INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2015; **102**:632–670
Published online 18 August 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.4731

Simulating curvilinear crack propagation in two dimensions with universal meshes

Ramsharan Rangarajan[1], Maurizio M. Chiaramonte[2], Michael J. Hunsweck[2], Yongxing Shen[3,4,*,†] and Adrian J. Lew[2,*,†]

[1]*School of Engineering, Brown University, Providence, RI 02912, USA*
[2]*Department of Mechanical Engineering, Stanford University, Stanford, CA 94305-4040, USA*
[3]*Laboratori de Càlcul Numèric, Universitat Politècnica de Catalunya (UPC BarcelonaTech), Barcelona 08034, Spain*
[4]*University of Michigan–Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, China*

SUMMARY

We formulate a class of delicately controlled problems to model the kink-free evolution of quasistatic cracks in brittle, isotropic, linearly elastic materials in two dimensions. The evolving crack satisfies familiar principles—Griffith's criterion, local symmetry, and irreversibility. A novel feature of the formulation is that in addition to the crack path, the loading is also treated as an unknown. Specifically, a scaling factor for prescribed Dirichlet and Neumann boundary conditions is computed as part of the solution to yield an always-propagating and ostensibly kink-free crack and a continuous loading history beyond the initial step. A dimensionless statement of the problem depends only on the Poisson's ratio of a homogeneous material, and is in particular, independent of its Young's modulus and fracture toughness.

Numerical resolution of the formulated problem relies on two new ideas. The first is an algorithm to compute triangulations conforming to cracked domains by locally deforming a *given* background mesh in the vicinity of evolving cracks. The algorithm is robust under mild assumptions on the sizes and angles of triangles near the crack and its smoothness. Hence, a subset, if not the entire family, of cracked domains realized during the course of a simulation can be discretized with the same background mesh; we term the latter a *universal mesh* for such a family of domains. Universal meshes facilitate adopting a discrete representation for the crack (as splines in our examples), preclude the need for local splitting/retriangulation operations, liberate the crack from following directions prescribed by the mesh, and enable the adoption of standard FEMs to compute the elastic fields in the cracked solid. Second, we employ a method specifically designed to approximate the stress intensity factors for curvilinear cracks. We examine the performance of the resulting numerical method with detailed examples, including comparisons with an exact solution of a crack propagating along a circular arc (which we construct) and comparisons with experimental fracture paths. In all cases, we observe convergence of computed paths, their derivatives, and loading histories with refinement of the universal mesh.

Received 20 March 2014; Revised 2 June 2014; Accepted 3 June 2014

KEY WORDS: hydraulic fracture; meshing; *r*-adaptivity

Dedicated to Ted Belytschko on the occasion of his 70th birthday.

1. INTRODUCTION

Computational fracture mechanics play an essential role in applications ranging from the design and reliability analysis of devices and structures to the modeling of hydrocarbon recovery strategies such

*Correspondence to: Yongxing Shen, University of Michigan-Shanghai Jiao Tong University Joint Institute, 800 Dongchuan Rd, Shanghai, China, 200240; Adrian J. Lew, Department of Mechanical Engineering, Stanford University, Stanford, CA 94305-4040, USA.
†E-mail: yongxing.shen@sjtu.edu.cn; lewa@stanford.edu

as hydraulic fracturing. It encompasses the prediction of the onset, propagation, and arrest of cracks and/or catastrophic failure. Once cracks have emerged, the accurate prediction of crack propagation is founded upon two pillars: (1) a fracture propagation criterion corresponding to the material of interest, which defines the crack evolution (modeling), and (2) an algorithm to approximate this evolution, akin to integrating a system of ordinary differential equations, which in turn necessitates accurate approximations to all fields required to evaluate this criterion (numerical approximation). Accurate approximations of the stress field are always needed, but other fields may also be necessary, such as the temperature field in the case of thermal cracking, the plastic deformation in the case of ductile fracture, and the concentration of relevant chemical components in the case of corrosion cracking.

Crack evolution algorithms have remained of low order for the most part, likely because of the challenges involved in accurately computing the necessary fields as the crack evolves, and in perpetually updating the spatial discretization for the evolving computational domain. Additionally, higher-order algorithms to compute the crack evolution may require iterating over possible cracks, a computationally costly step.

Several efficient methods to update the spatial discretization for domains with evolving cracks have been proposed. Many of these methods can be classified on the basis of whether or not basis functions interpolating the displacement field are constructed from a mesh that conforms to the cracked domain. In both cases, evolving cracks are typically represented as curves made of connected straight segments.

In the first approach, the domain with an evolving crack is remeshed with each update to the crack path. By ensuring that the mesh for the domain always conforms to the crack path, any displacement discontinuity along the crack is easily accommodated. At the same time, such methods demand robust and fast remeshing algorithms. These algorithms typically involve local retriangulation near the crack. Consequently, new elements may be added near the crack and existing ones deleted or cut [1–3]. As an alternative to local retriangulation-based algorithms, r-adaptive procedures have also been proposed in [4–7]. Related to the latter are finite element spaces with embedded discontinuities [8, 9].

The second family of methods bypass the need for remeshing by constructing basis functions over the uncracked domain and subsequently modifying them to accommodate the crack. Methods in this category include meshless methods such as the element-free Galerkin method [10, 11], the meshless local Petrov–Galerkin method [12, 13], and the enriched meshless method [14], as well as mesh-based methods such as the strong discontinuity [15–17] and the extended FEM (XFEM) [18, 19].

The XFEM is one of Ted Belytschko's seminal contributions. It has swayed the field of computational fracture mechanics in the last 15 years by inspiring a vast number of related methods, and it has been adopted as a standard tool in some commercial and national security finite element softwares. The XFEM enriches the space of discrete displacement fields with additional basis functions in the vicinity of the crack. These enrichments may be combined with those added to account for the singular stress field at the crack tip. Several variants of XFEM have been formulated to improve shortcomings of the original method, and the list is simply too long to review here. The salient feature of the XFEM, namely, the possibility of propagating cracks along arbitrary directions, has markedly ameliorated the dependence of the computed crack path on the choice of the mesh. In contrast with methods that only allow for prescribed crack propagation directions, such as solely along element faces (e.g., [20] for a cohesive element approach), the family of possibly discrete cracks in XFEM can approximate any rectifiable crack and its length (2D) or area (3D). These quantities are directly relevant to the computation of the surface energy of a crack in brittle fracture, so this feature simplifies the approximation of the potential energy of any given cracked specimen. The drawbacks of XFEMs and some related approaches include the addition of degrees of freedom (DOFs), the cumbersome procedure for numerical integration over partial elements, and the possibly ill-conditioned stiffness matrices due to arbitrarily small elements cut by the crack or near loss of linear independence of the set of basis functions. Methods to alleviate such difficulties have been developed, such as the integration schemes used by [21, 22], the use of only two enrichment basis

DOI: 10.1002/nme

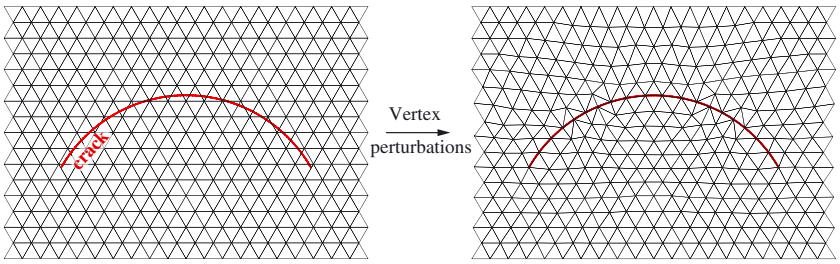


Figure 1. An example to illustrate the idea of perturbing vertices in a nonconforming triangulation to conform it to an embedded crack. The mesh on the left does not conform to the crack. Nevertheless, moving some of its vertices near the crack in a manner we prescribe yields the conforming mesh on the right. In the process, no new vertices are introduced, and the connectivities of the two meshes are identical.

functions near the crack tip in two dimensions, and the requirement on the mesh that all nodes lie either on the crack or a distance away from it [23].

Despite the vast amount of literature that addresses crack propagation problems, very few have systematically studied the convergence of the crack path against analytic solutions (see [24] for the case of a prescribed crack path). This is partly due to the insufficient understanding of what the problem is and what constitutes a solution. In this regard, [25] and [26] aimed at establishing a rigorous problem statement.

As a byproduct, methods based on a regularization of the fracture set have emerged [27–35]. Prominent among them are the so-called *phase field methods*. These methods represent sharp cracks with a diffuse scalar field. Therefore, cracks nucleate, propagate, and bifurcate as an outcome of this scalar field, which has a close physical meaning to the damage field in damage mechanics. The computational cost of phase field methods is high, because a refined mesh is needed to resolve the phase field. By construction, approaches adopting a diffuse representation for cracks lead to an ambiguity in the definition of the crack path, which can be debilitating in applications where an explicit representation is indispensable, such as in the study of hydraulic fracturing [36].

In this work, we regard cracks as sharp interfaces. We address the problem of updating the spatial discretization as the crack evolves by extending to cracked domains an algorithm introduced in [37] for triangulating smooth domains. The main idea behind it consists in perturbing vertices of a given background triangulation so that the resulting mesh conforms to an embedded crack (Figure 1). Specifically, we snap selected vertices of the background mesh to their closest point on the crack. Additionally, we 'relax' vertices in a small neighborhood of the crack to accommodate such snapping. Under conditions detailed in Section 4, it is possible to perturb the same background mesh to yield meshes conforming to the family of cracks that appear and propagate as the loading proceeds, at least for part of the evolution. If this is the case, we term the background mesh a *universal mesh* for such family of cracked domains.

Perturbations of vertices in a universal mesh, described in detail in Section 4, are restricted to a small neighborhood of the crack. Consequently, most triangles in the universal mesh remain unaltered. Perhaps more significantly, vertices are neither added nor deleted, and connectivities of triangles in the mesh are maintained. Hence, the sparsity of data structures (stiffness matrix) remains the same as the crack propagates, except for changes resulting from the duplication of displacement DOFs along the crack and from the possible addition and removal of enrichments.

These are all features shared with r-adaptive methods. Creating robust vertex perturbation-based meshing algorithms has been a lingering challenge, because such perturbations are prone to result in degenerate or inverted triangles, to yield tangled meshes, or to result in badly shaped triangles. Although an analysis of the mesh motion strategy adopted for universal meshes is still outstanding, the results shown here are encouraging, and numerous tests reveal that the method renders meshes with high qualities as the crack evolves. Additionally, the results in [38] guarantee that edges along crack faces do not become arbitrarily small as a result of the vertex perturbation strategy. This

DOI: 10.1002/nme

feature is especially desirable for problems in which a finite element space needs to be constructed over the crack faces to approximate the solution of a PDE therein (e.g., hydraulic fractures).

To ensure success of the meshing algorithm, we have to impose restrictions on the background mesh and on the geometry of the crack. Specifically, we request that the crack be twice continuously differentiable (C^2), that the background mesh be sufficiently refined in the vicinity of the crack, and that certain angles in the triangles intersected by the crack be strictly acute. The rationale behind these assumptions are discussed in Section 4. To maintain the desired regularity of the crack during its evolution, we represent it by fitting cubic splines through the sequence of computed crack tip positions. Alternative representations, for example, C^2-level set functions, could also be employed.

