completely inside a tetrahedron. The latter situation could be alleviated by refining the mesh, but this in turn increases the computational cost. Hence, in this work we use constrained Delaunay tetrahedralization for creating new tetrahedral sub-domains [39].

Since each background element intersected by discontinuities is tetrahedralized in sequence, the *constrained Delaunay* algorithm is needed to avoid the creation of a *non-conforming integration mesh*, which as explained later has important consequences. Fig. 8 shows such a case, where two contiguous tetrahedral elements split by

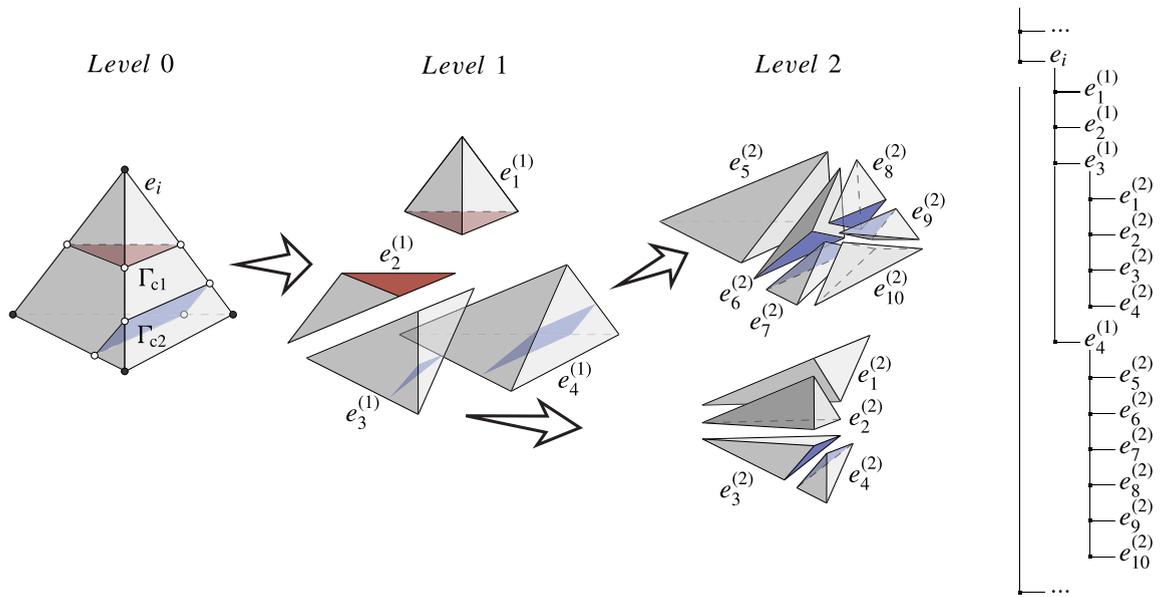


Fig. 9. Mesh tetrahedral element e_i intersected first by Γ_{c1} (which creates the first level of integration elements) and then by Γ_{c2} (which creates the second level). The generated children elements are stored in an ordered tree hierarchical data structure (right).

a discontinuity could have sub-tetrahedra with non-matching faces. In other words, there exists the possibility of creating integration tetrahedra in element e_m with triangular faces (1, 2, 4) and (2, 3, 4), whereas those tetrahedra in element e_n could have faces (1, 2, 3) and (1, 3, 4). In order to avoid this situation, the constrained Delaunay function sets the faces of already created integration elements as the constraints for new elements sharing the same face. It should be emphasized that we are not conducting remeshing since the partition of unity (PoU) of the background mesh is kept intact, and constrained Delaunay is only used to locally generate integration elements; because these are used for numerical quadrature and for constructing enrichment functions, their requirements in terms of aspect ratios are less strict than those needed for Delaunay meshes.

The discussion above deals only with volumetric elements. It should be noted, however, that lower-dimensional elements (3-node triangles and 2-node lines) are also needed at times, *e.g.*, to apply boundary conditions—consider a distributed pressure applied to triangular faces of tetrahedra split by a crack. Therefore, lower-dimensional elements are also extracted from the created volumetric integration elements.

3.2. Hierarchical bookkeeping

An ordered tree data structure is used to store the relationships between original (parent) and new (children) elements. This hierarchical structure later facilitates the evaluation of weak and strong enrichment functions during the numerical quadrature of local element arrays. Children integration elements can in turn become parents of new children elements deeper in the hierarchy created by subsequent discontinuities.

The procedure is illustrated in Fig. 9 for a tetrahedral element e_i cut by two discontinuities Γ_{c1} and Γ_{c2} . This mesh element becomes the parent element for integration elements generated at the first hierarchical level by the first discontinuity Γ_{c1} . These children elements in turn become the parents for newly created integration elements at the second hierarchical level by a second discontinuity Γ_{c2} . The final hierarchical structure is also illustrated in Fig. 9.

This procedure can be used to create integration elements for mesh elements split by an arbitrary number of discontinuities. An element split by n discontinuities has $n + 1$ hierarchical levels in the tree, and this tree has the depth of the element split with the most number of discontinuities.

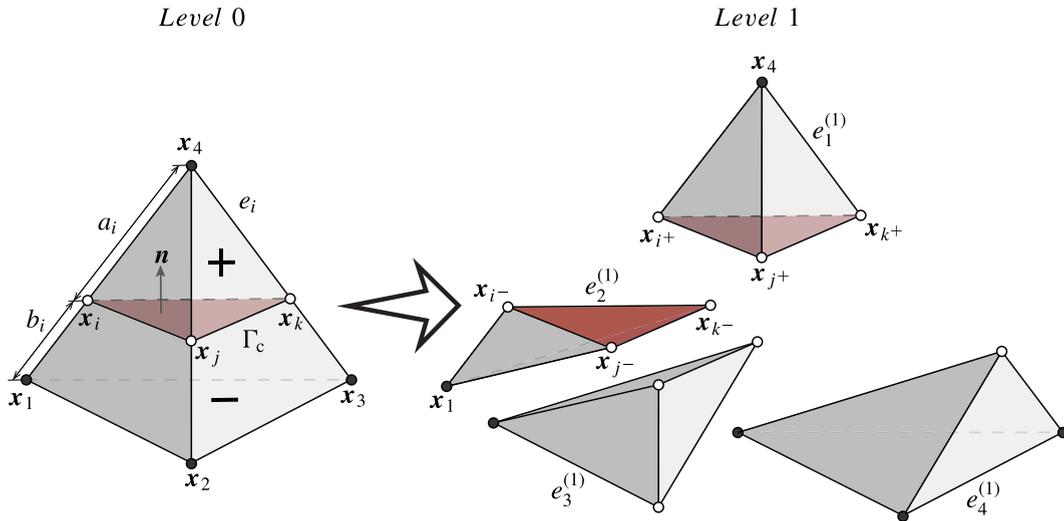


Fig. 10. Three enriched nodes x_i, x_j and x_k are created when a tetrahedral element e_i is subdivided into four children elements $e_1^{(1)}, e_2^{(1)}, e_3^{(1)}$ and $e_4^{(1)}$ by one discontinuity Γ_c .

4. Enrichment functions

In order to calculate the element stiffness matrix k_e and force vector f_e in (7), we need to evaluate shape and enrichment functions, together with their derivatives. Lagrange shape functions N are obtained as in standard FEM, following an iso-parametric formulation. Here we focus on the evaluation of enrichment functions ψ and χ .

4.1. Single discontinuity

Consider a tetrahedral element e_i intersected by a single crack Γ_c , as shown in Fig. 10. Because the element is split by a single discontinuity, all children elements created will be leaves (integration elements) in the ordered tree. The orientation of the crack Γ_c , given by its normal n , is used to define positive and negative regions. Three intersection points, labeled x_i, x_j , and x_k , are determined by the geometric engine. Enriched nodes are then created at these locations and denoted with the same symbols henceforth for simplicity.¹ A positive (negative) sign superscript is used if the node is referred in the positive (negative) side of the crack. In addition, Γ_c subdivides e_i into four children (integration) elements such that $e_1^{(1)}$ lays on at the positive side, and $e_2^{(1)}, e_3^{(1)}$, and $e_4^{(1)}$ are on the negative side. Enrichment functions will be constructed with the aid of standard Lagrange shape functions in these integration elements.

Now consider the leaf integration element $e_1^{(1)}$ on the positive side, which has three enriched nodes x_{i+}, x_{j+} , and x_{k+} . The Lagrange shape functions in this element, *i.e.*, $N_\iota^{(1)}(\mathbf{x}), \iota = \{i^+, j^+, k^+\}$, are used to create the weak and strong enrichment functions acting on this element. These functions can be expressed as

$$\psi_{i+} = w_{i+} N_{i+}^{(1)}(\mathbf{x}), \quad \psi_{j+} = w_{j+} N_{j+}^{(1)}(\mathbf{x}), \quad \psi_{k+} = w_{k+} N_{k+}^{(1)}(\mathbf{x}), \tag{11}$$

and

$$\chi_{i+} = s_{i+} N_{i+}^{(1)}(\mathbf{x}), \quad \chi_{j+} = s_{j+} N_{j+}^{(1)}(\mathbf{x}), \quad \chi_{k+} = s_{k+} N_{k+}^{(1)}(\mathbf{x}), \tag{12}$$

where w_z and s_z are weak and strong scaling (weight) factors for the z th enriched node, respectively, yet to be determined.

Similarly, for a children element on the negative side, such as $e_2^{(1)}$, weak and strong enrichment functions acting on the element are given by

$$\psi_{i-} = w_{i-} N_{i-}^{(1)}(\mathbf{x}), \quad \psi_{j-} = w_{j-} N_{j-}^{(1)}(\mathbf{x}), \quad \psi_{k-} = w_{k-} N_{k-}^{(1)}(\mathbf{x}), \tag{13}$$

¹As mentioned earlier, in the computer implementation it is convenient to create double nodes at each location when dealing with strong discontinuities; weak DOFs are associated to one of the nodes and strong DOFs to the other.

and

$$\chi_{i-} = s_{i-} N_{i-}^{(1)}(\mathbf{x}), \quad \chi_{j-} = s_{j-} N_{j-}^{(1)}(\mathbf{x}), \quad \chi_{k-} = s_{k-} N_{k-}^{(1)}(\mathbf{x}), \tag{14}$$

where w_z and s_z are the z th enriched node’s scaling factors for the negative side, and $N_{\iota}^{(1)}(\mathbf{x}), \iota = \{i^-, j^-, k^-\}$ are the Lagrange shape functions of this integration element. In order to enforce C^0 -continuity of the weak enrichment function, the weak scaling factors in both sides are given by the same expression (*i.e.*, $w_{z+} = w_{z-}, z = \{i, j, k\}$). For the weak scaling, we use a theoretically derived optimal function in order to preserve stability in the formulation [40]: $w_z = \sqrt{2\zeta_z(1 - \zeta_z)}$, where $0 \leq \zeta_z \leq 1$ is the relative length between the enriched and original nodes measured along the corresponding element edge. For enriched node \mathbf{x}_i , for instance, ζ_i is either $a_i/(a_i + b_i)$ or $b_i/(a_i + b_i)$ (since w_z is symmetric), where a_i, b_i are the Euclidean distances between \mathbf{x}_i and $\mathbf{x}_4, \mathbf{x}_1$ measured along the edge of the parent tetrahedron (see Fig. 10). The strong scaling factors are responsible for the field jump. Similarly to DE-FEM in 2-D [1], we choose the strong scaling factors so that the field jump has a unit magnitude, *i.e.*, $[[\chi_z]] = \chi|_{x_{z+}} - \chi|_{x_{z-}} = 1, z = \{i, j, k\}$. As a result, the associated strong DOFs β_z physically represent the displacement jump at \mathbf{x}_z . For example, the strong scalings corresponding to the enriched node \mathbf{x}_i are obtained by setting $s_{i+} = -\zeta_i$ and $s_{i-} = 1 - \zeta_i$. The weak and strong scaling factors for enriched nodes \mathbf{x}_j and \mathbf{x}_k are obtained analogously.

For enriched node \mathbf{x}_i , which is connected to integration elements $e_1^{(1)}$ and $e_2^{(1)}$, the enrichment functions ψ_i and χ_i are:

$$\psi_i = \begin{cases} w_i N_{i+}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_1^{(1)} \\ w_i N_{i-}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_2^{(1)}, \end{cases} \tag{15}$$

and

$$\chi_i = \begin{cases} s_{i+} N_{i+}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_1^{(1)} \\ s_{i-} N_{i-}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_2^{(1)}, \end{cases} \tag{16}$$

respectively. Fig. 11 shows a schematic representation of weak and strong enrichment functions, which attain a maximum absolute value at the location of the enriched node and decrease linearly to 0 at other element nodes (both standard and enriched).

It is worth mentioning that in DE-FEM the condition $[[\chi_z]] = \chi|_{x_{z+}} - \chi|_{x_{z-}} = 1, z = \{i, j, k\}$ is fulfilled by using the partition of unity property of the background mesh element [1]. Specifically, for enriched node \mathbf{x}_i , (16) is equivalent to using one parent nodal shape function on one side of the discontinuity and the negative of another parent nodal shape function on the other side. To wit,

$$\chi_i = \begin{cases} N_1^{(0)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_1^{(1)} \\ -N_4^{(0)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_2^{(1)}, \end{cases} \tag{17}$$

where the superscript (0) is used to indicate that the function belongs to the parent mesh element.

4.2. Multiple discontinuities

We now consider the more general situation where a mesh element is cut by more than one discontinuity. For constructing enrichment functions in a hierarchical way, we traverse the tree starting from leaf elements.

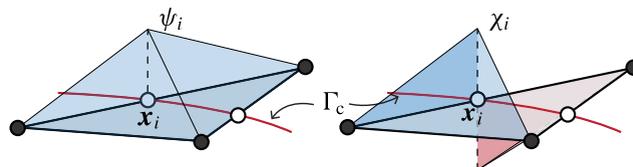


Fig. 11. Weak and strong enrichment functions ψ_i and χ_i corresponding to enriched node \mathbf{x}_i acting on a triangular surface of a split tetrahedron.

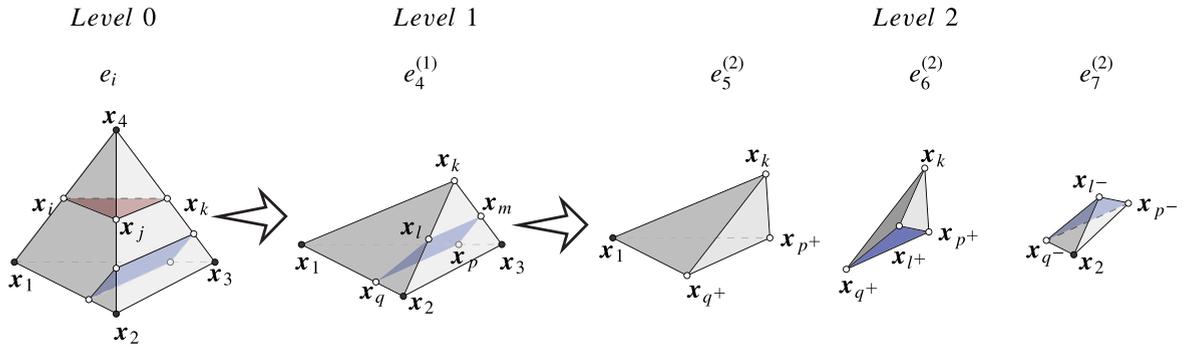


Fig. 12. The hierarchical relationship between the original parent element e_i , the element $e_4^{(1)}$ generated by the first discontinuity, the integration elements $e_5^{(2)}$, $e_6^{(2)}$, and $e_7^{(2)}$ in the last level. The enriched nodes x_i , x_j , and x_k are created by the first discontinuity, and x_l , x_m , x_p , and x_q are generated when handling the second one.

Consider the case shown in Fig. 12, which shows a mesh element crossed by two non-intersected discontinuities. The figure shows the hierarchical relationship between e_i (mesh element), $e_4^{(1)}$ (at hierarchical level 1), and $e_5^{(2)}$, $e_6^{(2)}$, and $e_7^{(2)}$ (at hierarchical level 2). We take enriched node x_q created after splitting children element $e_4^{(1)}$ by a second discontinuity. This enriched node is associated with leaf integration elements $e_5^{(2)}$, $e_6^{(2)}$ and $e_7^{(2)}$ at the second hierarchical level. We follow the same strategy as that used for handling one discontinuity to construct the enrichment functions associated with x_q . To wit,

$$\psi_q = \begin{cases} w_q N_{q^+}^{(2)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_5^{(2)} \\ w_q N_{q^+}^{(2)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_6^{(2)} \\ w_q N_{q^-}^{(2)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_7^{(2)}, \end{cases} \tag{18}$$

and

$$\chi_q = \begin{cases} s_{q^+} N_{q^+}^{(2)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_5^{(2)} \\ s_{q^+} N_{q^+}^{(2)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_6^{(2)} \\ s_{q^-} N_{q^-}^{(2)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_7^{(2)}. \end{cases} \tag{19}$$

where w_q, s_{q^\pm} are, as before, the weight factors, and $N_l^{(2)}(\mathbf{x}), l = \{q^+, q^-\}$ the Lagrange shape functions. It should be mentioned that an enrichment function at the second level of hierarchy—deepest level in this example—is non-zero only in elements connected to its corresponding enriched node (the enriched node’s support). The function is exactly zero in integration elements created at shallower hierarchical levels outside the enriched node’s support.

The same strategy is used to obtain enrichment functions for any enriched node at any hierarchical level. Integration element $e_4^{(1)}$, that is created by the first discontinuity, is connected to mesh nodes x_1, x_2, x_3 and enriched node x_k . For the latter, the weak and strong enrichment functions ψ_k and χ_k are obtained, respectively, by

$$\psi_k = \begin{cases} w_k N_{k^+}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_1^{(1)} \\ w_k N_{k^-}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_2^{(1)} \\ w_k N_{k^-}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_3^{(1)} \\ w_k N_{k^-}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_4^{(1)}, \end{cases} \tag{20}$$

and

$$\chi_k = \begin{cases} s_{k+} N_{k+}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_1^{(1)} \\ s_{k-} N_{k-}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_2^{(1)} \\ s_{k-} N_{k-}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_3^{(1)} \\ s_{k-} N_{k-}^{(1)}(\mathbf{x}) & \text{for } \mathbf{x} \in e_4^{(1)}. \end{cases} \quad (21)$$

After evaluating enrichment functions (and their derivatives), we append them to the Lagrange shape functions of parent elements to calculate the vector $\boldsymbol{\varphi}$ (and the derivatives are appended to those of parent shape functions to get the strain–displacement matrix \mathbf{B}). For instance, for one of the integration elements in the second level of hierarchy, such as $e_5^{(2)}$, the arrays $\boldsymbol{\varphi}$ and \mathbf{B} are given by

$$\boldsymbol{\varphi} = \left[N_1^{(0)} \dots N_4^{(0)} \quad w_k N_{k-}^{(1)} \quad w_q N_{q+}^{(2)} \quad w_p N_{p+}^{(2)} \quad s_{k-} N_{k-}^{(1)} \quad s_{q+} N_{q+}^{(2)} \quad s_{p+} N_{p+}^{(2)} \right], \quad (22)$$

$$\mathbf{B} = \left[\partial N_1^{(0)} \dots \partial N_4^{(0)} \quad w_k \partial N_{k-}^{(1)} \quad w_q \partial N_{q+}^{(2)} \quad w_p \partial N_{p+}^{(2)} \quad s_{k-} \partial N_{k-}^{(1)} \quad s_{q+} \partial N_{q+}^{(2)} \quad s_{p+} \partial N_{p+}^{(2)} \right], \quad (23)$$

where $N_i^{(0)}(\mathbf{x})$, $i = \{1, 2, 3, 4\}$ are the Lagrange shape functions of the parent mesh element e_i .

Elements not intersected by discontinuities are handled as in the standard FEM. On cut elements, the hierarchical structure adopted for handling an arbitrary number of intersections eases the computer implementation for conducting numerical quadrature. As an alternative to recursion, Algorithms 1 and 2 outline pseudo-code using an iterative loop for obtaining local arrays by Gauss integration. The algorithm shown is the same irrespective of the number of discontinuities.

In order to construct the local arrays for a (leaf) integration element $e^{(k)}$ at hierarchical level k , we input the modified mesh \mathcal{M} consisting of nodes and elements. The node set is composed by the original mesh nodes \mathcal{N} and the newly created enriched nodes \mathcal{N}_e . Similarly, the element set \mathcal{E} is composed by the original uncut mesh elements and the hierarchical tree data structure \mathcal{H} , which contain not only all children elements but also cut mesh elements. The function `getLocalArrays` proceeds as in standard FEM, where the stiffness matrix \mathbf{k} and the force vector \mathbf{f} are obtained by performing Gauss quadrature. Specifically, at each quadrature point ξ_i , the contributions to the local arrays are obtained by evaluating arrays of shape and enrichment functions \mathbf{N} (and their derivatives \mathbf{B}), weighted by the product between the quadrature weight Γ_i and the Jacobian determinant j .