The aforementioned assumptions have some implications on the class of fracture mechanics problems that can be simulated and on the numerical scheme adopted. Arguably, however, the assumptions are not very restrictive. Because the crack is expected to be a C^2 curve, the algorithm to deform the universal mesh is not yet able to robustly accommodate the mesh around a kink, although it is likely that this restriction will be resolved in the near future. In the numerical examples shown here, kinks appear as soon as the crack begins to propagate, but the resulting crack is approximated as a C^2 curve with a very high curvature therein. Bifurcations and crack intersections embody similar issues as kinks and are therefore not yet considered here.

We take advantage of the algorithm to deform a universal mesh to formulate a method to compute the propagation of quasistatic brittle cracks in two dimensions, evolving according to Griffith's criterion and the principle of local symmetry (PLS). To this end, we formulate a class of delicately controlled problems in which the crack is always propagating. This control is achieved by appropriately multiplying the boundary conditions by a scaling factor. The problem then is given a domain with an initial crack, and boundary conditions that depend on the crack length, find *both* the crack path *and* the scaling factor so that the crack is always propagating while satisfying the PLS. One feature of this problem is that the solution does not depend on Young's modulus or the toughness of the material.

The formulation of this problem has two advantages. The first one is that it ostensibly gives rise to crack paths that are kink-free (C^2), at least away from the initial crack tip. This is convenient for the current status of the algorithm to deform the universal mesh. The second advantage is that because the boundary conditions depend on the crack length and the crack is always propagating, the crack length increment, rather than time, serves as the natural evolution parameter. This results in a substantially simpler algorithm; because otherwise, iterations over the crack geometry may be needed to find the crack path at a given time instant.

The evolution of the discrete crack is computed in a rather standard way. The discrete crack is constructed by interpolating a sequence of computed crack tips with a cubic spline. Alternatively, the discrete crack might be regarded as a sequence of connected kink segments. The two curves do not generally coincide, but one interpolates the other, so for the order of accuracy of the algorithm, the distinction is unnecessary. This second perspective is useful to construct the discrete evolution of the crack. The relative angle between consecutive kink segments is computed as in [39], by taking advantage of a first-order asymptotic expansion of the crack tip stress intensity factors for an infinitesimally short added kink [40].

We test the performance of the algorithm to simulate crack propagation using four examples. In all cases, continuous piecewise affine finite element spaces are adopted to approximate the displacement field. In the first example, we use a known analytical solution for the elastic fields in an infinite medium with a circular arc-shaped crack to construct a loading history that is parameterized by crack length and that results in a circular crack path. We take advantage of this solution to assess the order of convergence of the crack path and its first derivative. Other examples include a comparison with a set of experiments from the literature and an example that appears to be commonly studied in the literature on numerical methods for crack propagation problems.

1.1. Organization

We begin with the description of the continuous problem defining the evolution of an always-propagating crack in an isotropic linear elastic solid in Section 2. The algorithm to approximate this

DOI: 10.1002/nme

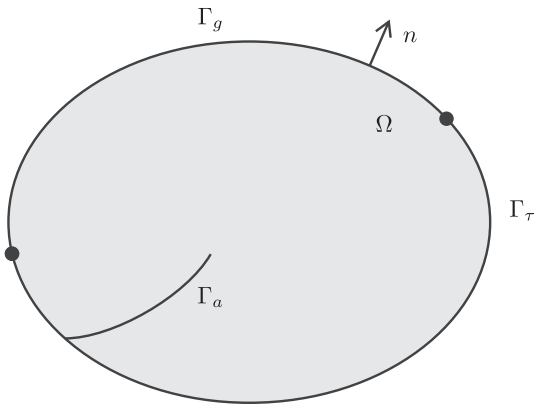


Figure 2. A schematic of the quasistatic crack propagation problem. A closed domain Ω contains a crack Γ_a. The boundary $\partial\Omega$ has prescribed displacement along Γ_g and prescribed traction along Γ_τ.

evolution is presented in Section 3, which uses a mesh of triangles that conforms to the crack to compute approximations to the stress intensity factors at each step of the evolution. These conforming triangulations for cracked domains are constructed by deforming universal meshes with the algorithm discussed in Section 4. Numerical examples demonstrating the evolution of discrete cracks are shown in Section 5. We examine in detail the accuracy of various features of the computed fractures.

2. PROBLEM STATEMENT

We consider a 2D homogeneous linear isotropic elastic medium without body forces initially occupying a closed smooth domain with nonempty interior $\Omega \subset \mathbb{R}^2$. The domain contains an initial crack $\Gamma_a = \Gamma_a(\ell_0) \subset \Omega$, which is a rectifiable and simple curve[‡] with length $\ell_0 > 0$ (Figure 2). We assume that $\Gamma_a(\ell_0)$ is an edge crack, namely, a crack that emanates from $\partial\Omega$ so that $\partial\Omega \cap \Gamma_a(\ell_0)$ is a single point.

We wish to compute the subsequent crack path $\Gamma_a = \Gamma_a(\ell)$ parametrized by its length $\ell > \ell_0$ until it reaches a maximum length $\ell_{\max} > \ell_0$, for any loading history induced by a class of boundary conditions constructed so that the crack is always propagating. To this end, we first introduce the elasticity problem associated with the crack path $\Gamma_a(\ell)$ in Section 2.1 and then describe the crack evolution problem in Section 2.2.

2.1. Elasticity problem

With $\Gamma_a = \Gamma_a(\ell)$, the governing equations of this cracked medium read as follows:

$$\begin{aligned} \nabla \cdot \sigma(\nabla u) &= 0, && \text{in } \Omega_\ell, \\ u &= g(\ell), && \text{on } \Gamma_g, \\ \sigma(\nabla u) \cdot n &= \tau(\ell), && \text{on } \Gamma_\tau, \\ \sigma(\nabla u) \cdot n &= 0, && \text{on } \Gamma_a^+ \cup \Gamma_a^-, \end{aligned} \tag{1}$$

where $\Omega_\ell := \operatorname{int}\Omega \setminus \Gamma_a(\ell)$, $u : \Omega_\ell \to \mathbb{R}^2$ is the displacement field and $\sigma : \mathbb{R}^{2\times 2} \to \mathbb{R}^{2\times 2}$ is the constitutive relation. Abusing notation, we shall also use σ for the stress field $\sigma(\nabla u)$, $\sigma : \Omega_\ell \to \mathbb{R}^{2\times 2}$. The vector field n is either the unit outward normal to Ω (on $\Gamma_\tau \subset \partial\Omega$) or a unit normal to Γ_a. In the latter case, we use Γ_a^+ and Γ_a^- to denote the same point set Γ_a but emphasize that the boundary values (traces) of σ are evaluated from different sides of Γ_a. These different boundary

[‡]A simple curve is one that does not intersect itself. Equivalently, a curve with parametrization $\alpha : [a, b] \to \mathbb{R}^2$ is simple if the mapping α is one-to-one.

DOI: 10.1002/nme

values should be matched with the correct orientation of n so that n points exterior to the region over which σ is evaluated. Subsets of the boundary Γ_g and Γ_τ satisfy $\Gamma_g \cup \Gamma_\tau = \partial\Omega$ and $\Gamma_g \cap \Gamma_\tau = \emptyset$.

With Young's modulus $E > 0$ and Poisson's ratio ν such that $-1 < \nu < 0.5$, the stress σ is related to the displacement u through the constitutive relation:

$$\sigma(\nabla u) = E\overline{\sigma}(\nabla u) = \frac{\nu E}{(1+\nu)(1-2\nu)}(\nabla \cdot u)\,\mathrm{I} + \frac{E}{2(1+\nu)}\left(\nabla u + \nabla u^T\right), \tag{2}$$

where $\overline{\sigma}$ is a dimensionless stress tensor. This is the constitutive relation for plane strain problems. Plane stress cases involve a standard substitution of the elastic constants. The fracture toughness of the material is denoted by K_c.

We consider a class of boundary conditions such that the crack is always propagating. The displacement and traction boundary conditions $g(\ell) : \Gamma_g \to \mathbb{R}^2$ and $\tau(\ell) : \Gamma_\tau \to \mathbb{R}^2$ are of the following forms:

$$\begin{aligned} g(\ell) &= C(\ell)\overline{g}(\ell)K_c/E, \\ \tau(\ell) &= C(\ell)\overline{\tau}(\ell)K_c, \end{aligned}$$

with $\overline{g}(\ell) : \Gamma_g \to \mathbb{R}^2$ and $\overline{\tau}(\ell) : \Gamma_\tau \to \mathbb{R}^2$ known functions of ℓ. The function $C : (\ell_0, \ell_{\max}] \to \mathbb{R}$ serves to linearly scale the boundary conditions so that they can be adjusted to critically load the crack. Roughly speaking, we can say that in the class of problems we consider, the 'shape' of the loading on the boundary is known ($\overline{g}$ and $\overline{\tau}$), but its 'magnitude' has to be solved for. We call the function $C(\ell)$ the 'loading history' and, together with the crack path, is one of the unknowns.

For simplicity, we will also assume that $\overline{g}$ and $\overline{\tau}$ are such that the crack never again gets arbitrarily close to the boundary during the evolution. More precisely, the distance of $\Gamma_a(\ell) \setminus \Gamma_a(\ell_0)$ to $\partial\Omega$ remains greater than some $\epsilon > 0$ for all $\ell \in (\ell_0, \ell_{\max}]$.

The stress intensity factors are defined from the stress field σ as follows:

$$K_I = \lim_{r\to 0}\left(\sqrt{2\pi r}\;\sigma_{\vartheta\vartheta}(r, \vartheta = 0)\right), \quad K_{II} = \lim_{r\to 0}\left(\sqrt{2\pi r}\;\sigma_{r\vartheta}(r, \vartheta = 0)\right),$$

where r and ϑ are polar coordinates of the local crack tip coordinate system, with $r = 0$ at the crack tip and $\vartheta = \pm\pi$ coincident with the tangent to the crack at the crack tip.

Therefore, given $\Gamma_a(\ell)$ and $C(\ell)$, a unique value for each of the stress intensity factor is defined. We denote them with $K_I[\Gamma_a(\ell), C(\ell)]$ and $K_{II}[\Gamma_a(\ell), C(\ell)]$.

Dimensionless problem. A convenient restatement of Equation (1) whose solution is independent of E and K_c is as follows: Find $\overline{u} \in H^1(\Omega_\ell, \mathbb{R}^2)$ such that

$$\begin{aligned} \nabla \cdot \overline{\sigma}(\nabla \overline{u}) &= 0, && \text{in } \Omega_\ell, \\ \overline{u} &= \overline{g}(\ell), && \text{on } \Gamma_g, \\ \overline{\sigma}(\nabla \overline{u}) \cdot n &= \overline{\tau}(\ell), && \text{on } \Gamma_\tau, \\ \overline{\sigma}(\nabla \overline{u}) \cdot n &= 0, && \text{on } \Gamma_a^+ \cup \Gamma_a^-. \end{aligned} \tag{3}$$

It is then straightforward to see that if $\overline{u}$ solves Equation (3), then $u = C(\ell)\overline{u}K_c/E$ solves Equation (1).

For a given crack $\Gamma_a(\ell)$, the stress intensity factors in Equation (3) are the following:

$$\begin{aligned} \overline{K}_I[\Gamma_a(\ell)] &:= \lim_{r\to 0}\left(\sqrt{2\pi r}\;\overline{\sigma}_{\vartheta\vartheta}(r, \vartheta = 0)\right), \\ \overline{K}_{II}[\Gamma_a(\ell)] &:= \lim_{r\to 0}\left(\sqrt{2\pi r}\;\overline{\sigma}_{r\vartheta}(r, \vartheta = 0)\right), \end{aligned}$$

while those corresponding to Equation (1) follow from these as follows:

$$K_I[\Gamma_a(\ell), C(\ell)] = \lim_{r\to 0}\left(\sqrt{2\pi r}\sigma_{\vartheta\vartheta}(r, \vartheta = 0)\right) = K_c C(\ell)\overline{K}_I[\Gamma_a(\ell)], \tag{4a}$$

DOI: 10.1002/nme

$$K_{II}[\Gamma_a(\ell), C(\ell)] = \lim_{r\to 0}\left(\sqrt{2\pi r}\sigma_{r\vartheta}(r, \vartheta = 0)\right) = K_c C(\ell)\overline{K}_{II}[\Gamma_a(\ell)]. \tag{4b}$$

These expressions explicitly show the dependence of the stress intensity factors K_I and K_{II} on the loading history $C(\ell)$.

2.2. The problem to be solved—an always-propagating crack

Next, we focus on a particular class of quasistatic crack propagation problems. This is the set of problems in which the loading history corresponds to a delicately controlled experiment that renders an always-propagating crack. For smooth functions $\overline{g}$ and $\overline{\tau}$, such restriction ostensibly simplifies the problem by suppressing kinks once the crack begins to grow.

From Griffith's criterion, an always-propagating crack implies that

$$K_I(\ell) := K_I\left[\Gamma_a(\ell), C(\ell)\right] = K_{\mathrm{c}}, \quad \forall \ell \geqslant \ell_0. \tag{5}$$

We further assume the crack propagation to be irreversible, that is:

$$\Gamma_a(\ell_1) \subset \Gamma_a(\ell_2), \quad \ell_1 < \ell_2, \tag{6}$$

and to follow the principle of local symmetry (PLS), that is:

$$K_{II}(\ell) := K_{II}\left[\Gamma_a(\ell), C(\ell)\right] = 0, \quad \forall \ell > \ell_0. \tag{7}$$

The problem then consists in finding the crack $\Gamma_a(\ell)$ and the loading history $C(\ell)$ for $\ell \in (\ell_0, \ell_{\max}]$, such that Equations (5), (6), and (7) are satisfied.

It turns out that changing the values of E and K_c does not change the solution $\Gamma_a(\ell)$ and $C(\ell)$, as we show in the succeeding texts. Thus, E and K_{c} need not be specified in the problem statement. As a result, the problem to be solved can be stated as follows:

P: Given Ω, ν, $\Gamma_a(\ell_0)$, $\overline{g}(\ell)$, $\overline{\tau}(\ell)$, and some $\ell_{\max} > \ell_0$, find $C(\ell)$ and $\Gamma_a(\ell)$ such that $K_I[\Gamma_a(\ell), C(\ell)] = 1$ and $K_{II}[\Gamma_a(\ell), C(\ell)] = 0$ for all $\ell \in (\ell_0, \ell_{\max}]$,

Note that this problem includes the standard case in which $\overline{g}$ and $\overline{\tau}$ do not depend on ℓ.

We next see that $\Gamma_a(\ell)$ and $C(\ell)$ that solve problem **P** also satisfy Equations (5), (6), and (7), regardless of the values of E and K_c. First, notice that because they are solutions of problem **P**,

$$\begin{aligned} K_I[\Gamma_a(\ell), C(\ell)] = C(\ell)\overline{K}_I[\Gamma_a(\ell)] = 1, \\ \overline{K}_{II}[\Gamma_a(\ell)] = 0, \end{aligned}$$

for $\ell > \ell_0$. This implies that the solution $C(\ell)$ satisfies the following:

$$C(\ell) = 1/\overline{K}_I[\Gamma_a(\ell)]. \tag{8}$$

Therefore, abusing notation, if now $K_I(\ell)$ and $K_{II}(\ell)$ denote the stress intensity factors for given values of E and K_c, it follows from Equation (4) that $\Gamma_a(\ell)$ also satisfies $K_I(\ell) = K_c C(\ell)\overline{K}_I[\Gamma_a(\ell)] = K_c$ and $K_{II}(\ell) = K_c C(\ell)\overline{K}_{II}[\Gamma_a(\ell)] = 0$ for any $\ell \in (\ell_0, \ell_{\max}]$. These are precisely Equations (5) and (7). This demonstrates that the solutions $\Gamma_a(\ell)$ and $C(\ell)$ are independent of the value of E and K_c and that they can be computed as solutions of problem **P**.

Equation (8) provides an explicit expression for $C(\ell)$. We will use it in the following section to approximate the solution $(\Gamma_a(\ell), C(\ell))$ of problem **P**.

3. DISCRETE CRACK EVOLUTION ALGORITHM

In this section, we discuss the numerical scheme to compute an approximation to the solution of problem **P**. Roughly speaking, the numerical method consists in the following: (1) deforming the

DOI: 10.1002/nme

universal mesh to obtain a mesh conforming to the existing crack; (2) with such a mesh, computing a finite element solution for the displacement field in the cracked medium subject to the prescribed loads; (3) calculating the orientation of the crack increment from the computed solution; and (4) finally advancing the location of the crack tip to the subsequent instant.

We postpone the discussion of part (1) on constructing meshes conforming to cracked domains from a universal mesh until Section 4. For the time being, let us assume that such a mesh is available for each crack path realized during a simulation. We only note that to take advantage of universal meshes, we will adopt a smooth (C^2) representation for the crack, even though it is approximated most easily by a sequence of connected segments.

In the following, we focus on aspects (2)–(4) of the scheme. In Section 3.1, we describe how we represent evolving cracks. The finite element approximation of displacement and stress fields in a cracked medium and the evaluation of the corresponding stress intensity factors are described in Section 3.2. The stepping algorithm to compute the evolution of the discrete crack is discussed in Section 3.3, ultimately yielding an approximation to the solution $(\Gamma_a(\ell), C(\ell))$ of problem **P**.

3.1. Representing the crack

A *discrete crack* Γ_a^h is defined by a set of control points $\{\mathrm{a}_{-p}, \ldots, \mathrm{a}_0, \ldots, \mathrm{a}_n\}$, for integers $p > 1$ and $n \geqslant 0$. The discrete crack Γ_a^h is then defined to be the cubic spline that interpolates these control points and that satisfies the end conditions given by Equation (10). We indicate the dependence of the spline on these control points with $\Gamma_a^h(\mathrm{a}_{-p}, \ldots, \mathrm{a}_n)$.

An *admissible* discrete crack is a discrete crack that satisfies the following:

(i) $\{\mathrm{a}_{-p}, \ldots, \mathrm{a}_0\} \subset \Gamma_a(\ell_0)$, so that the discrete crack formed by these points interpolates the initial crack;
(ii) $\partial\Gamma_a(\ell_0) = \{\mathrm{a}_{-p}, \mathrm{a}_0\}$, so that the end points of the initial crack are interpolated by the discrete crack, and in particular, a_0 is the initial crack tip; and
(iii) $\Gamma_a^h \cap \partial\Omega = \{\mathrm{a}_{-p}\}$, so that discrete cracks do not cross the boundary of Ω. In particular, this implies that $\{\mathrm{a}_{-(p-1)}, \ldots, \mathrm{a}_n\} \subset \operatorname{int}\Omega$.

The last condition is a discrete analog to the condition imposed on Γ_a to remain away from the boundary, Section 2.

3.1.1. Construction of the cubic spline Γ_a^h. Let S_a^h be the union of the segments joining consecutive points in the set $\{\mathrm{a}_{-p}, \ldots, \mathrm{a}_n\}$ (Figure 3), namely:

$$S_a^h(\mathrm{a}_{-p}, \ldots, \mathrm{a}_n) := \bigcup_{i=-p+1}^{n} \overline{\mathrm{a}_{i-1}\mathrm{a}_i}, \tag{9}$$

where $\overline{p_1 p_2}$ denotes the (closed) line segment joining points $p_1, p_2 \in \mathbb{R}^2$. Let $s : S_a^h \to \mathbb{R}$ denote the arc-length parametrization of S_a^h, with $s_{-p} = 0$, and set $s_i := s(\mathrm{a}_i)$ for $-p \leqslant i \leqslant n$. It then follows that $|\mathrm{a}_i - \mathrm{a}_{i-1}| = s_i - s_{i-1}$, for $-p < i \leqslant n$.

Abusing notation, we shall identify the discrete crack Γ_a^h with its parametric representation $\Gamma_a^h : [s_{-p}, s_n] \to \mathbb{R}^2$. The parametric representation of Γ_a^h as a cubic spline is uniquely defined by the following conditions:

(i) The restriction of $\Gamma_a^h(s)$ to each interval $[s_i, s_{i+1}]$ is a cubic polynomial in s, for $-p \leqslant i < n$.
(ii) The curve Γ_a^h interpolates $\{\mathrm{a}_{-p}, \ldots, \mathrm{a}_n\}$, namely,

$$\Gamma_a^h(s_i) = \mathrm{a}_i, \quad \text{for } -p \leqslant i \leqslant n.$$

(iii) The first and second derivatives of Γ_a^h with respect to s are continuous at points $\{\mathrm{a}_{-(p-1)}, \ldots, \mathrm{a}_{n-1}\}$, which ensures that Γ_a^h is C^2.

DOI: 10.1002/nme

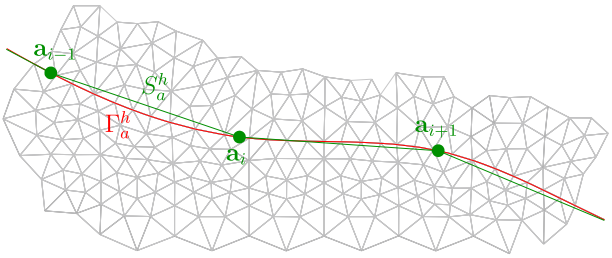


Figure 3. Sketch of a part of a discrete crack Γ_a^h, a cubic spline defined by the points $\{\mathrm{a}_{-p}, \ldots, \mathrm{a}_n\}$, and the chord S_a^h that interpolates the same points. The figure also shows how the universal mesh is conformed to Γ_a^h. The boundary of the isoparametric approximation of the domain $\Omega \setminus \Gamma_a^h$, namely, the collection of edges of the mesh that interpolate Γ_a^h, is indistinguishable from Γ_a^h at this scale.

(iv) End conditions:

$$\frac{d\,\Gamma_a^h}{ds}(\mathrm{a}_{-p}) = \frac{\mathrm{a}_{-(p-1)} - \mathrm{a}_p}{s_{-(p-1)} - s_p}, \qquad \text{and} \qquad \frac{d\,\Gamma_a^h}{ds}(\mathrm{a}_n) = \frac{\mathrm{a}_n - \mathrm{a}_{n-1}}{s_n - s_{n-1}}. \tag{10}$$

Notice that the right-hand sides of the end conditions (Equation (10)) are unit vectors. These end conditions have the trait of depending only on $\{\mathrm{a}_{-(p-1)}, \ldots, \mathrm{a}_{n-1}\}$. As we discuss later in Section 3.3, the crack evolution algorithm is based on deciding a direction for $\mathrm{a}_n - \mathrm{a}_{n-1}$, so these end conditions guarantee that the tangent to the crack at the crack tip coincides with the computed direction. Alternative conditions could be considered, such as matching the exact values of the first and second derivatives at a_{-p}, which may reproduce the initial crack path more accurately. Nevertheless, for simplicity and symmetry, we impose similar conditions (Equation (10)) at both ends of the crack.