Algorithm 1 Local stiffness matrix and force vector for an integration element

Input: Modified mesh $\mathcal{M} = \{\mathcal{N} \cup \mathcal{N}_e, \mathcal{E} \cup \mathcal{H}\}$, ordered tree \mathcal{H} , scaling factor map \mathcal{W} , leaf integration element $e^{(k)}$ at hierarchical level k , quadrature point weights $\boldsymbol{\Gamma}$ and coordinates $\boldsymbol{\xi}$, constitutive matrix \mathbf{C} and body force \mathbf{b} ;

Output: Local stiffness matrix \mathbf{k} and local force vector \mathbf{f} .

```

function GETLOCALARRAYS ( $\mathcal{M}, \mathcal{H}, \mathcal{W}, e^{(k)}, \boldsymbol{\Gamma}, \boldsymbol{\xi}, \mathbf{C}, \mathbf{b}$ )
     $\{\mathbf{k}, \mathbf{f}\} \leftarrow \{\mathbf{0}, \mathbf{0}\}$            – initialize local arrays
    for  $i \leftarrow 1, \dots, |\boldsymbol{\xi}|$  do     – loop over integration points
         $\{j, \mathbf{N}, \mathbf{B}\} \leftarrow \text{functions}(\mathcal{M}, \mathcal{H}, \mathcal{W}, e^{(k)}, \boldsymbol{\xi}_i)$  – get shape/enrichment functions (and their derivatives) and
        Jacobian determinant
        – add contributions to stiffness matrix and force vector
         $\mathbf{k} \leftarrow \mathbf{k} + \Gamma_i j \mathbf{B}^\top \mathbf{C} \mathbf{B}$  – update stiffness matrix
         $\mathbf{f} \leftarrow \mathbf{f} + \Gamma_i j \mathbf{N}^\top \mathbf{b}$    – update nodal force vector
    return  $\mathbf{k}, \mathbf{f}$ 
end function

```

The differences with respect to standard FEM are accounted for in functions. For a given quadrature point, the idea is to traverse the hierarchy from the leaf element $e^{(k)}$ towards its root parent mesh element $e^{(0)}$, so that all enrichment functions (together with their derivatives) are calculated and stacked in their corresponding arrays along the way. Only then the partition of unity shape functions (and corresponding derivatives) of the parent mesh element are accounted for.

Enrichment functions and their derivatives are obtained by first iterating over the enriched nodes belonging to the leaf element $e^{(k)}$. Depending on the type of discontinuity, contributions will be made to the weak term and possibly to the strong term when dealing with a strong discontinuity. Lagrange shape functions $N^{(k)}$ and their derivatives $N_{,\xi}^{(k)}$ are evaluated at $\xi_i^{(k)}$. Shape functions are used to calculate the global coordinates \mathbf{x} and their derivatives to compute the Jacobian matrix $\mathbf{J}^{(k)}$ (which is used throughout to compute derivatives with respect to global coordinate \mathbf{x}). Notice that we save the determinant j of the Jacobian matrix of the leaf integration element only. In the function `enrich`, $N^{(k)}$ and $N_{,\xi}^{(k)}$, together with the scaling factors w_j and s_j obtained from the map \mathcal{W} , are used to construct the enrichment functions $\psi^{(k)}$, $\chi^{(k)}$ and their derivatives $\psi_{,\mathbf{x}}^{(k)}$, $\chi_{,\mathbf{x}}^{(k)}$; these are then stacked to \mathbf{F} and $\mathbf{F}_{,\mathbf{x}}$ arrays, respectively. Then, we obtain the parent element (either mesh element or another integration element) from the ordered tree \mathcal{H} and the master coordinate at that element is obtained by an inverse mapping. The iteration continues until we reach the original mesh element (condition fulfilled by $k = 0$ in the algorithm).

Once all enrichment functions and their derivatives are computed, we add the contribution of the parent element through $N^{(0)}$ and their derivatives $N_{,\xi}^{(0)}$, which are also stacked in their corresponding arrays. To account for multiple DOFs per node in elasticity, the arrays are expanded and stored in \mathbf{N} and \mathbf{B} ; in the end these are returned together with the saved Jacobian determinant j .

Algorithm 2 Shape and enrichment functions, together with their derivatives

Input: Modified mesh $\mathcal{M} = \{\mathcal{N} \cup \mathcal{N}_e, \mathcal{E} \cup \mathcal{H}\}$, ordered tree \mathcal{H} , scaling factor map \mathcal{W} , leaf integration element $e^{(k)}$ at hierarchical level k , quadrature point at this level labeled $\xi_i^{(k)}$;
Output: Jacobian determinant j , and arrays of shape and enrichment functions \mathbf{N} together with their derivatives \mathbf{B} .

```

function FUNCTIONS ( $\mathcal{M}, \mathcal{H}, \mathcal{W}, e^{(k)}, \xi_i^{(k)}$ )
     $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_4]^\top \leftarrow (\mathcal{N} \cup \mathcal{N}_e) \cap e^{(k)}$  – get element coordinates
    for  $n_j \leftarrow \mathcal{N}_e \cap e^{(k)}$  do – loop over element’s enriched nodes
         $\{w_j, s_j\} \leftarrow \mathcal{W}(n_j)$  – get scaling factors for weak/strong enrichments at level  $k$ 
        do
             $\{N^{(k)}, N_{,\xi}^{(k)}\} \leftarrow \text{Lagrange}(e^{(k)}, \xi_i^{(k)})$  – shape functions and their derivatives
             $\mathbf{x} \leftarrow \mathbf{X}^\top N^{(k)}$  – obtain global coordinate
             $\mathbf{J}^{(k)} \leftarrow \mathbf{X}^\top N_{,\xi}^{(k)}$  – obtain Jacobian matrix
            if not  $j$  then – check if leaf Jacobian determinant is set
                 $j \leftarrow \det(\mathbf{J}^{(k)})$  – save leaf Jacobian determinant
             $N_{,\mathbf{x}}^{(k)} \leftarrow N_{,\xi}^{(k)} [\mathbf{J}^{(k)}]^{-1}$  – compute derivatives with respect to  $\mathbf{x}$ 
            – add child contribution
             $\{\psi^{(k)}, \psi_{,\mathbf{x}}^{(k)}\}, \{\chi^{(k)}, \chi_{,\mathbf{x}}^{(k)}\} \leftarrow \text{enrich}(w_j, s_j, N^{(k)}, N_{,\mathbf{x}}^{(k)})$  – compute enrichments/derivatives at level  $k$ 
             $\{\mathbf{F}, \mathbf{F}_{,\mathbf{x}}\} \leftarrow \left\{ \begin{bmatrix} \mathbf{F} & \psi^{(k)} & \chi^{(k)} \end{bmatrix}, \begin{bmatrix} \mathbf{F}_{,\mathbf{x}} & \psi_{,\mathbf{x}}^{(k)} & \chi_{,\mathbf{x}}^{(k)} \end{bmatrix} \right\}$  – stack functions and derivatives
             $k \leftarrow k - 1$  – decrease level of hierarchy
             $e^{(k)} \leftarrow \mathcal{H}(k, e^{(k)})$  – get parent element in the hierarchy
             $\xi_i^{(k)} \leftarrow \text{invert}(\mathbf{x}, e^{(k)})$  – obtain local master coordinate (invert map)
        while  $k \neq 0$  – if  $k \leftarrow 0$  then parent mesh element reached
    – add parent contribution
     $\{N^{(0)}, N_{,\xi}^{(0)}\} \leftarrow \text{Lagrange}(e^{(0)}, \xi_i^{(0)})$  – shape functions and their derivatives
     $\{\mathbf{F}, \mathbf{F}_{,\mathbf{x}}\} \leftarrow \left\{ \begin{bmatrix} N^{(0)} & \mathbf{F} \end{bmatrix}, \begin{bmatrix} N_{,\mathbf{x}}^{(0)} & \mathbf{F}_{,\mathbf{x}} \end{bmatrix} \right\}$  – stack parent shape functions and their derivatives
     $\{\mathbf{N}, \mathbf{B}\} \leftarrow \{\text{expand}(\mathbf{F}), \text{expand}(\mathbf{F}_{,\mathbf{x}})\}$  – expand to multiple DOFs per node
    return  $j, \mathbf{N}, \mathbf{B}$ 
end function

```

5. Numerical examples

In this section several numerical examples are provided to illustrate the performance of DE-FEM in 3-D. Unless explicitly stated, geometrical parameters, Young’s moduli, and magnitudes of applied tractions are understood under any consistent unit system.

5.1. Discontinuous patch test

Following [1], we first verify the DE-FEM formulation through a discontinuous patch test. Fig. 13(a) shows a schematic, where a cube is split by a horizontal crack Γ_c (the red plane) and tractions $\bar{t} = t e_1$ and $2\bar{t}$ are applied to the front surface above and below the crack, respectively. The back surface is fixed and the top and bottom ones are constrained along the e_3 direction as shown. An elastic modulus $E = 10$ and Poisson’s ratio $\nu = 0$ are used. The original finite element discretization comprises 16 nodes and 42 linear tetrahedra (see Fig. 13(b)); the new discretization after introducing the crack is also shown in Fig. 13(c).

The resulting stress field is shown in Fig. 13(d) on the deformed configuration. As apparent from the figure, DE-FEM passes this discontinuous patch test since there are two different constant stress states at either side of the crack. This means that DE-FEM is capable of generating two independent kinematic configurations in 3-D.

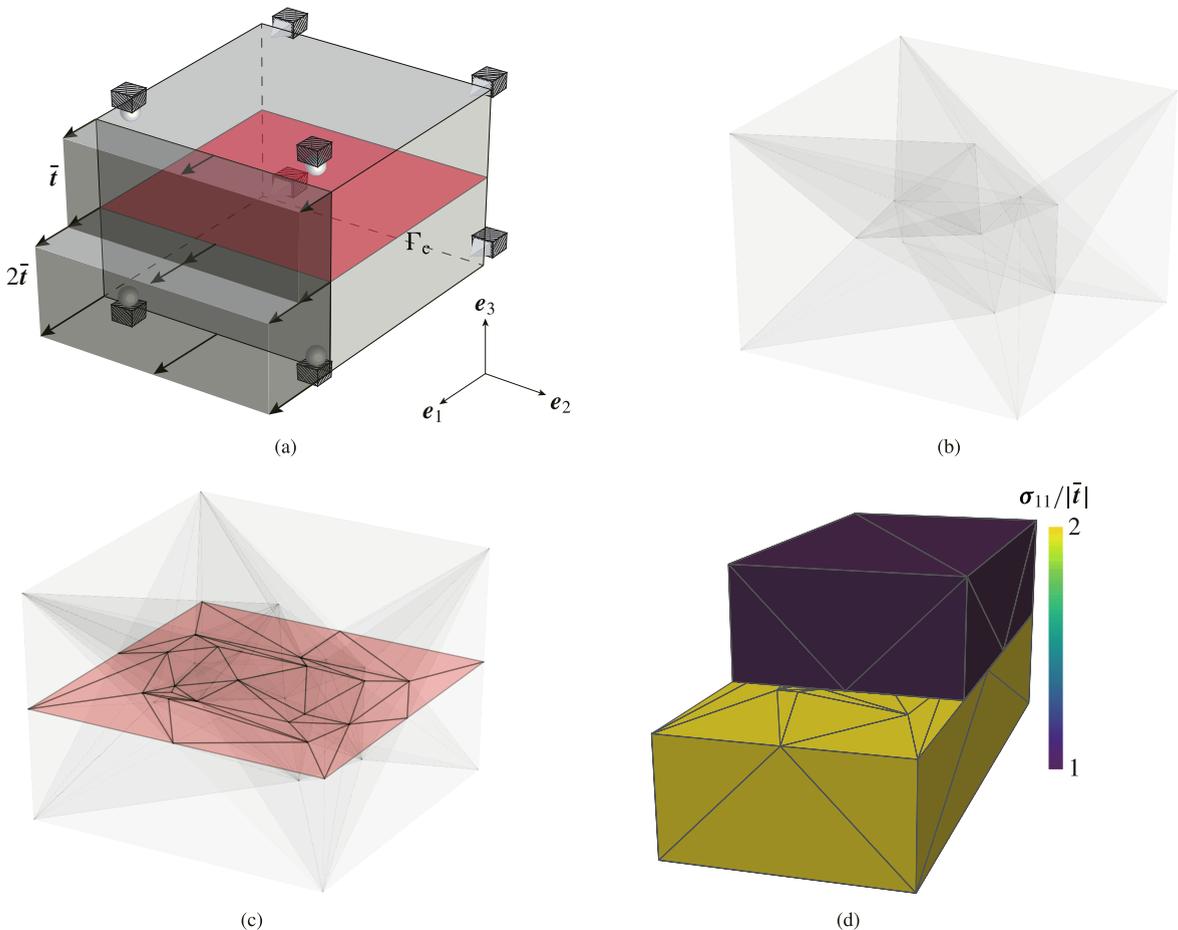


Fig. 13. (a) Schematic of the discontinuous patch test where tractions \bar{t} and $2\bar{t}$ are applied to the front surface above and below a crack Γ_c , respectively; (b) The original mesh composed by 42 linear tetrahedral elements; (c) New discretization after introducing the horizontal crack; (d) The normalized stress field on the deformed configuration, showing DE-FEM’s capability of recovering two independent kinematic fields. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