One advantage of selecting S_a^h as the parametric domain for the spline is that when $\Delta a := \max_{0<i\leqslant n}(s_i - s_{i-1}) \to 0$, the parametrization Γ_a^h converges to the arc-length parametrization. We take advantage of this fact later for convergence studies.

3.2. Approximating the stress intensity factors

Let an admissible discrete crack Γ_a^h and a triangulation conforming to $\Omega \setminus \Gamma_a^h$ be given (Figure 3). Let us assume that nodes lying on Γ_a^h are duplicated to represent the kinematics on either side of the crack. Then, the stress intensity factors are computed as follows:

(i) We construct a standard finite element space $V_h \subset H^1\left(\Omega \setminus \Gamma_a^h, \mathbb{R}^2\right)$ over the mesh of triangles, consisting of continuous functions that are affine over each triangle, that is, P_1-elements. We then find $\overline{u}_h \in V_h$ that minimizes the potential energy

$$I[v_h] = \frac{1}{2}\int_{\Omega\setminus\Gamma_a^h} \overline{\sigma}(\nabla v_h) : \nabla v_h \, d\Omega - \int_{\Gamma_\tau} \overline{\tau}(\ell) \cdot v_h \, d\Gamma_\tau$$

among all $v_h \in V_h$ that satisfy $v_h = \overline{g}(\ell)$ on Γ_g[§], where $\ell = |S_a^h|$. The discrete solution $\overline{u}_h$ is an approximation to the solution $\overline{u}$ of Equation (3).

(ii) Once $\overline{u}_h$ is known, and therefore, $\overline{\sigma}(\nabla \overline{u}_h)$ is also known, we compute the approximation to the stress intensity factors $\overline{K_I^h}$ and $\overline{K_{II}^h}$ corresponding to this deformation of $\Omega \setminus \Gamma_a^h$, taking advantage of an auxiliary field for an interaction integral specifically designed for curved cracks [41]. We summarize the essential features of this method in Section 6.

[§]More precisely, $v_h = I_h\overline{g}$ on Γ_g, where $I_h\overline{g}$ is the interpolated value of $\overline{g}$.

DOI: 10.1002/nme

(iii) Because we are interested in crack evolutions satisfying $K_I(\ell) = 1$ for all crack lengths, we exploit the linearity of the problem to construct a finite element solution $u_h = C(\ell)\overline{u}_h$[¶], whose stress intensity factors satisfy the following:

$$K_I^h = C(\ell)\overline{K_I^h}, \qquad K_{II}^h = C(\ell)\overline{K}_{II}^h. \tag{11}$$

Then, by setting $C(\ell) = 1/\overline{K_I^h}$ (as in Equation (8)), we have $K_I^h = 1$.

A number of approximations have been made in this computation. Given a crack Γ_a^h, the exact values of the stress intensity factors for Equation (3) with $\ell = \left|\Gamma_a^h\right|$ are $\overline{K}_I\left[\Gamma_a^h\right]$ and $\overline{K}_{II}\left[\Gamma_a^h\right]$ (Section 2.1). However, $\overline{K_I^h}$ and $\overline{K}_{II}^h$ are only approximations to the exact values because of errors that stem from the following: (1) setting $\ell = |S_a^h|$; (2) the finite element approximation of displacements and stresses; and (3) the use of numerical quadrature in evaluating interaction integrals.

In summary, given Γ_a^h, the aforementioned three steps define an algorithm that returns values $\overline{K_I^h}\left[\Gamma_a^h\right]$ and $K_{II}^h\left[\Gamma_a^h\right]$. We use these two values in the next section to compute approximations to the evolution of the crack and the loading history in problem **P**.

3.3. Computing the crack evolution

Next, we specify an evolution of the discrete crack which yields approximations for $\Gamma_a(\ell)$ and $C(\ell)$ in problem **P**. In the following, for $\theta \in \mathbb{R}$, let

$$t(\theta) = \cos\theta\ \mathrm{e}_1 + \sin\theta\ \mathrm{e}_2$$

be the unit vector that forms an angle θ with a reference orthonormal basis $(\mathrm{e}_1, \mathrm{e}_2)$.

(i) Initialization:

(a) Given $\Delta a > 0$, construct an admissible crack $\Gamma_a^{h,0}$ with $n = 0$, such that $|\mathrm{a}_i - \mathrm{a}_{i-1}| < \Delta a$ for $-p < i \leqslant 0$. Hence, $\Gamma_a^{h,0}$ approximates $\Gamma_a(0)$ as $\Delta a \to 0$.
(b) Let $\theta_0 \in [0, 2\pi)$ be the angle that the tangent to the crack at the crack tip forms with respect to the basis $(\mathrm{e}_1, \mathrm{e}_2)$. Hence,

$$t(\theta_0) = \frac{\mathrm{a}_0 - \mathrm{a}_{-1}}{|\mathrm{a}_0 - \mathrm{a}_{-1}|}.$$

(ii) Length stepping:
Given $N \in \mathbb{N}$ such that $\ell_0 + N\Delta a \leqslant \ell_{\max}$, for $n = 0, \ldots, N-1$,

(a) Compute $K_{II}^n = K_{II}^h\left[\Gamma_a^{h,n}\right]$ and $\overline{K_I^n} = \overline{K_I^h}\left[\Gamma_a^{h,n}\right]$. Of course, $K_I^n = 1$ (Section 3.2).
(b) Set $C^n = 1/\overline{K_I^n}$, which we shall use later as an approximation to $C(\ell_0 + n\Delta a)$.
(c) Compute the crack propagation direction according to the PLS. That is, the kink angle at the crack tip is determined with

$$\alpha_{n+1} = 2\arctan\left[\frac{-2K_{II}^n/K_I^n}{1 + \sqrt{1 + 8\left(K_{II}^n/K_I^n\right)^2}}\right], \tag{12}$$

and the crack propagation direction is $t(\theta^{n+1})$, where

$$\theta_{n+1} = \theta_n + \alpha_{n+1}. \tag{13}$$

[¶]It is the finite element solution to a problem whose traction on Γ_τ is $\tau = C\overline{\tau}$, and the displacement on Γ_g is $C\overline{g}$.

DOI: 10.1002/nme

Equation (12) follows from imposing $K_{II} = 0$ on an asymptotic expansion of the stress intensity factors upon the addition of an infinitesimal kink [40], and it is accurate up to the first order with respect to the kink angle α_{n+1}. The expression coincides with that from the maximum hoop stress criterion [39] to first order as well. This is a common feature of various fracture criteria that is discussed in detail in [40]. Equation (12) returns values of α_{n+1} between $\pm 2 \arctan\left(1/\sqrt{2}\right) = \pm 70.5°$.

(d) Compute a new crack tip a_{n+1} with

$$\mathrm{a}_{n+1} = \mathrm{a}_n + \Delta a \; t(\theta_{n+1}).$$

(e) Set $\Gamma_a^{h,n+1} = \Gamma_a^h(\mathrm{a}_{-p}, \ldots \mathrm{a}_0, \ldots, \mathrm{a}_{n+1})$.

The error of $\Gamma_a^{h,n}$ in approximating $\Gamma_a(\ell_0 + n\Delta a)$ is controlled by the two parameters Δa and h. The expectation is that Γ_a^h should converge to Γ_a as $\Delta a \to 0$ and $h < \Delta a$. In Section 5, we showcase through numerical examples that C^n approximates $C(\ell_0 + n\Delta a)$ for $0 \leqslant n < N$ and that $\Gamma_a^{h,n}$ approximates $\Gamma_a(\ell_0 + n\Delta a)$, the solutions of problem **P** in Section 2.2.

3.4. Discussion

Further approximating the geometry. A convenient further approximation to make is to replace the curved domain $\Omega \setminus \Gamma_a^h$ by its isoparametric approximation, as it is standard in finite element computations. In this case, this means that computations are performed on the polygonal domain whose boundary is the union of edges joining nodes on $\partial\Omega \cup \Gamma_a^h$ (Figure 3). The numerical examples in Section 5 were performed in this way.

Discrete irreversibility condition. The construction of the cubic spline cracks does not guarantee that $\Gamma_a^{h,n} \subset \Gamma_a^{h,n+1}$. In fact, $\Gamma_a^{h,n} \not\subset \Gamma_a^{h,n+1}$ most of the times. However, $\Gamma_a^{h,n+1}$ interpolates $\Gamma_a^{h,n}$ at $\{\mathrm{a}_{-p}, \ldots, \mathrm{a}_n\}$.

4. CONFORMING TRIANGULATIONS FROM UNIVERSAL MESHES

In this section, we describe an algorithm for computing a triangulation that conforms to a given discrete crack Γ_a^h by deforming a universal mesh $\mathcal{T}_h$. The algorithm builds on ideas introduced in [37, 38, 42] and [43]. It consists of three steps: (1) extension of the crack on a neighborhood of the crack tip(s); (2) projection of a specific set of nodes onto the discrete crack, to obtain a mesh conforming to it; and (3) directional relaxation of the nodes near the crack to improve the quality of the resulting mesh. Each one of these steps is discussed in the succeeding texts.

4.1. Extension of discrete cracks

The meshing algorithm requires examining the orientation $\mathtt{s} \in \{-1, +1\}$ of the crack at a few nearby vertices, specifically at vertices of a subset of the triangles in $\mathcal{T}_h$ intersected by Γ_a^h. It is well known that the normal to Γ_a^h provides a canonical definition of an orientation within its tubular neighborhood. However, it is possible that a triangle is intersected by Γ_a^h, yet its vertices do not all lie in the tubular neighborhood of Γ_a^h. Such an instance is depicted in Figure 5(c) and can occur irrespective of the mesh size of $\mathcal{T}_h$. Therefore, we seek an extension of $\mathtt{s}$ over neighborhoods of the endpoints a_{-p} and a_n. This is most conveniently achieved by extending the crack Γ_a^h to $\Gamma_a^{h,\mathrm{ext}}$ and using the normal of the latter to define $\mathtt{s}$. Figure 4 illustrates the idea.

In addition to defining $\mathtt{s}$, $\Gamma_a^{h,\mathrm{ext}}$ simultaneously provides a differentiable extension of the local normal to the crack in neighborhoods of its endpoints. Later in Section 4.2.4, we use this extended normal field (and the direction orthogonal to it) in a directional vertex relaxation algorithm to optimize qualities of triangles that are deformed in the universal mesh.