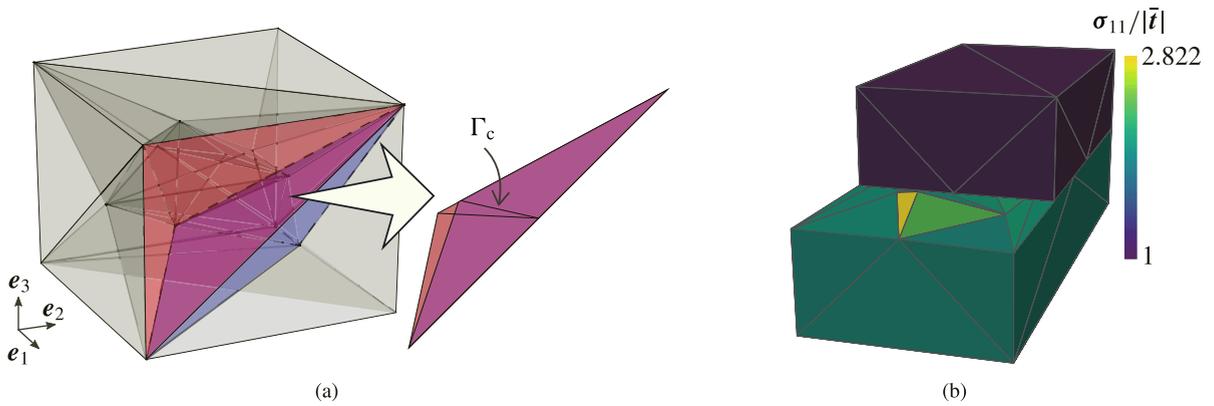


Fig. 14. (a) For the two contiguous mesh elements shown in color, we force a non-conforming integration mesh by manually changing the connectivity of integration elements below the crack (see mismatch of triangles below Γ_c); (b) Uniform and non-uniform stress fields, above and below the crack, respectively. The latter is caused by the non-conforming mesh. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As mentioned in Section 3, a non-conforming enriched mesh may be generated by the Delaunay algorithm during the creation of children integration elements. In order to display the effect of using a non-conforming mesh, consider two contiguous mesh elements, shown in Fig. 14(a) in different colors. We modified the connectivity of their corresponding children elements below the crack by hand. Fig. 14(b) shows that the stress is no longer constant. In this work we have chosen to use constrained Delaunay to avoid these situations altogether.

5.2. Convergence

The convergence behavior of DE-FEM is investigated by comparing the convergence rates with those obtained by standard FEM on matching meshes for problems under modes I, II, and III. The computational domain used in this example consists of a cube discretized by linear tetrahedral elements. An elastic modulus $E = 10$ and Poisson’s ratio $\nu = 0.3$ are used. The matching FE meshes used in standard FEM have 24, 384, 3072, 24 576, and 196 608 tetrahedra, as shown in Figs. 15(a)–15(e). Five different non-matching FE meshes, containing 6, 162, 2058, 20 250, and 178 746 elements as shown in Figs. 15(f)–15(j), are used in DE-FEM. The exact displacement field, given in the Appendix for reference, is prescribed on the cube surface.

The error with respect to the exact solution is then measured in the energy norm, *i.e.*,

$$\frac{\|\mathbf{u} - \mathbf{u}^h\|_{E(\Omega^h)}}{\|\mathbf{u}\|_{E(\Omega^h)}} = \sqrt{\frac{\sum_{e \in \Omega^h} \int_e (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h)^\top \mathbf{C} (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h) \, d\Omega}{\sum_{e \in \Omega^h} \int_e \boldsymbol{\varepsilon}^\top \mathbf{C} \boldsymbol{\varepsilon} \, d\Omega}}, \tag{24}$$

where \mathbf{u} is the analytical solution, \mathbf{u}^h is the FE approximation by either DE-FEM or standard FEM, $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}^h$ are their corresponding strains, and \mathbf{C} is the elasticity tensor. Fig. 16 shows deformed configurations for modes I, II, and III, obtained by standard FEM 16(a)–16(c) and DE-FEM 16(d)–16(f). For the latter, the deformation field is accurately visualized after post-processing [1].

Convergence results are summarized in Fig. 17, where we see that convergence rates of DE-FEM are the same as those of standard FEM with the use of matching meshes. It is worth noting that convergence rates are not optimal for either method due to the lack of special techniques for handling the singularity along the crack front.

To investigate the influence of integration elements’ aspect ratios, the element-wise error in the energy norm, given by

$$\|\mathbf{u} - \mathbf{u}^h\|_{E(e)} = \sqrt{\int_e (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h)^\top \mathbf{C} (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h) \, de}, \tag{25}$$

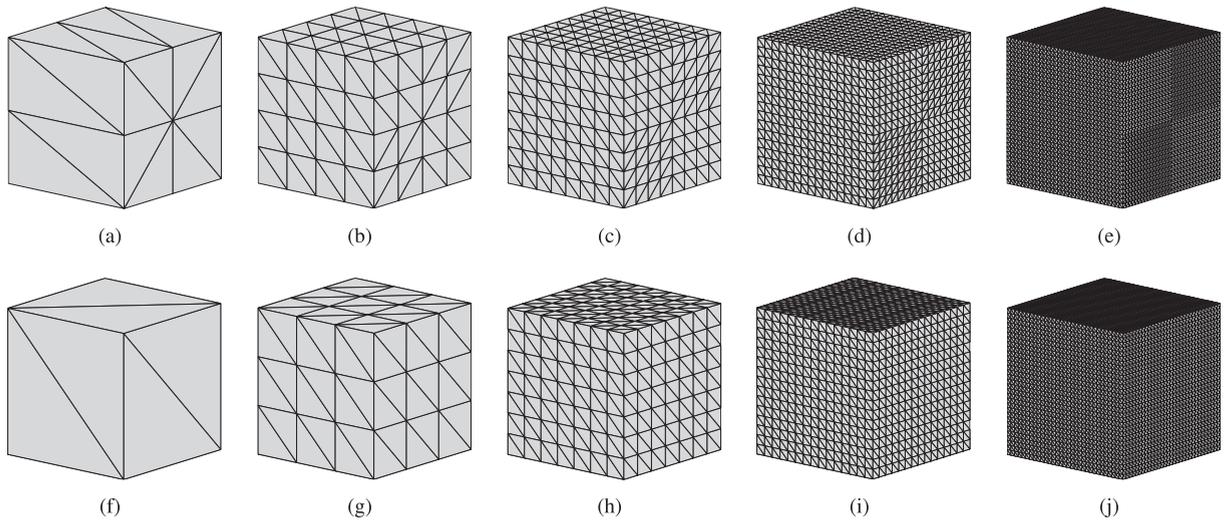


Fig. 15. Matching meshes with (a) 24, (b) 384, (c) 3072, (d) 24 576, and (e) 196 608 linear tetrahedral elements used for standard FEM, and the non-matching meshes for DE-FEM with (f) 6, (g) 162, (h) 2058, (i) 20 250, and (j) 178 746 elements.

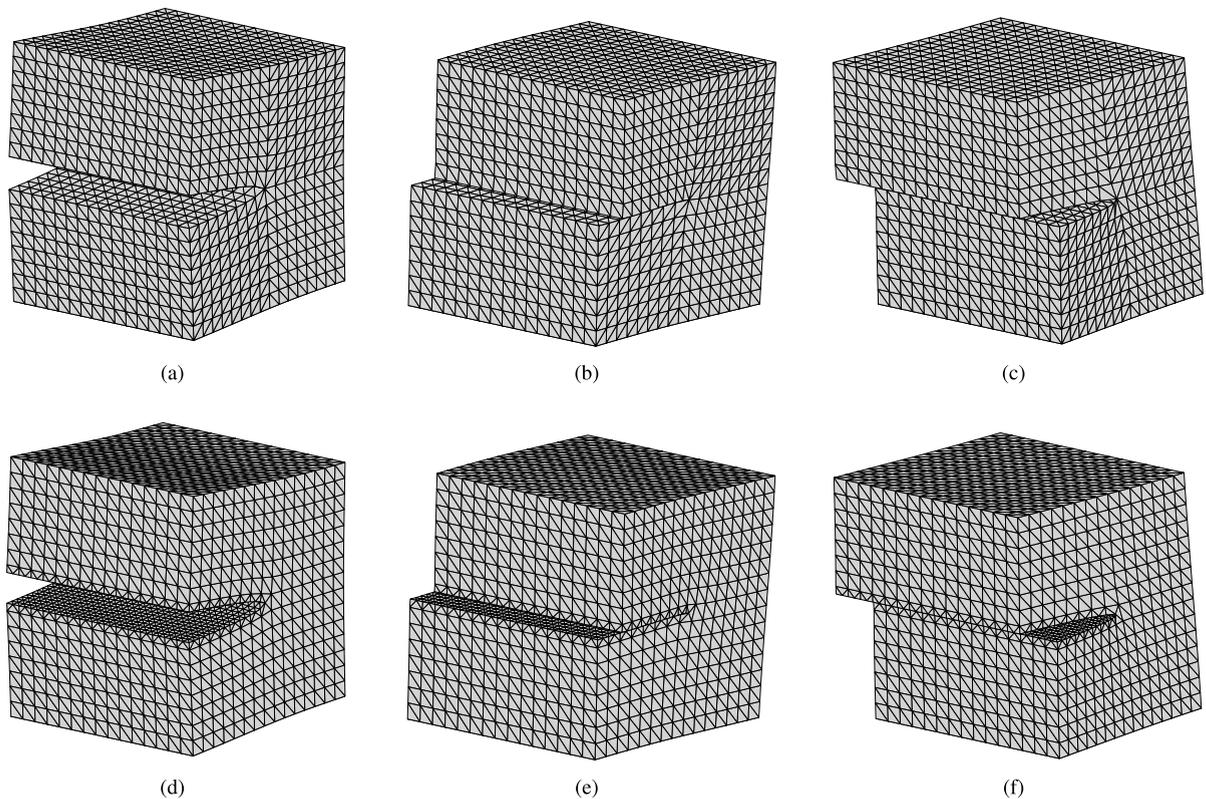


Fig. 16. Deformed configurations for modes I, II, and III for standard FEM (a)–(c) and DE-FEM (d)–(f). The latter are obtained after post-processing.

is evaluated for different locations of the crack on the non-matching mesh shown in Fig. 15(h). The crack, which moves from the middle of the background element to the position close to background mesh nodes, cuts original hexahedral units (composed of six tetrahedra) and creates integration elements (see Figs. 18(a) and 18(b)). The

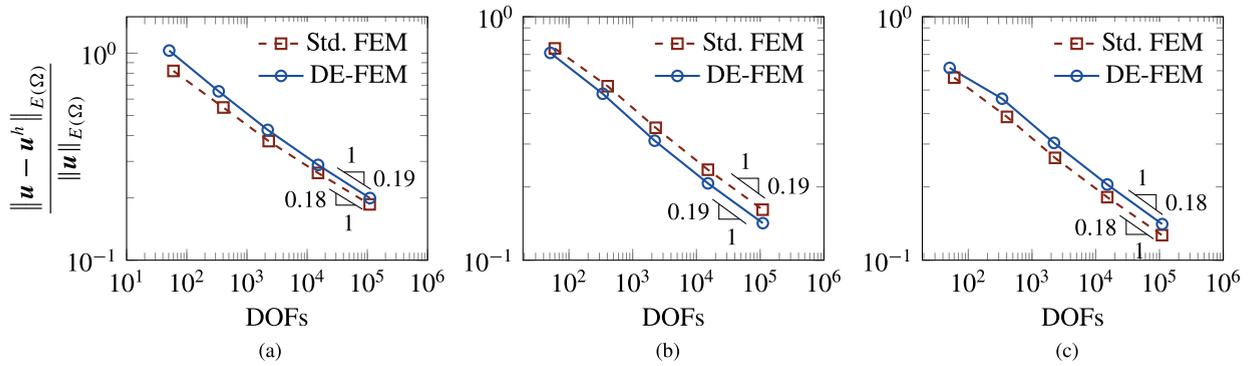


Fig. 17. Error in energy norm given by (24) as a function of the total number of DOFs for modes I (a), II (b), and III (c). The figures show that both methods have the same convergence rates, although these are not optimal.

darkest integration elements shown in the figure are then considered for investigating the influence of aspect ratio Υ , which we define following [41] as

$$\Upsilon = \frac{l_{\text{avg}}^3/V}{8.48}, \tag{26}$$

where l_{avg} is the average edge length, and V is the tetrahedron volume. The aspect ratios considered range from 1.35 to 894.76. It should be mentioned that the aspect ratio of an integration element is not always minimum when the crack cuts through the middle of the background mesh. Fig. 18(c) shows the squared local error in the energy norm defined by (25) for the two integration elements e_I and e_{II} shown in Figs. 18(a) and 18(b). The figure also shows the error measure normalized by the integration elements’ volume. Because the local error involves integrating over element’s volume, we normalize to single out the effect of aspect ratio. Indeed the solid curves show that increasing Υ also increases the local error, although the effect is more pronounced at low values of aspect ratio. The dashed curves show the combined volume/aspect ratio effect. The increase in local error caused by increasingly bad aspect ratios is counterbalanced by the element’s volume, which decreases as the crack approaches the background mesh nodes (Fig. 18(b)). For the element in the corner, which has the constant aspect ratio $\Upsilon = 1.24$, the local error without normalization shows the same tendency as the above two integration elements due to the decreasing element volume, and the normalized error ranges from 1×10^{-5} to 1×10^{-4} . Consequently, bad aspect ratios in integration elements have not much influence in the global measure defined in (24), and reducing the mesh size even further makes the contribution of the local error in integration elements even less significant.

5.3. Stress intensity factors

We now study DE-FEM’s capability to extract stress intensity factors (SIFs) for the single edge-crack tension problem of Fig. 19(a), where for a crack width $a = 2$, we set $a/w = 0.5$, $l/w = 1.5$, and $h/w = 1.75$. Traction $\vec{t} = \pm\sigma_0\mathbf{e}_3$ are applied at top and bottom surfaces. A local Cartesian coordinate system $(\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3)$ is located at the center of the block, where x'_2 is measured along the crack front. The material properties are $E = 10$, and $\nu = 0.3$. The non-matching mesh used, shown in Fig. 19(b), is defined over a $39 \times 39 \times 39$ Cartesian grid, resulting in 355 914 linear tetrahedra.

We use the domain integral method to calculate the energy release rate and the M_1 integral method to extract SIFs [42,43]. For a point along the crack front, which is parameterized by the coordinate s (see Fig. 20), the local interaction energy release rate is given by

$$I(s) = \frac{1 - \nu^2}{E} \left(2K_I^{(1)}K_I^{(2)} + 2K_{II}^{(1)}K_{II}^{(2)} \right) + \frac{1 + \nu}{E} \left(2K_{III}^{(1)}K_{III}^{(2)} \right), \tag{27}$$

where superscripts (1) and (2) indicate the actual and auxiliary fields, respectively, and $K_i^{(1,2)}$, $i \in \{I, II, III\}$ denote their corresponding SIFs. $I(s)$ can be extracted from the total interaction energy release rate $\bar{I}(s)$ under the

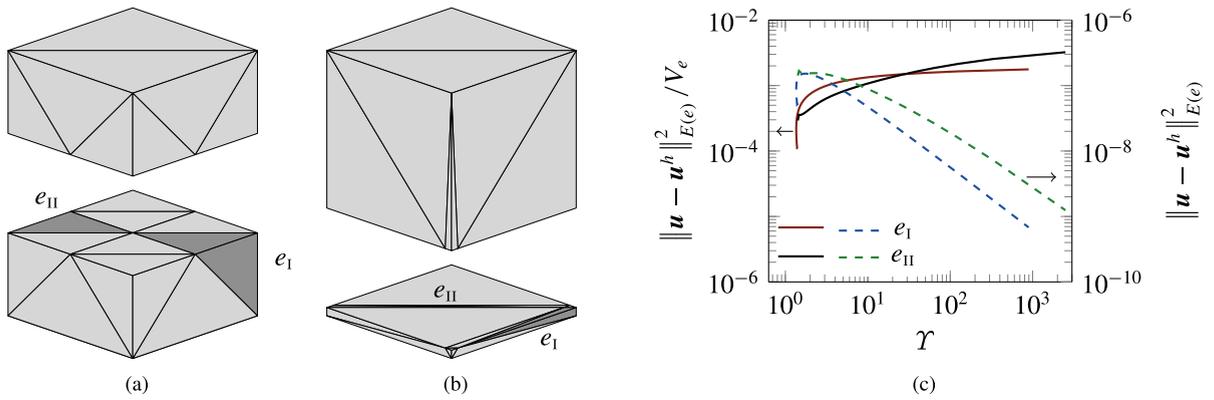


Fig. 18. (a,b) Integration elements created when crack cuts a hexahedral unit composed by six tetrahedra: (a) crack through the center; and (b) crack cutting close to background mesh nodes. The darkest integration elements are considered for investigating the influence of aspect ratio \mathcal{T} in the local error. (c) Squared local error in the energy norm defined by (25) with and without normalizing by its volume, shown by solid and dashed curves, respectively. While the former curve shows the local error increases with increasing values of aspect ratio, the latter shows that the effect of high aspect ratios is counterbalanced by smaller volumes.

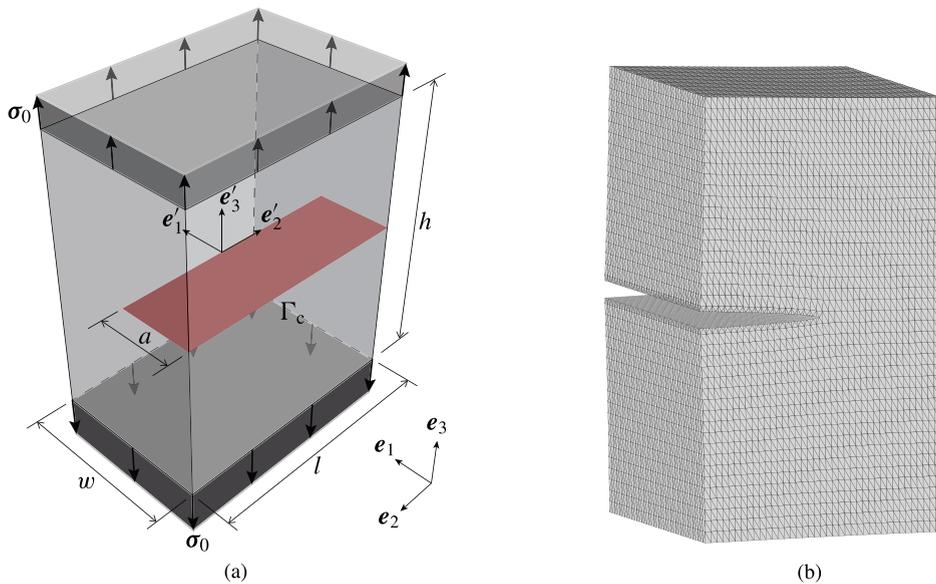


Fig. 19. Single edge-crack tension example for extracting SIFs: (a) Problem schematic, where a cracked block is loaded by tractions $\bar{\mathbf{i}}$ acting on top and bottom faces; (b) DE-FEM's deformed configuration.

assumption that it is constant within the integration domain $V(s)$ of length 2η that encloses a crack front segment (see Fig. 20). $I(s)$ and $\bar{I}(s)$ are given, respectively, by

$$I(s) = \frac{\bar{I}(s)}{\int_{s-\eta}^{s+\eta} q_k(s') \mu_k(s') ds'}, \tag{28}$$

and

$$\bar{I}(s) = \int_{V(s)} \left((\sigma_{ij}^{(1)} \frac{\partial u_i^{(2)}}{\partial x_k} + \sigma_{ij}^{(2)} \frac{\partial u_i^{(1)}}{\partial x_k}) \frac{\partial q_k}{\partial x_j} - \sigma_{ij}^{(1)} \epsilon_{ij}^{(2)} \frac{\partial q_k}{\partial x_k} \right) dV, \tag{29}$$

where q_k is a continuous function with zero value outside $V(s)$, and $\mu_k(s')$ are the components of the unit vector generated from the junction of planes at point s' which are perpendicular to the crack front and tangential to the

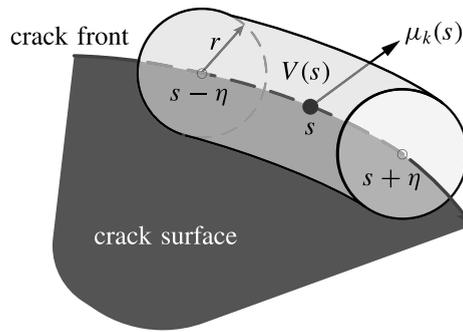


Fig. 20. The integration domain $V(s)$ enclosing the crack front segment between $s - \eta$ and $s + \eta$ for calculating the interaction energy release rate $\bar{I}(s)$.