The definition of Γ_a^h as a cubic spline (more generally, as a parametric curve) facilitates a natural extension $\Gamma_a^{h,\mathrm{ext}} : [s_{-p} - \varepsilon_{-p}, s_n + \varepsilon_n] \to \mathbb{R}^2$ for sufficiently small $\varepsilon_{-p}, \varepsilon_n > 0$. The extension is defined by merely evaluating the polynomial expressions for Γ_a^h in the intervals $[s_{-p}, s_{-p+1}]$ and

DOI: 10.1002/nme

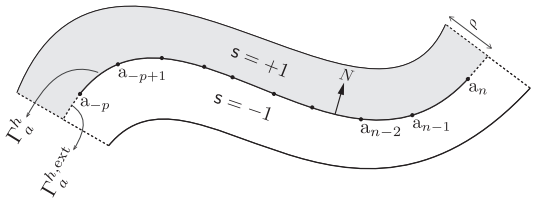


Figure 4. Extension $\Gamma_a^{h,\text{ext}}$ of the cubic spline Γ_a^h to define an orientation $\mathtt{s}$ in a neighborhood of the discrete crack.

$[s_{n-1}, s_n]$ over the extended intervals $[-\varepsilon_{-p} + s_{-p}, s_{-p+1}]$ and $[s_{n-1}, s_n + \varepsilon_n]$, respectively. Then, denoting the (oriented) unit normal to $\Gamma_a^{h,\text{ext}}$ by N, we set the following:

$$\mathtt{s}(x) := \begin{cases} +1 & \text{if } x \in \mathcal{N}^+, \\ -1 & \text{if } x \in \mathcal{N}^-, \\ \text{undefined} & \text{otherwise,} \end{cases} \tag{14a}$$

$$\text{where} \quad \mathcal{N}^+ := \left\{ \Gamma_a^{h,\text{ext}}(\xi) + [0,\rho)N(\xi) \middle| \xi \in (-\varepsilon_{-p} + s_{-p}, s_n + \varepsilon_n) \right\}, \tag{14b}$$

$$\text{and} \quad \mathcal{N}^- := \left\{ \Gamma_a^{h,\text{ext}}(\xi) - (0,\rho)N(\xi) \middle| \xi \in (-\varepsilon_{-p} + s_{-p}, s_n + \varepsilon_n) \right\}. \tag{14c}$$

Here, $\rho > 0$ is a parameter. Notice that $\mathtt{s} = +1$ on $\Gamma_a^{h,\text{ext}}$ by virtue of Equation (14b). The importance of these neighborhoods in the algorithm is that only triangles contained in $\mathcal{N}^\pm$ are deformed in the universal mesh.

The parameters ε_{-p} and ε_n are chosen such that $\mathcal{N}^\pm$ contains the set of triangles intersected by Γ_a^h. Alternatively, such a condition is satisfied for given ε_{-p} and ε_n when $\mathcal{T}_h$ is sufficiently refined near the crack. We have adopted the second point of view in our implementation, by checking that $\mathtt{s}$ is defined at vertices of triangles intersected by Γ_a^h rather than estimating the requisite values for ε_{-p} and ε_n. We expand on these aspects of the implementation in Section 4.2.5.

The parameter ρ in Equation (14b) and (14c) is the radius of the neighborhoods $\mathcal{N}^\pm$ of $\Gamma_a^{h,\text{ext}}$. Its choice is dictated by the curvatures and feature sizes of $\Gamma_a^{h,\text{ext}}$ and is bounded from above by the distance of $\Gamma_a^{h,\text{ext}}$ to its medial axis. We also note that ρ can be smaller than the radius of the tubular neighborhood of Γ_a^h, particularly when the extended crack has large curvatures at the endpoints. When ε_{-p} and ε_n are small, ρ will be close to the radius of the tubular neighborhood of Γ_a^h. The algorithm does not explicitly require estimating ρ. However, it requires that the local mesh size in the vicinity of the crack be sufficiently small compared with ρ.

4.2. Deforming a universal mesh to conform to a crack

4.2.1. Definition of a universal mesh. To introduce the notion of a universal mesh for the crack Γ_a^h, we recall terminology introduced in [37, 38], suitably extended to the case of cracks considered herein. Let π denote the closest point projection onto $\Gamma_a^{h,\text{ext}}$ defined as follows:

$$\pi(x) := \arg \min_{y \in \Gamma_a^{h,\text{ext}}} d(x, y), \tag{15}$$

where $d(x, y)$ is the Euclidean distance between points x and y. We say that a triangle K in $\mathcal{T}_h$ with vertices $\{u, v, w\}$ is *positively cut* by Γ_a^h if

(i) $\mathtt{s}$ is defined at u, v and w;
(ii) $\mathtt{s} = 1$ at two of its vertices and $\mathtt{s} = -1$ at the remaining one; and
(iii) $\mathtt{s}(u) = \mathtt{s}(v) = 1 \Rightarrow \pi(u) \cup \pi(v) \subset \Gamma_a^h$.

The edge in a positively cut triangle having $\mathtt{s} = 1$ at both vertices is called a *positive edge*. The interior angle in a positively cut triangle at the vertex of its positive edge closest to Γ_a^h is called its *conditioning angle*. Figure 5 illustrates these definitions with examples.

DOI: 10.1002/nme

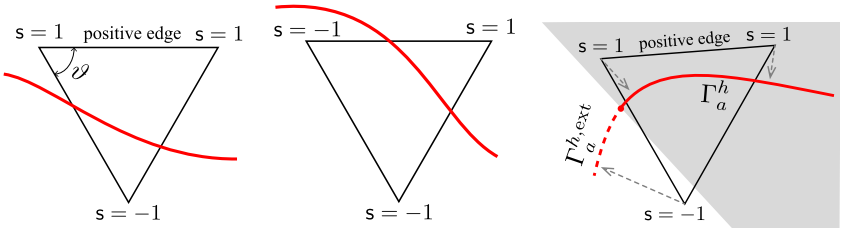


Figure 5. Examples of positively cut triangles, their positive edges, and conditioning angles. Triangles in (a) and (c) are positively cut, and their positive edges are indicated. The conditioning angle of the triangle in (a) is indicated by the angle ϑ. Even though the triangle in (b) is intersected by the crack, it is not positively cut. Example (b) also highlights that the definition of positively cut triangles requires only inspecting s at vertices of the triangle; it does not explicitly require examining triangle-crack intersections. In (c), the tubular neighborhood of Γ_a^h is shaded in gray and contains both vertices of the positive edges. The remaining vertex lies in the tubular neighborhood of the extension $\Gamma_a^{h,\text{ext}}$.

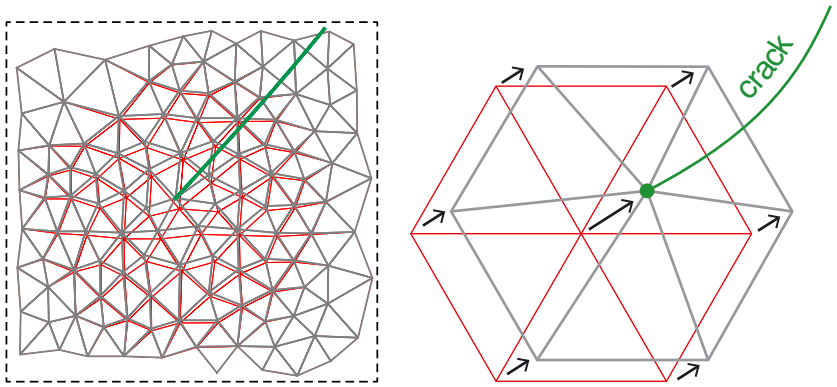


Figure 6. Sketch depicting perturbations near crack tips to satisfy assumption **UM**1. The closest vertex is snapped to the crack tip, and vertices within a small radius are perturbed to accommodate such snapping. Resulting changes in angles and qualities of triangles remain small. Figure (a) shows an example of a mesh in red and its perturbation in gray. Figure (b) depicts the perturbations of triangles in the 1-ring of the vertex snapped to the crack tip.

We say that a triangulation $\mathcal{T}_h$ of Ω is a universal mesh for the crack Γ_a^h with orientation s if

(**UM**1) There is a vertex of $\mathcal{T}_h$ coincident with the endpoints a_{-p} and a_n of Γ_a^h;
(**UM**2) The conditioning angle in each positively cut triangle is strictly acute; and
(**UM**3) The size of each positively cut triangle is sufficiently small; see the discussion in the succeeding texts.

Observe that there are no conformity requirements imposed on a universal mesh. In other words, none of the vertices in a universal mesh need to lie on Γ_a^h. Condition **UM**1 is of course not automatically satisfied by an evolving crack. Instead, it is always possible to add small perturbations to vertices near the crack tips to satisfy it. To this end, we follow the strategy described in [38], which for brevity is not described here but is depicted with an example in Figure 6. The requirement of acute conditioning angles (**UM**2) is most conveniently satisfied by ensuring that triangles in the vicinity of the crack are acute angled. In the examples in Section 5, we adopt meshes with all acute-angled triangles. The perturbations of vertices to satisfy **UM**1 can be performed while keeping all triangles acute angled [38].

The restriction on the mesh size in **UM**3 depends on the curvatures and feature sizes of $\Gamma_a^{h,\text{ext}}$, as well as on the qualities and conditioning angles of positively cut triangles in $\mathcal{T}_h$. Estimates for it

DOI: 10.1002/nme

can be found in [42]. Basically, conditioning angles closer to 90°, larger curvatures, smaller feature sizes, and worse triangle qualities, all lead to smaller required mesh sizes. For a circle immersed in a mesh of equilateral triangles, the upper bound on the mesh size computed in [42] is approximately one fifth of its radius.

4.2.2. Deforming a universal mesh. We recover a mesh conforming to the crack Γ_a^h through a sequence of perturbations of a given universal mesh. A mesh conforms to a discrete crack Γ_a^h if the set of edges interpolating the crack (i.e., both vertices on Γ_a^h) is a nonempty polygonal chain with terminal vertices a_{-p} and a_n. In particular, this implies that a_{-p} and a_n are vertices of the mesh. To separate the kinematics of the crack faces, after conforming the mesh to the crack, we duplicate the nodes lying on Γ_a^h while omitting the node at the propagating tip of the edge crack.

To facilitate describing these transformations/deformations, some notation is helpful. A triangulation $\mathcal{T}_h$ of Ω is specified by the following: (1) an index set I for its vertices; (2) a corresponding list of their coordinates V; and (3) a list of triangle connectivities that are 3-tuples of indices in I all oriented in the same way. We denote this identification by $\mathcal{T}_h = (V, I, C)$. Cartesian coordinates of a vertex with index $i \in I$ are denoted by $v_i \in V$. The star of a vertex $i \in I$, denoted $\mathrm{Star}(i)$, is the set of triangles in $\mathcal{T}_h$ that have i as one of their vertices. In the literature, $\mathrm{Star}(i)$ is also commonly referred to as triangles in the 1-ring of i. Of course, a triangulation here is to be understood in the usual sense, namely, the intersection between the closure of two triangles is either empty, a common vertex, or a common edge.

Let $\mathcal{T}_h^c$ denote the mesh conforming to Γ_a^h computed by perturbing a universal mesh $\mathcal{T}_h$ for Ω. The two triangulations differ only in the location of a few vertices. This relation between the two meshes is succinctly summarized as $\mathcal{T}_h^c = (V', I, C)$, conveying that both meshes have identical index sets and connectivities. In particular, vertices are neither introduced nor removed in the algorithm.

The algorithm to deform the mesh effectively reduces to deciding which vertices in $\mathcal{T}_h$ to perturb and how to move them, in such a way that the resulting mesh $\mathcal{T}_h^c$ is valid (no inverted triangles and conforming to the domain) and has triangles with good quality. The strategy is presented in two parts. First, we identify vertices that are projected onto the crack. Then, we describe a 'relaxation' step that moves vertices near the crack to improve the quality of the resulting mesh.