Table 1

The values of normalized SIF $K_I/\sigma_0\sqrt{\pi a}$ obtained by DE-FEM.

x'_2/l	0	0.051	0.103	0.154	0.205	0.256	0.307	0.346	0.397	0.448	0.5
$K_I/\sigma_0\sqrt{\pi a}$	2.747	2.744	2.738	2.728	2.713	2.691	2.660	2.628	2.566	2.441	2.039

crack plane. For the sake of obtaining the mixed-mode SIFs from (27), (28), and (29), an appropriate auxiliary equilibrium state is chosen. For instance, if mode I is set as the auxiliary field, then $K_I^{(2)} = 1$, $K_{II}^{(2)} = K_{III}^{(2)} = 0$, and the mode I SIF of the actual field is $K_I^{(1)} = EI(s)/(2(1 - \nu^2))$. A similar strategy is used to obtain SIFs for modes II and III. For the integration domain to compute SIFs, we set $\eta = 12/39$ and $r = 16/39$.

Fig. 21 shows the normalized SIF $K_I/\sigma_0\sqrt{\pi a}$ as a function of the relative coordinate x'_2/l along the crack front, and the corresponding values are given in Table 1. Under plane strain conditions, the analytical normalized SIF is 2.8284 [44]. The results obtained by DE-FEM show the same trend as other reference solutions; the SIF values approach the result obtained under the plane strain condition as x'_2 reaches the center. Moreover, in accordance with other references, DE-FEM also displays a decreasing K_I as x'_2 reaches the boundary of the domain. For the study of Raju and Newman [45], singularity elements (where square-root terms are introduced to approximate the displacement field) were used to capture the singular stress field along the crack front. Generating the required mesh was, however, quite involved. For the study of Sukumar et al. [12], which used X/GFEM, special enrichment functions were used to capture the singular stress field. Yet, the required computer implementation is also more involved since, among other things, singular enrichments need special integration procedures [26]. There is always a trade-off between accuracy and complexity, and DE-FEM can obtain SIFs in a fully mesh-independent manner with a simpler computer implementation than that needed for X/GFEM, at the expense of some accuracy lost—1.4% below the value obtained by X/GEM at the center [12]. All these results were obtained on a coarse mesh, so for reference we also added SIFs computed by X/GFEM where the values were obtained using the mesh with local refinement along the crack front [46]. The SIF at the center, which is affected by the finite domain boundary, is higher than the one under the plane strain assumption.

5.4. Stability

In this section we study how the condition number of the stiffness matrix grows with mesh refinement for both DE-FEM and standard FEM. As mentioned in the introduction, enriched methods like X/GFEM suffer from stability issues when discontinuities get arbitrarily close to nodes of the background mesh, and discontinuity-enriched methods like HIFEM/DE-FEM are not oblivious to this problem. For an integration element, the enriched components of the local stiffness matrix are inversely proportional to the element’s volume (in fact to the determinant of its Jacobian matrix) [40]. Therefore, vanishingly small integration elements quickly deteriorate the overall condition number of the system matrix. For weak discontinuities alone, i.e., for IGFEM/HIFEM, an optimal function based on a 1-D analysis can be used to scale weak enrichments, resulting in a stable formulation [40]. However, it is also shown in that work that even if weak enrichments are not scaled, using a simple Jacobi-like preconditioner also results in a stable condition number. This is the approach we follow next.

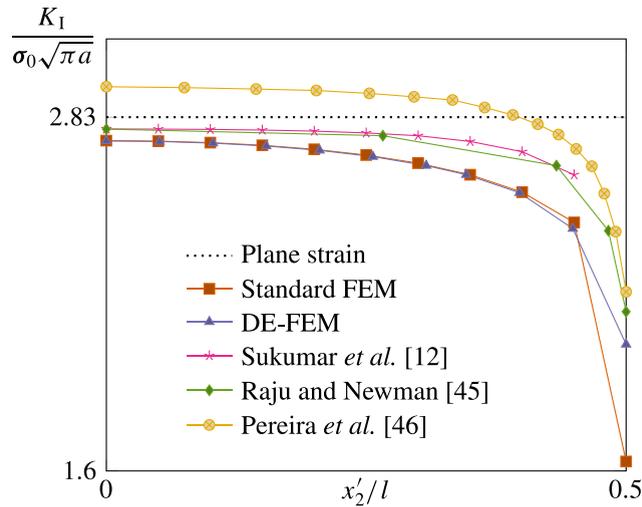


Fig. 21. Normalized SIF values as a function of relative local coordinate x'_2/l . The results obtained by DE-FEM show the same trend as other reference solutions and accuracy that is on a par with that of standard FEM on matching meshes.

As it is commonly done in SGFEM, we study the condition number of a modified stiffness matrix $\hat{\mathbf{K}} = \mathbf{DKD}$, where $D_{ij} = \delta_{ij}/\sqrt{K_{ij}}$ and thus $\hat{K}_{ii} = 1$. This modification is tantamount to applying a Jacobi-like preconditioner. The condition number $\kappa(\hat{\mathbf{K}})$ is then obtained as the ratio between the maximum eigenvalue λ_{\max} and the minimum (non-zero) eigenvalue λ_{\min} . We compare this condition number to that of the standard FEM component $\kappa(\hat{\mathbf{K}}_{uu})$ (refer to (10)).

Two problems are used to study the stability. The first one consists of a solid cube containing a slanted crack with an orientation that favors the creation of arbitrarily shaped integration elements (see Fig. 22(a)). The planar crack’s normal is $\mathbf{n} = 0.1617\mathbf{e}_1 + 0.1819\mathbf{e}_2 + 0.9699\mathbf{e}_3$. For the second example, we introduce an extra spherical material interface with radius $r = 0.59$ that intersects the planar crack (see Fig. 22(b)). The structured background meshes considered here are the same as those used in the convergence study.

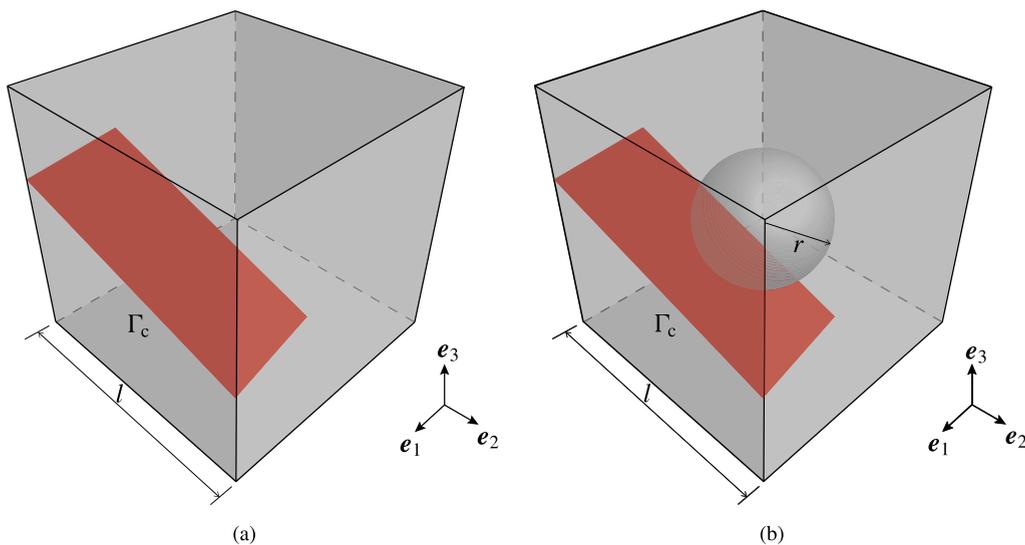


Fig. 22. Problem schematics for studying DE-FEM’s stability: (a) cube of side length $l = 2$ partially cut by a planar crack ; and (b) spherical material interface with radius $r = 0.59$ intersects the planar crack.

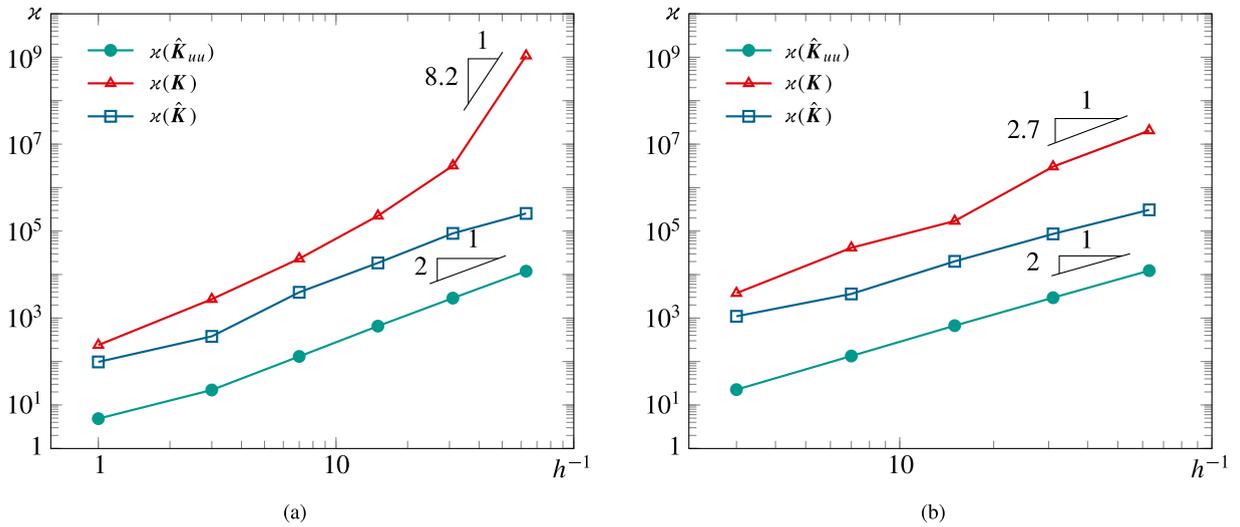


Fig. 23. Condition number as a function of h^{-1} for (a) the planar crack shown in Fig. 22(a); and (b) the intersecting discontinuities shown in Fig. 22(b) show that, while the condition number of the raw system matrix $\kappa(\mathbf{K})$ is unstable, applying a simple preconditioner results in $\kappa(\hat{\mathbf{K}})$, which grows roughly at the same rate as that of the standard FEM component $\kappa(\hat{\mathbf{K}}_{uu})$.

The results of both cases are reported in Fig. 23, where the condition number is shown as a function of the reciprocal of the mesh size. Our reference curve is that of the standard FEM component $\kappa(\hat{\mathbf{K}}_{uu})$, which scales as $\mathcal{O}(h^{-2})$. The figure shows that the condition number of the raw stiffness matrix $\kappa(\mathbf{K})$, without the use of a preconditioner, is unstable. Using the preconditioner results in the curve $\kappa(\hat{\mathbf{K}})$, which has about an order of magnitude higher condition number than the standard part. However, the rate of growth is roughly the same, and therefore DE-FEM is stable if a simple preconditioner is applied.

5.5. Bone fracture

We now use DE-FEM to simulate the fracture mechanics of a femur whose geometry was obtained from [47], as shown in Fig. 24(a). The femur is subjected to a prescribed displacement field $\bar{\mathbf{u}} = \pm 1 \text{ mm } \mathbf{e}_1$ at its ends and a crack Γ_c that partially splits the bone is introduced.