4.2.3. Projections onto the crack. Let Γ_+^h denote the set of all positive edges in $\mathcal{T}_h$, and $I_+ \subset I$ be the set of all vertices in Γ_+^h. Algorithm 1 provides the steps in identifying Γ_+^h. In the following, we identify a subset of edges $\tilde{\Gamma}_+^h$ in $\mathcal{T}_h$ whose vertices $\tilde{I}_+ \subset I_+$ are mapped to their closest point in Γ_a^h, as illustrated in Figure 7. Our definition of $\tilde{\Gamma}_+^h$ and the resulting set of vertices $\tilde{I}_+$ are such that

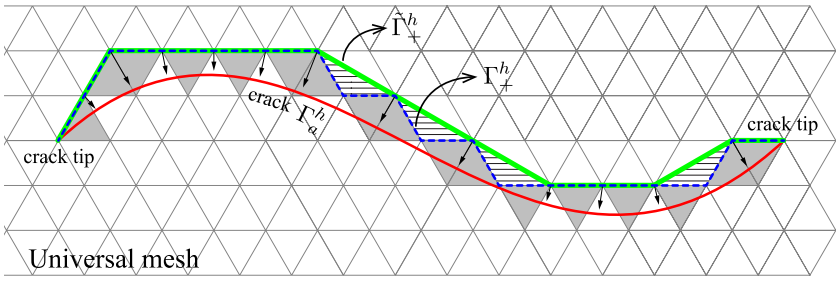


Figure 7. Projecting vertices of a universal mesh onto a crack. A crack, shown in red, is immersed in a universal mesh that has a vertex at each crack tip. Triangles shaded in gray are positively cut (and equilateral), and the set of edges Γ_+^h shown in dashed blue are their positive edges. Notice that Γ_+^h satisfies conditions **A**1–**A**2 in Section 4.2.3 but not **A**3 because the hatched triangles in the mesh have all three vertices in Γ_+^h. The set $\tilde{\Gamma}_+^h$ shown in green and computed in Algorithm 2, however, satisfies all three. Projecting its vertices onto their closest point in the crack results in a mesh that conforms to the latter, without mapping all three vertices of any triangle onto the crack.

DOI: 10.1002/nme

(**A**1) $\pi : \tilde{\Gamma}_+^h \to \Gamma_a^h$ is a homeomorphism;
(**A**2) vertices coincident with crack tips are included in $\tilde{I}_+$, namely, $\mathrm{a}_{-p}, \mathrm{a}_n \in \tilde{\Gamma}_+^h$; and
(**A**3) no triangle in $\mathcal{T}_h$ has all three vertices in $\tilde{I}_+$. Equivalently, $(i, j, k) \in C \Rightarrow \{i, j, k\} \not\subset \tilde{I}_+$.

Because $\mathcal{T}_h$ is a universal mesh for Γ_a^h, we know $\pi : \Gamma_+^h \to \Gamma_a^h$ is a homeomorphism [42]. When $\mathcal{T}_h$ is sufficiently refined at a_{-p} and a_n, both a_{-p} and a_n are necessarily vertices of positive edges, that is, $\mathrm{a}_{-p}, \mathrm{a}_n \in \Gamma_+^h$ (see [44, Chapter 4]). Hence, Γ_+^h and its vertices I_+ satisfy the first two requirements **A**1 and **A**2. If by coincidence, no triangle in $\mathcal{T}_h$ has all three vertices in I_+, we can indeed set $\tilde{\Gamma}_+^h = \Gamma_+^h$ and $\tilde{I}_+ = I_+$. This is however not the case in general, as the example in Figure 7 shows.

Condition **A**3 is essential to prevent all three vertices of a triangle from being mapped onto the crack. In the example in Figure 7 for instance, projecting vertices in I_+ onto the crack results in all three vertices of the hatched triangles being mapped onto the crack. The qualities of the resulting triangles depend exclusively on the local shape of the crack and can be arbitrarily poor.

It is in fact possible to identify $\tilde{I}_+ \subseteq I_+$ that guarantees **A**3. To this end, let $\mathcal{E}_+$ denote the set of polygonal chains in $\mathcal{T}_h$ whose vertices are included in I_+ and with terminal vertices a_{-p} and a_n. This collection is nonempty because $\Gamma_+^h \in \mathcal{E}_+$. Borrowing terminology from the theory of planar graphs, we identify $\tilde{\Gamma}_+^h \in \mathcal{E}_+$ as any 'triangle-free' chain in $\mathcal{E}_+$. As the name suggests, we say that a chain $\gamma \in \mathcal{E}_+$ with vertices $\left(\mathrm{a}_{-p} = v_{i_0}, v_{i_1}, \ldots, v_{i_m} = \mathrm{a}_n\right)$ is triangle-free if $(i, j, k) \in C \Rightarrow \{i, j, k\} \not\subset \{i_0, i_1, \ldots, i_n\}$.

There may be multiple triangle-free chains in $\mathcal{E}_+$, and any one among them can be chosen as $\tilde{\Gamma}_+^h$. It appears that a canonical definition for $\tilde{\Gamma}_+^h$ would be as the shortest chain in $\mathcal{E}_+$, where the length of a chain is given by the number of edges comprising it. We choose a more efficiently computable solution given by algorithm 2, which identifies $\tilde{I}_+$ as an ordered list of vertices.

The set $\tilde{\Gamma}_+^h$ follows as the set of edges in $\mathcal{T}_h$ that joins consecutive vertices in $\tilde{I}_+$.

Input: Crack Γ_+^h and orientation s
Input: Universal mesh $\mathcal{T}_h = (V, I, C)$
$\Gamma_h \leftarrow \emptyset$
foreach $(i_1, i_2, i_3) \in C$ **do**
  $I_\pm \leftarrow \emptyset$
  for $j = 1$ **to** 3 **do**
    if s *is defined at* v_{i_j} **then**
      if $\mathsf{s}(v_{i_j}) = 1$ & $\pi(v_{i_j}) \in \Gamma_+^h$ **then** $I_+ \leftarrow I_+ \cup \{i_j\}$
      else if $\mathsf{s}(v_{i_j}) = -1$ **then** $I_- \leftarrow I_- \cup \{i_j\}$
    end
  end
  if $\#I_+ = 2$ & $\#I_- = 1$ **then**
    $\{k, \ell\} \leftarrow I_+$
    Update $\Gamma_+^h \leftarrow \Gamma_+^h \cup \overline{v_k v_\ell}$
  end
end
return Γ_+^h

Algorithm 1: `Identify positive edges`

The set $\tilde{\Gamma}_+^h$ as defined earlier in the text satisfies **A**2 and **A**3 essentially by definition. We expect that it satisfies **A**1 because of the corresponding property of the set of positive edges Γ_+^h, but a rigorous demonstration is pending. We have verified it for numerous test cases, including all the examples in Section 5.

DOI: 10.1002/nme

Input: Universal mesh $\mathcal{T}_h = (V, I, C)$ and set of positive edges Γ_+^h
$(i_1, i_2 \ldots, i_n) \leftarrow$ polygonal chain representation of Γ_+^h
$\tilde{I}_+ = \{i_1\}$
$j = 1$
while $j \neq n$ **do**
  $k \leftarrow j + 1$
  for $\ell = j + 2$ **to** n **do**
    if i_j *and* i_ℓ *are vertices of an edge in* $\mathcal{T}_h$ **then** $k \leftarrow \ell$
  end
  Update $\tilde{I}_+ \leftarrow \tilde{I}_+ \cup \{i_k\}$
  $j \leftarrow k$
end
return $\tilde{I}_+$

Algorithm 2: `Triangle-free vertices of positive edges`

4.2.4. Directional vertex relaxation. The next set of perturbations are aimed at controlling the qualities of triangles in the 1-rings of vertices in $\tilde{I}_+$, to accommodate their projection onto the crack. A small set of vertices $I_R \subset I$ in the vicinity of $\tilde{I}_+$ are relaxed following an optimization-based directional vertex relaxation algorithm introduced in [43].

A good candidate for I_R in a mesh with relatively uniformly sized triangles in the vicinity of the crack is set as follows:

$$\left\{i \in I \setminus \tilde{I}_+ \mid d(v_i, \Gamma) \leqslant r, v_i \in \mathcal{N}^{\pm}\right\}, \tag{16}$$

where r equals a few multiples of the mesh size of $\mathcal{T}_h$ around the crack. Observe that I_R is disjoint from $\tilde{I}_+$ because vertices in the latter are projected onto the crack. We require that vertices in I_R belong to $\mathcal{N}^{\pm}$ because their direction of relaxation is along the local normal to $\Gamma_a^{h,\text{ext}}$, which is well defined in $\mathcal{N}^{\pm}$.

In adaptively refined meshes, however, Equation (16) inevitably results in either too many or too few vertices being relaxed. A definition suitable for such meshes is based on the notion of the graph distance to $\tilde{I}_+$. Let $gd(i, j)$ denote the graph distance between vertices i and j, defined as the length of the shortest polygonal chain in $\mathcal{T}_h$ with terminal vertices i and j. Then, for a given positive integer N_{relax}, we set the following:

$$I_R := \left\{i \in I \setminus \tilde{I}_+ \mid gd\left(i, \tilde{I}_+\right) \leqslant N_{\text{relax}}, v_i \in \mathcal{N}^{\pm}\right\}, \tag{17}$$

where $gd\left(i, \tilde{I}_+\right) := \min_{j \in \tilde{I}_+} gd(i, j)$.

We presume that a relaxation direction is prescribed for each vertex in I_R, an assumption that is crucial for the simplicity and efficiency of the optimization algorithm. It is also a natural one in the current context, because the normal to the crack at the closest point projection of a vertex serves as an obvious choice. Specifically, the relaxation direction is defined for vertices in I_R as follows:

$$d_n(x) := N(\pi(x)), \tag{18}$$

the normal to $\Gamma_a^{h,\text{ext}}$ at the closest point projection of point x. Equation (18) is well defined over $\mathcal{N}^{\pm}$, and hence, over I_R, because by definition of Equation (17) (or Equation (16)), all vertices in I_R belong to $\mathcal{N}^{\pm}$.

It is also beneficial to relax vertices along the local tangential direction to $\Gamma_a^{h,\text{ext}}$, which is the direction orthogonal to d_n. This is especially so in meshes having conditioning angles that are close to 90°. In such meshes, tangential relaxations mainly help improve the qualities of positively cut triangles whose positive edges are projected to short lengths. Details of tangential relaxations can be found in [43]. We use both normal and tangential relaxations in our numerical examples.

DOI: 10.1002/nme

The perturbation distances for vertices in I_{R} along their respective relaxation directions are computed to optimize the qualities of triangles in their 1-rings. For a given quality metric Q, the perturbation distance λ_i for $i \in I_{\mathrm{R}}$ is computed as follows. Let $\{K_{i_j}\}_{j=1}^{m}$ be the triangles in the Star(i). Let $q_{i_j}(\lambda)$ denote the quality Q of triangle K_{i_j} when its vertex i is perturbed to $v_i + \lambda\, d_n(v_i)$. We set the following:

$$\lambda_i := \arg\max_{\lambda \in \mathbb{R}} \min_{1 \leqslant j \leqslant m} q_{i_j}(\lambda). \tag{19}$$

Computing the perturbation distance in Equation (19) in fact only requires the solution of scalar equations. In our examples, we choose the metric Q introduced in [45] that assigns to triangle K with signed area Δ_K and edges with lengths ℓ_1, ℓ_2, ℓ_3, the quality

$$Q(K) := \frac{4\sqrt{3}\Delta_K}{\ell_1^2 + \ell_2^2 + \ell_3^2}. \tag{20}$$

The metric in Equation (20) is related if not equivalent to other commonly used metrics. For this specific choice of Q, resolution of Equation (19) simplifies further to computing roots of quadratic and cubic polynomials as explained in [43].

Algorithm 3 summarizes the relaxation procedure with the choice of d_n in Equation (18) as the relaxation direction. The relaxation algorithm is performed iteratively, with each iteration consisting in relaxing vertices in I_{R} in a specific order. Anticipating that triangles in the vicinity of the crack require most improvement in quality because of projecting vertices in $\tilde{I}_+$ onto Γ_a^h, we relax vertices in I_{R} in ascending order of their Euclidean distance from $\Gamma_a^{h,\mathrm{ext}}$.

Convergence of these relaxation iterations is examined in [43]. For the examples in Section 5, we perform a fixed number of iterations.

Input: Triangulation $\mathcal{T}_h = (V, I, C)$
Input: Extended crack $\Gamma_a^{h,\mathrm{ext}}$
Input: Ordered list of vertices to relax I_{R}
Input: Number of relaxation iterations N_{iter}
for *iter* $= 1$ **to** N_{iter} **do**
  for $j = 1$ **to** *Length*(I_R) **do**
    $i \leftarrow j^{\mathrm{th}}$ vertex in I_{R}
    Compute relaxation direction $d_n(v_i)$ using (18)
    $\{K_{i_\ell}\}_\ell \leftarrow$ Star(i)
    Compute perturbation distance $\lambda_i := \arg\max_\lambda \min_\ell q_{i_\ell}(\lambda)$
    Update $v_i \leftarrow v_i + \lambda_i\, d_n(v_i)$
  end
end
return Updated set of coordinates V

Algorithm 3: `Relax vertices`

4.2.5. Remarks on implementation.

Choice of algorithmic parameters. At first sight, it may appear that the algorithm requires explicitly choosing parameters $\varepsilon_{-p}, \varepsilon_n$ and ρ. After all, the definition of the orientation s depends on them and is required for identifying the following: (1) the set of positively cut triangles and (2) the set I_{R} of vertices to be relaxed. We discuss in the succeeding texts that we can avoid estimating these parameters when the background mesh is sufficiently refined around the crack.

Let us examine the identification of positively cut triangles described in Section 4.2.1. Given triangle K with vertices $\{u, v, w\}$, Equation (14) reveals that condition (1) is equivalent to checking if $\{u, v, w\} \subset \mathcal{N}^{\pm}$.

DOI: 10.1002/nme

Such a check explicitly requires defining the sets $\mathcal{N}^{\pm}$, which in turn entails choosing parameters ε_{-p}, ε_n, and ρ. Additionally, it may also be computationally expensive to perform in practice.

Notice however that a positively cut triangle is necessarily intersected by Γ_a^h and that the collection of triangles intersected by Γ_a^h is independent of ρ and $\varepsilon_{-p}, \varepsilon_n$. Therefore, the collection of positively cut triangles is identified by checking (2) and (3) over triangles that are intersected by Γ_a^h. It is not necessary to precisely compute triangle-crack intersections. Instead, it suffices to check (2) and (3) over a collection that merely includes all triangles intersected by Γ_a^h, which can be determined efficiently using local bounding boxes around the crack for instance.

The aforementioned observations show that for fixed ε_{-p}, ε_n, and ρ (whatever they may be), a sufficiently small mesh size around Γ_a^h ensures that all vertices of triangles intersected by the crack belong to the sets $\mathcal{N}^{\pm}$. Hence, in a practical implementation, refinement around the crack avoids the need to specify ε_{-p}, ε_n, and ρ while identifying positively cut triangles. When ε_{-p}, ε_n, and ρ are explicitly chosen, it may be possible to additionally estimate the necessary refinement of the mesh around the crack. This is an important step in automating the algorithm but is one that is still outstanding.

Similarly, identifying the set I_{R} requires checking if vertices within a few mesh sizes from the crack (based on Euclidean or graph distance) are included in $\mathcal{N}^{\pm}$. For given ε_{-p}, ε_n, and ρ, this will be the case when the background mesh is sufficiently refined.

Computation of the graph distance. Identifying the set of vertices to relax according to Equation (17) requires computing the graph distances of vertices in the vicinity of the crack to vertices in $\tilde{I}_+$. These distances can be computed, for instance, using Dijkstra's algorithm with a graph-based representation for the triangulation $\mathcal{T}_h$ [46].

As illustrated in Figure 8, a simpler alternative is possible in our case. It consists in recursively gathering vertices in 1-rings, starting from the 1-rings of vertices in $\tilde{I}_+$. Specifically, let $\overset{\star}{\mathsf{v}}(\{i\}) \subset I$ denote the set of vertices of triangles in $\mathrm{Star}(i)$, and define $\overset{\star}{\mathsf{v}}(A \subseteq I) := \cup_{i \in A} \overset{\star}{\mathsf{v}}(\{i\})$. Then, notice that

$$\{i \in I \,:\, gd(i, A) \leqslant 1\} = \overset{\star}{\mathsf{v}}(A) \quad \text{for any } A \subset I. \tag{21}$$

In particular, choosing $A = \tilde{I}_+$ in Equation (21) shows that $I_{\mathrm{R}} = \overset{\star}{\mathsf{v}}(\tilde{I}_+) \setminus \tilde{I}_+$ for the choice $N_{\mathrm{relax}} = 1$.

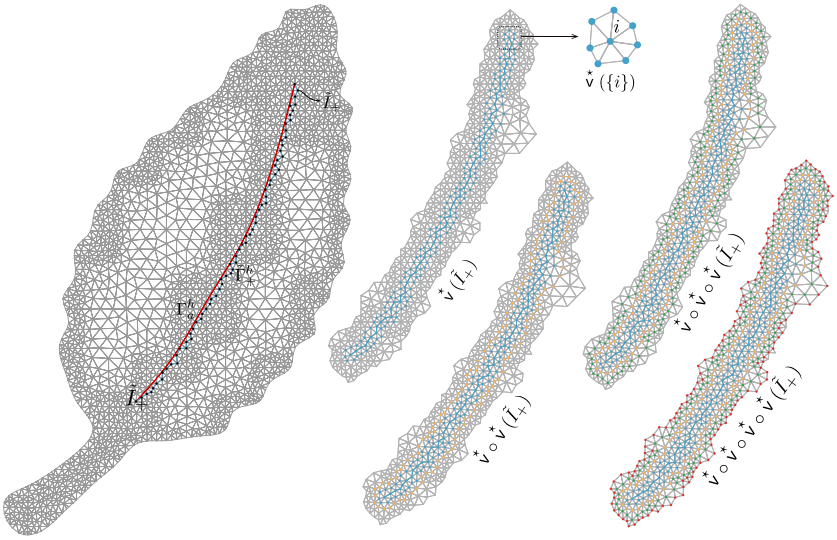


Figure 8. Identifying vertices around $\tilde{I}_+$ based on the graph distance. For the example discussed in Section 4.3, the figure illustrates identifying vertices to be relaxed according to Equation (17) by recursively gathering vertices in 1-rings. For example, the set $\overset{\star}{\mathsf{v}} \circ \overset{\star}{\mathsf{v}} \circ \overset{\star}{\mathsf{v}}(\tilde{I}_+) \setminus \tilde{I}_+$ equals I_{R} for the choice $N_{\mathrm{relax}} = 3$.

DOI: 10.1002/nme

A choice $N_{\text{relax}} > 1$ only requires a recursive application of Equation (21) because we have the following relation:

$$\left\{i \in I \,:\, gd\left(i, \tilde{I}_+\right) \leqslant N_{\text{relax}}\right\} = \underbrace{\overset{\star}{\mathtt{v}} \circ \overset{\star}{\mathtt{v}} \circ \ldots \circ \overset{\star}{\mathtt{v}}}_{N_{\text{relax}} \text{ times}}\left(\tilde{I}_+\right). \tag{22}$$

Equation (22) holds for any $A \subset I$, but has been stated for $\tilde{I}_+$, the set of interest. Hence, I_{R} in Equation (17) is computed as follows:

$$I_{\text{R}} = \left\{\underbrace{\overset{\star}{\mathtt{v}} \circ \overset{\star}{\mathtt{v}} \circ \ldots \circ \overset{\star}{\mathtt{v}}}_{N_{\text{relax}} \text{ times}}\left(\tilde{I}_+\right)\right\} \setminus \tilde{I}_+. \tag{23}$$

Equation (23) also explicitly reveals the recursive nature of its implementation. To facilitate an efficient implementation, we note that because

$$\underbrace{\overset{\star}{\mathtt{v}} \circ \overset{\star}{\mathtt{v}} \circ \ldots \circ \overset{\star}{\mathtt{v}}}_{m \text{ times}}\left(\tilde{I}_+\right) \subseteq \underbrace{\overset{\star}{\mathtt{v}} \circ \overset{\star}{\mathtt{v}} \circ \ldots \circ \overset{\star}{\mathtt{v}}}_{m+1 \text{ times}}\left(\tilde{I}_+\right),$$

vertices are tagged once their 1-rings have been inspected to avoid revisiting them during the recursive aggregation in Equation (23).

4.3. Meshing algorithm

Algorithm 4 summarizes the steps to determine a mesh $\mathcal{T}_h^c$ conforming to Γ_a^h by projecting and relaxing vertices of a universal mesh $\mathcal{T}_h$. Three integer parameters are required as inputs:

(i) N_{relax}: graph distance-based radius to identify the set of vertices to relax I_{R};
(ii) N_{steps}: number of steps in which vertices in $\tilde{I}_+$ are projected onto the crack; and
(iii) N_{iter}: number of iterations of vertex relaxation.

It is indeed possible to estimate each of these parameters during the execution of the algorithm, but we have retained them as inputs for simplicity and to accurately reflect the implementation used in the examples in Section 5. The reasoning behind projecting vertices in $\tilde{I}_+$ onto the crack in multiple steps is discussed in [43]. While vertices in $\tilde{I}_+$ are finally snapped to their closest point in Γ_a^h irrespective of the choice of N_{steps}, the locations of vertices in I_{R} do depend on N_{steps}. Algorithm 4 includes additional relaxation iterations once vertices in $\tilde{I}_+$ are projected onto the crack because a convergence criterion for such iterations has not been discussed here. Relaxation iterations can be terminated if qualities of triangles are deemed to have converged in a sense that is discussed in [43].

We conclude this section with the example shown in Figure 9 demonstrating the application of Algorithm 4 and examining the qualities of triangles in the computed mesh. The figure shows a leaf-shaped domain with its midrib representing a crack in it. The boundary of the domain is a closed cubic spline interpolating 61 points, while the midrib is a natural cubic spline interpolating 6 points. For illustrative purposes, we constructed an adaptively refined triangulation to serve as a universal mesh for the domain by tiling an adaptively refined quadtree with stencils of acute-angled triangles [47]. The algorithm described in [37, 43], which closely resembles Algorithm 4, yields a triangulation conforming to the domain boundary that serves as a universal mesh for the midrib. Then, application of Algorithm 4 yields the final mesh for the domain conforming to the midrib. The figure also shows statistics comparing the qualities of triangles before and after the application of Algorithm 4.