Bone can be considered a composite material with cancellous (inner) and cortical (outer) phases with completely different material properties [48]. For simplicity, here we consider them as isotropic materials with elastic constants $E = 10.4 \text{ GPa}$, $\nu = 0.12$ (cancellous bone) and $E = 18.6 \text{ GPa}$, $\nu = 0.3$ (cortical bone) [49,50]. In addition, a level set function is introduced to describe their interface. Fig. 24(b) shows the finite element mesh used, composed of 7355 tetrahedral elements. Although the mesh represents the femur external boundary accurately, the mesh is non-conforming not only to the interface between materials, but also to the crack Γ_c .

The results of this simulation are shown in Fig. 25, where we see that new children (integration) elements are created properly at each side of the interface (Fig. 25(a)). Fig. 25(b) displays the resulting von Mises stress field, highlighting the stress concentration near the crack front and the higher stress in cortical bone. This example showcases DE-FEM’s versatility not only in modeling both weak and strong discontinuities with a unified formulation, but also in resolving them even if they intersect the same element. This is possible due to the hierarchical implementation discussed in Section 2, which allows for the splitting of elements by several discontinuities seamlessly.

6. Summary and conclusions

In this paper we introduced DE-FEM for solving problems with both weak and strong discontinuities in 3-D. We discussed our implementation of a geometric engine to handle complex 3-D configurations. By means of constrained Delaunay, the geometric engine handles all possible scenarios that arise when splitting tetrahedral elements with

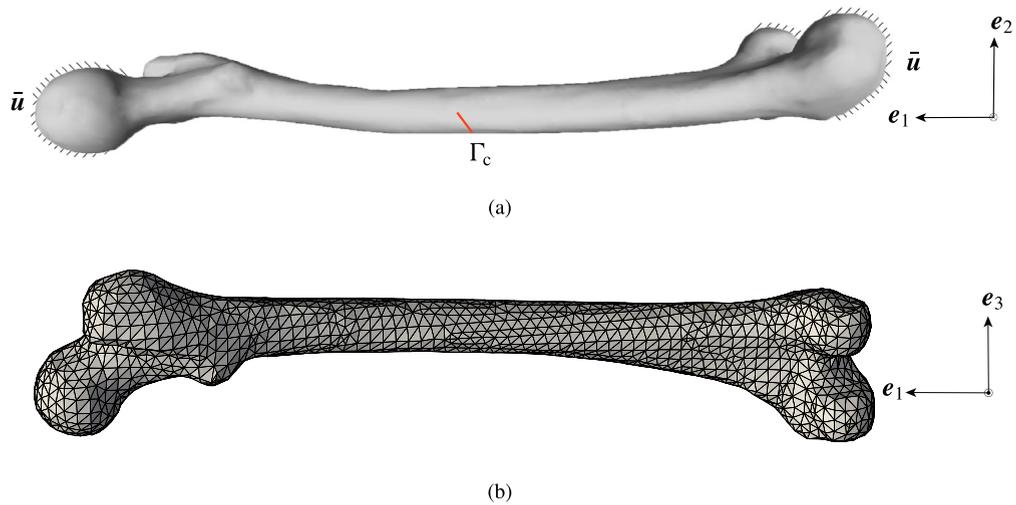


Fig. 24. (a) Essential boundary conditions prescribed on the femur model containing a crack Γ_c ; (b) The mesh used matches the femur's external boundary but not the material interface nor the crack.

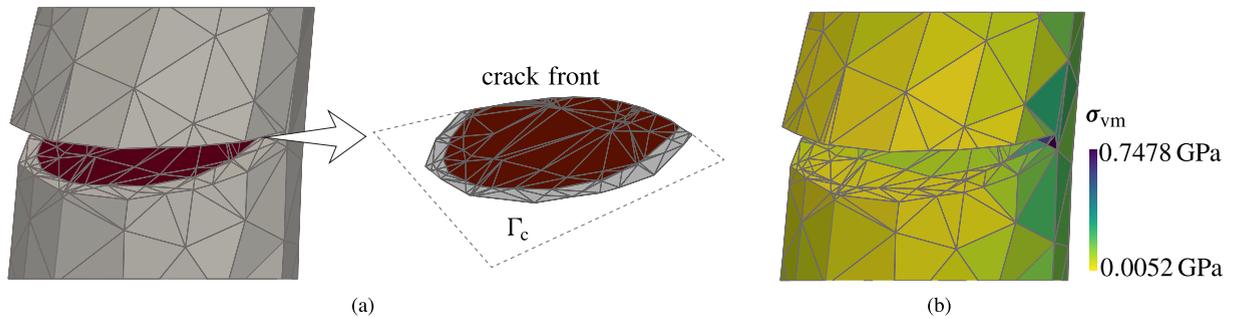


Fig. 25. Discretized model near the crack extracted from the whole bone where the background mesh is not matching to the interface nor to the crack: (a) The red and gray parts model cancellous and cortical bone, respectively. The resulting integration mesh is matching to both the interface and the crack; (b) von Mises stress field highlighting the higher stress in cortical bone. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

complex interfaces and/or cracks. In addition, a hierarchical data structure was discussed for resolving an arbitrary number of discontinuities within a single mesh element. This data structure is then used to construct hierarchically weak and strong enrichment functions.

The *integration mesh* that is created in our DE-FEM implementation serves several purposes. Firstly, the tetrahedra that result from subdividing parent mesh elements are used to construct the enriched approximation. Indeed the enrichment functions are built with the aid of Lagrange shape functions of integration subdomains. Secondly, by using tetrahedra as subdomains we guarantee robustness when handling multiple discontinuities since tetrahedra are the simplest 3-D volumes to split. Thirdly, creating integration tetrahedra with sides aligned to discontinuities ensures that the discontinuous functions within these elements are smooth. Other integration schemes that do not require subdividing into integration elements, but which can otherwise recursively perform the numerical quadrature, are also possible for weak discontinuities. However, because of the non-smooth nature of the functions, the computational cost is higher because several levels of recursion are needed along discontinuities [51]. Lastly, the integration mesh is also used to visualize the post-processed displacement field.

Through a discontinuous patch test we showed that the DE-FEM formulation is able to generate two independent kinematic fields. We also showed that DE-FEM converges at the same rate as the standard FEM with matching meshes and that DE-FEM can be used to extract stress intensity factors. Finally, we simulated the challenging problem of resolving the interplay between weak and strong discontinuities within a complex-shaped femur model.

Table 2

Characteristics of Standard FEM, X/GFEM, and DE-FEM.

	Standard FEM	X/GFEM	DE-FEM
Decoupling between mesh and discontinuities	×	✓	✓
PoU for localizing enrichments	Not applicable	✓	Local by construction
Physical meaning of standard DOFs	✓	Shifting required	✓
Non-zero essential boundary conditions	Direct	Penalty, Lagrange multipliers, etc.	Direct
Condition number growth (stability)	$\mathcal{O}(h^{-2})$	$\mathcal{O}(h^{-2})$ with SGFEM [16]	$\mathcal{O}(h^{-2})$
Numerical integration	Standard	Integration mesh in cut elements	Conforming integration mesh
Optimal convergence	×	✓	TBD

In Table 2 we compare DE-FEM to both standard FEM and to X/GFEM. We see that DE-FEM also decouples the underlying mesh from discontinuities, which is X/GFEM's most attractive feature. However, contrary to X/GFEM, DE-FEM also retains the attractive properties of standard FEM. Because enrichment functions are local to cut elements by construction, there is no need to use the mesh's partition of unity shape functions to localize enrichments. Also, since enrichments are exactly zero at nodes of the background mesh, DOFs associated to these keep their physical meaning. Another consequence of enrichments vanishing at mesh nodes is that it is straightforward to impose Dirichlet boundary conditions. Non-homogeneous essential boundary conditions can be prescribed strongly also at enrichment node locations through a simple multiple point constraint. In fact, there is ongoing work on the use of DE-FEM as an immersed boundary or fictitious domain technique with strong enforcement of Dirichlet boundary conditions [27]. Moreover, the requirements for the quality of the integration mesh are less strict than those of a proper *matching* mesh. Regarding stability, we showed that the condition number of the whole stiffness matrix grows at roughly the same rate as that of standard FEM. Therefore DE-FEM is stable with the use of a simple preconditioner.

The simple formulation in DE-FEM leads to a simpler computer implementation when compared to X/GFEM. The current formulation cannot get optimal convergence rates for problems with singularities because the current formulation is devoid of singular enrichments. These could be added if higher accuracy is required—*e.g.*, for extracting stress intensity factors. Nevertheless, adding singular enrichments to the formulation would complicate the computer implementation, not only because of the need for bookkeeping new enriched DOFs (and variable DOFs per mesh node), but also because singular enrichments need to be integrated accurately, for which many more quadrature points are required. Adding DOFs also leads to increasing the cost of solving the linear system of equations. Although we did not use any enrichments to capture singularities, the predictions of SIFs are not far from other more accurate values for the same mesh size h ; our predicted result at the center was just 1.4% below the value obtained by X/GFEM.

The geometric engine is the most intricate part of the implementation because of the complexity of computational geometry operations involved and the need for robustness with respect to the placement of discontinuities. This is the crux not only of DE-FEM and other interface-enriched formulations such as IGFEM or HIFEM, but also for X/GFEM when dealing with discontinuities. For DE-FEM we showed the importance of creating a conforming enriched discretization for integration. Although the geometric engine described in this work suffices for the types of problems described, there is wide room for improvement and this will be the subject of subsequent work.

Acknowledgment

Jian Zhang acknowledges financial support from China Scholarship Council under contract 201606060130.

Appendix. Analytical fields for modes I, II, and III

For completeness, the equations of the displacement and stress fields for modes I, II, and III are provided. For displacement fields, they are expressed as

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \frac{K_1}{\sqrt{2\pi r}} \begin{bmatrix} \frac{r}{4\mu} [(2\kappa - 1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2}] \\ \frac{r}{4\mu} [(2\kappa + 1) \sin \frac{\theta}{2} - \sin \frac{3\theta}{2}] \\ -\frac{2\nu_1 z}{E} \cos \frac{\theta}{2} \end{bmatrix}, \quad (\text{A.1})$$

$$\mathbf{u} = \frac{K_{II}}{\sqrt{2\pi r}} \begin{bmatrix} \frac{r}{4\mu} [(2\kappa + 3) \sin \frac{\theta}{2} + \sin \frac{3\theta}{2}] \\ \frac{r}{4\mu} [-(2\kappa - 3) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2}] \\ \frac{2\nu_1 z}{E} \sin \frac{\theta}{2} \end{bmatrix}, \tag{A.2}$$

$$\mathbf{u} = \frac{K_{III}}{\mu} \sqrt{\frac{2r}{\pi}} \begin{bmatrix} 0 \\ 0 \\ \sin \frac{\theta}{2} \end{bmatrix}, \tag{A.3}$$

where μ is the shear modulus, κ and ν_1 are equal to $3 - 4\nu$ and 0 under the plane strain, or $(3 - \nu)/(1 + \nu)$ and ν under the plane stress, respectively, r and θ are polar coordinates centered at the point along the crack front.

Stress fields around the crack front are given, respectively, by

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = \frac{K_I}{\sqrt{2\pi r}} \cos \frac{\theta}{2} \begin{bmatrix} 1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \\ 1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \\ 2\nu_2 \\ \sin \frac{\theta}{2} \cos \frac{3\theta}{2} \\ 0 \\ 0 \end{bmatrix}, \tag{A.4}$$

$$\boldsymbol{\sigma} = \frac{K_{II}}{\sqrt{2\pi r}} \begin{bmatrix} -\sin \frac{\theta}{2} (2 + \cos \frac{\theta}{2} \cos \frac{3\theta}{2}) \\ \sin \frac{\theta}{2} \cos \frac{\theta}{2} \cos \frac{3\theta}{2} \\ -2\nu_2 \sin \frac{\theta}{2} \\ \cos \frac{\theta}{2} (1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2}) \\ 0 \\ 0 \end{bmatrix}, \tag{A.5}$$

$$\boldsymbol{\sigma} = \frac{K_{III}}{\sqrt{2\pi r}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \cos \frac{\theta}{2} \\ -\sin \frac{\theta}{2} \end{bmatrix}, \tag{A.6}$$

where ν_2 equals ν (0) under the plane strain (plane stress).

References

- [1] A.M. Aragón, A. Simone, The discontinuity-enriched finite element method, *Internat. J. Numer. Methods Engrg.* 112 (11) (2017) 1589–1613.
- [2] D.M. Tracey, Finite elements for determination of crack tip elastic stress intensity factors, *Eng. Fract. Mech.* 3 (3) (1971) 255–265.
- [3] R.S. Barsoum, On the use of isoparametric finite elements in linear fracture mechanics, *Internat. J. Numer. Methods Engrg.* 10 (1) (1976) 25–37.
- [4] J.R. Shewchuk, What is a good linear finite element? - interpolation, conditioning, anisotropy, and quality measures, *Tech. rep., In Proc. of the 11th International Meshing Roundtable*, 2002.
- [5] Q. Du, D. Wang, Recent progress in robust and quality delaunay mesh generation, *J. Comput. Appl. Math.* 195 (1) (2006) 8–23.
- [6] R. Rangarajan, A.J. Lew, Universal meshes: A method for triangulating planar curved domains immersed in nonconforming meshes, *Internat. J. Numer. Methods Engrg.* 98 (4) (2014) 236–264.
- [7] S. Soghrati, F. Xiao, A. Nagarajan, A conforming to interface structured adaptive mesh refinement technique for modeling fracture problems, *Comput. Mech.* 59 (4) (2017) 667–684.
- [8] J.T. Oden, C.A. Duarte, O.C. Zienkiewicz, A new cloud-based hp finite element method, *Comput. Methods Appl. Mech. Engrg.* 153 (1998) 117–126.
- [9] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, *Internat. J. Numer. Methods Engrg.* 46 (1) (1999) 131–150.
- [10] 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.
- [11] C.A. Duarte, O.N. Hamzeh, T.J. Liszka, W.W. Tworzydło, A generalized finite element method for the simulation of three-dimensional dynamic crack propagation, *Comput. Methods Appl. Mech. Engrg.* 190 (15) (2001) 2227–2262.

- [12] N. Sukumar, N. Moës, T. Belytschko, B. Moran, Extended finite element method for three-dimensional crack modelling, *Internat. J. Numer. Methods Engrg.* 48 (11) (2000) 1549–1570.
- [13] T. Belytschko, N. Moës, S. Usui, C. Parimi, Arbitrary discontinuities in finite elements, *Internat. J. Numer. Methods Engrg.* 50 (2001) 993–1013.
- [14] I. Babuška, U. Banerjee, J.E. Osborn, Survey of meshless and generalized finite element methods: A unified approach, *Acta Numer.* 12 (2003) 1–125.
- [15] N. Moës, E. Béchet, M. Tourbier, Imposing Dirichlet boundary conditions in the extended finite element method, *Internat. J. Numer. Methods Engrg.* 67 (12) (2006) 1641–1669.
- [16] I. Babuška, U. Banerjee, Stable generalized finite element method (SGFEM), *Comput. Methods Appl. Mech. Engrg.* 201–204 (2012) 91–111.
- [17] K. Kergrene, I. Babuška, U. Banerjee, Stable generalized finite element method and associated iterative schemes; application to interface problems, *Comput. Methods Appl. Mech. Engrg.* 305 (2016) 1–36.
- [18] I. Babuška, U. Banerjee, K. Kergrene, Strongly stable generalized finite element method: Application to interface problems, *Comput. Methods Appl. Mech. Engrg.* 327 (2017) 58–92.
- [19] V. Gupta, C. Duarte, I. Babuška, U. Banerjee, A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics, *Comput. Methods Appl. Mech. Engrg.* 266 (2013) 23–39.
- [20] V. Gupta, C. Duarte, I. Babuška, U. Banerjee, Stable GFEM (SGFEM): Improved conditioning and accuracy of GFEM/XFEM for three-dimensional fracture mechanics, *Comput. Methods Appl. Mech. Engrg.* 289 (2015) 355–386.
- [21] Q. Zhang, I. Babuška, U. Banerjee, Robustness in stable generalized finite element methods (SGFEM) applied to poisson problems with crack singularities, *Comput. Methods Appl. Mech. Engrg.* 311 (2016) 476–502.
- [22] 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.
- [23] T.P. Fries, A corrected XFEM approximation without problems in blending elements, *Internat. J. Numer. Methods Engrg.* 75 (5) (2008) 503–532.
- [24] G. Ventura, R. Gracie, T. Belytschko, Fast integration and weight function blending in the extended finite element method, *Internat. J. Numer. Methods Engrg.* 77 (1) (2009) 1–29.
- [25] 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.
- [26] K. Park, J.P. Pereira, C.A. Duarte, G.H. Paulino, Integration of singular enrichment functions in the generalized/extended finite element method for three-dimensional problems, *Internat. J. Numer. Methods Engrg.* 78 (10) (2008) 1220–1257.
- [27] S.J. van den Boom, J. Zhang, F. van Keulen, A.M. Aragón, A stable interface-enriched formulation for immersed domains with strong enforcement of essential boundary conditions, *Internat. J. Numer. Methods Engrg.* (2019) <http://dx.doi.org/10.1002/nme.6139>.
- [28] S. Soghrati, A.M. Aragón, C.A. Duarte, P.H. Geubelle, An interface-enriched generalized fem for problems with discontinuous gradient fields, *Internat. J. Numer. Methods Engrg.* 89 (8) (2012) 991–1008.
- [29] S. Soghrati, Hierarchical interface-enriched finite element method: An automated technique for mesh-independent simulations, *J. Comput. Phys.* 275 (2014) 41–52.
- [30] S. Soghrati, P.H. Geubelle, A 3D interface-enriched generalized finite element method for weakly discontinuous problems with complex internal geometries, *Comput. Methods Appl. Mech. Engrg.* 217 (2012) 46–57.
- [31] A.M. Aragón, S. Soghrati, P.H. Geubelle, Effect of in-plane deformation on the cohesive failure of heterogeneous adhesives, *J. Mech. Phys. Solids* 61 (2013) 1600–1611.
- [32] S. Soghrati, A.R. Najafi, J.H. Lin, K.M. Hughes, S.R. White, N.R. Sottos, P.H. Geubelle, Computational analysis of actively-cooled 3D woven microvascular composites using a stabilized interface-enriched generalized finite element method, *Int. J. Heat Mass Transfer* 65 (2013) 153–164.
- [33] A.C. Ramos, A.M. Aragón, S. Soghrati, P.H. Geubelle, J. Molinari, A new formulation for imposing Dirichlet boundary conditions on non-matching meshes, *Internat. J. Numer. Methods Engrg.* 103 (6) (2015) 430–444.
- [34] S. Soghrati, H. Ahmadian, 3D Hierarchical interface-enriched finite element method: Implementation and applications, *J. Comput. Phys.* 299 (2015) 45–55.
- [35] M. Safdari, A.R. Najafi, N.R. Sottos, P.H. Geubelle, A nurbs-based generalized finite element scheme for 3D simulation of heterogeneous materials, *J. Comput. Phys.* 318 (2016) 373–390.
- [36] S. Soghrati, R.A. Merel, NURBS enhanced HIFEM: A fully mesh-independent method with zero geometric discretization error, *Finite Elem. Anal. Des.* 120 (2016) 68–79.
- [37] A.M. Aragón, J.-F. Molinari, A hierarchical detection framework for computational contact mechanics, *Comput. Methods Appl. Mech. Engrg.* 268 (2014) 574–588.
- [38] A. Salzman, N. Moës, N. Chevaugeon, On use of the thick level set method in 3D quasi-static crack simulation of quasi-brittle material, *Int. J. Fract.* 202 (1) (2016) 21–49.
- [39] H. Si, Tetgen, a Delaunay-based quality tetrahedral mesh generator, *ACM Trans. Math. Softw. (TOMS)* 41 (2) (2015) 11:1–11:36.
- [40] A.M. Aragón, B. Liang, H. Ahmadian, S. Soghrati, On the stability and interpolating properties of the Hierarchical Interface-enriched Finite Element Method, *Comput. Methods Appl. Mech. Engrg.*, Submitted for publication.
- [41] Q. Zhang, S. Cen, *Multiphysics Modeling: Numerical Methods and Engineering Applications*: Tsinghua University Press Computational Mechanics Series, Elsevier, 2015.
- [42] T. Nakamura, D.M. Parks, Antisymmetrical 3-d stress field near the crack front of a thin elastic plate, *Int. J. Solids Struct.* 25 (12) (1989) 1411–1426.

- [43] M. Nagai, T. Ikeda, N. Miyazaki, Stress intensity factor analyses of three-dimensional interface cracks using tetrahedral finite elements, *Comput. Mech.* 51 (5) (2013) 603–615.
- [44] H. Tada, P. Paris, G. Irwin, *The Stress Analysis of Cracks Handbook*, Wiley, 2000.
- [45] I.S. Raju, J.C. Newman Jr, *Three dimensional finite-element analysis of finite-thickness fracture specimens*, NASA Langley Research Center; Hampton, VA, United States, 1977.
- [46] J.P. Pereira, C.A. Duarte, D. Guoy, X. Jiao, hp-Generalized FEM and crack surface representation for non-planar 3-D cracks, *Internat. J. Numer. Methods Engrg.* 77 (5) (2009) 601–633.
- [47] BodyParts3D/Anatomography, <http://lifesciencedb.jp/bp3d/>.
- [48] M.E. Launey, M.J. Buehler, R.O. Ritchie, On the mechanistic origins of toughness in bone, *Annu. Rev. Mater. Res.* 40 (1) (2010) 25–53.
- [49] Y.J. Rho, B.R. Ashman, H.C. Turner, Young's modulus of trabecular and cortical bone material: Ultrasonic and microtensile measurements, *J. Biomech.* 26 (2) (1993) 111–119.
- [50] D.C. Wirtz, N. Schiffers, T. Pandorf, K. Radermacher, D. Weichert, R. Forst, Critical evaluation of known bone material properties to realize anisotropic FE-simulation of the proximal femur, *J. Biomech.* 33 (10) (2000) 1325–1330.
- [51] A.M. Aragón, C.A. Duarte, P.H. Geubelle, Generalized finite element enrichment functions for discontinuous gradient fields, *Internat. J. Numer. Methods Engrg.* 82 (2) (2010) 242–268.