DOI: 10.1002/nme

Input: Crack Γ_+^h and extension $\Gamma_a^{h,\text{ext}}$
Input: Universal mesh $\mathcal{T}_h = (V, I, C)$ for Γ_a^h
Input: Radius of relaxation neighborhood N_{relax}
Input: Number relaxation iterations N_{iter}
Input: Number of projection steps N_{steps}
$\mathsf{s} \leftarrow$ Orientation defined by (14)
$\Gamma_+^h \leftarrow$ `Identify positive edges`$((V, I, C), \Gamma_+^h, \mathsf{s})$
$\tilde{I}_+ \leftarrow$ `Triangle-free vertices of positive edges`$((V, I, C), \Gamma_+^h)$
$I_{\text{R}} \leftarrow \{i \in I \setminus \tilde{I}_+ : gd(i, \tilde{I}_+) \leq N_{\text{relax}}, v_i \in \mathcal{N}^{\pm}\}$
Sort vertices in I_{R} in ascending order of distances to $\Gamma_+^{h,\text{ext}}$
$\lambda \leftarrow 1/N_{\text{steps}}$
for $j = 1$ **to** N_{steps} **do**
  foreach $i \in \tilde{I}_+$ **do**
    Update $v_i \leftarrow (1 - \lambda j)v_i + \lambda j \pi(v_i)$
  end
  for $k = 1$ **to** N_{iter} **do**
    $V \leftarrow$ `Relax vertices`$((V, I, C), I_{\text{R}}, \Gamma_a^{h,\text{ext}})$
  end
end
for $k = 1$ **to** N_{iter} **do**
  $V \leftarrow$ `Relax vertices`$((V, I, C), I_{\text{R}}, \Gamma_a^{h,\text{ext}})$
end
$\mathcal{T}_h^c \leftarrow (V, I, C)$
$i_n \leftarrow$ Vertex in $\mathcal{T}_h^C$ at crack tip a_n
Duplicate nodes $\tilde{I}_+ \setminus \{i_n\}$ in $\mathcal{T}_h^c$
return Triangulation $\mathcal{T}_h^c$

Algorithm 4: `Triangulate domain` Ω `with crack` Γ

4.4. Discussion

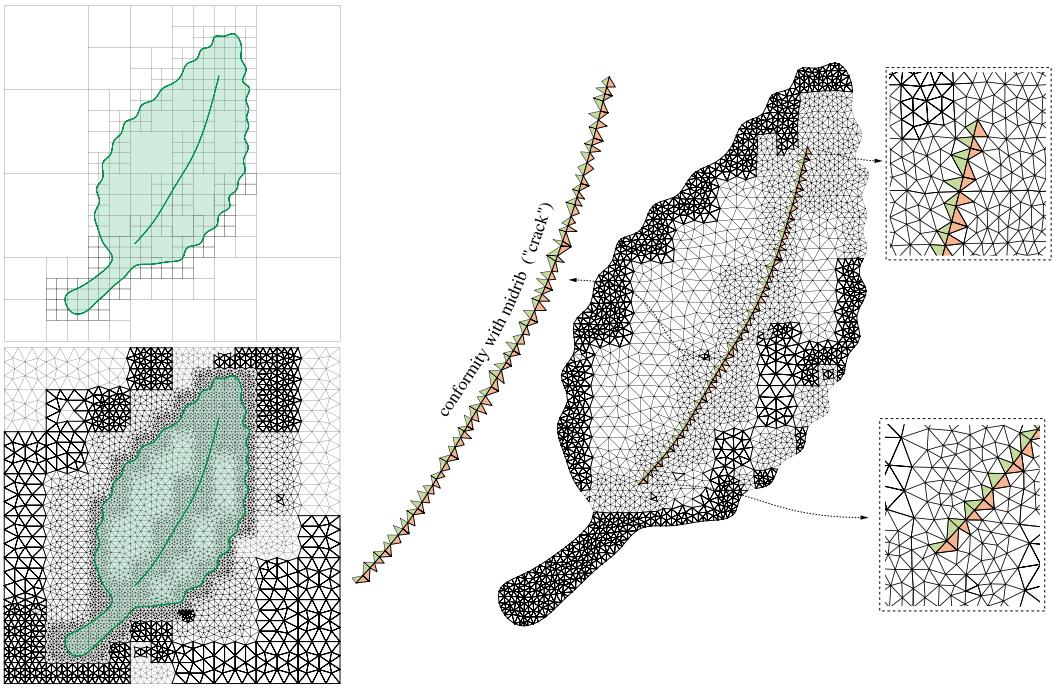
Robustness of the meshing algorithm for computed crack paths. The universal mesh $\mathcal{T}_h$ plays the essential role of returning a mesh conforming to the crack at each time step, which is used in step (3.3) in Section 3.3. A crucial question is under which sufficient conditions deforming the universal mesh to conform to the crack will return a valid mesh with all elements of a guaranteed minimum quality. On the basis of previous work for meshing domains [37, 42], one of the requirements is that at any point $x \in \Gamma_a^h$, the size of the elements in $\mathcal{T}_h$ near x be smaller than a fraction of the radius of curvature at x. This is the motivation behind the introduction of a crack defined by a cubic spline, sidestepping the kinks of cracks formed by a collection of connected segments such as S_a^h defined in (9). In fact, the algorithm to deform the mesh in Section 4.2 is not well defined for cracks with kinks. The smoothness of the crack also enables the construction of a convenient auxiliary field for the interaction integral, used for the computation of stress intensity factors (Section 6)‘.

For the crack evolution algorithm to be efficient, it is necessary to set $h = O(\Delta a)$, or as we do in the examples later, $\Delta a / h$ constant. Hence, we need to have the following:

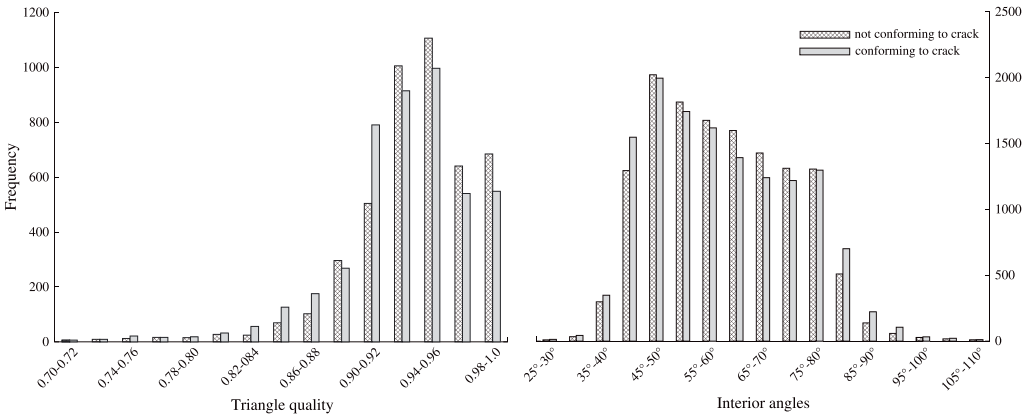
$$\liminf_{\Delta a \to 0} \frac{H}{\Delta a} > 0, \tag{24}$$

where $H = H\left(\Gamma_a^h\right)$ denotes the minimum radius of curvature of Γ_a^h. This means that either H is bounded away from zero independently of Δa or H tends to zero no faster than Δa. Otherwise, because the mesh deformation algorithm requires $h \leqslant cH$ for some constant $c > 0$ (see [42] and Section 4.2), if $H = o(\Delta a)$, we would need to have $h = o(\Delta a)$, an undesirable scaling.

DOI: 10.1002/nme



(a) Triangulation of a leaf-shaped domain with a mid-rib (crack). A universal mesh for the domain is constructed by tiling an adaptively refined quadtree (top left) with acute-angled triangles (bottom left). A mesh conforming to the domain boundary is computed using an algorithm described in [37, 43]. This mesh serves as the universal mesh for the mid-rib (crack); a mesh conforming to the mid-rib is recovered using Algorithm 4. Triangles around the mid-rib in the final mesh are highlighted.



(b) Comparison of qualities and angles in triangles before and after application of Algorithm 4. The minimum quality of a triangle in both meshes is 0.7, while the extreme angles are 26.5° and 109.3°.

Figure 9. An example demonstrating the triangulation of a cracked domain using Algorithm 4.

In all numerical examples in Section 5, the results indicate that the curvature of the discrete cracks remains bounded uniformly in h everywhere along the crack, except of course where kinks are expected to appear. A few of the examples display a kink as soon as the crack begins to propagate, and the radius of curvature of the splines that approximate the kink are roughly proportional to Δa. Hence, Equation (24) is satisfied for the numerical examples shown later. This observation is not a foregone conclusion, because the crack paths themselves are obtained as parts of the computation.

Periodically changing the universal mesh. If the mesh size of the universal mesh is small enough, it is in principle possible to use the same universal mesh for simulating the entire evolution of a crack. In practice, we often need to reduce the size of the mesh near the crack to accurately compute the

DOI: 10.1002/nme

stress intensity factors. Therefore, the same universal mesh may be used for only a few loading steps. It is then replaced by a new one that is better adapted to subsequent manifestations of the crack path.

5. NUMERICAL EXAMPLES

We examine the performance of the proposed framework to simulate evolving cracks with four examples. The first example is that of a circular arc crack in an infinite medium subjected to far-field stresses. Taking advantage of Muskhelishvili's analytical solution in [48], we construct an exact solution for problem **P** presented in Section 2.2 and use it to check the accuracy of the computed crack paths.

Then, we discuss two examples using geometries that have been considered before in the literature. The example in Section 5.2 is one of a precracked perforated plate subjected to a three-point bending test. The resulting crack paths are compared with experimental data from [1, 49]. The following example in Section 5.3 considers a clamped beam subjected to a shear load. Its solution is compared with that in [50]. The final example in Section 5.4 considers the propagation of a crack in a medium with a stiff inclusion where, for demonstration purposes, we periodically adapt the universal mesh around the evolving crack.

In each of the first three examples, we investigate the convergence of the computed crack paths in two norms we define later. We observe that the computed paths depend very weakly on the mesh and that they indeed converge as $\Delta a, h \searrow 0$.

The preexisting crack in each example is nearly orthogonal to the boundary of the domain and is sampled by 200 equidistant control points. Hence, $p = 200$ in the notation introduced in Section 3.1. Special care is needed while perturbing nodes on the boundary of the domain that are in the vicinity of the (nonpropagating) crack tip a_{-p}. In particular, the relaxation direction for such nodes in the meshing algorithm are chosen such that these nodes remain on the boundary.

Finally, as a convention, in showing the results of the numerical examples, we do not distinguish between exact and approximate quantities, unless necessary. For example, we use $K_{II}(\ell)$ to indicate the piecewise affine interpolant of $K_{II}^h\left[\Gamma_a^{h,i}\right]$ for $i = 0, 1, \ldots$.

5.1. Circular arc crack

The displacement and stress fields of an infinite medium that contains a crack shaped as a circular arc subjected to far-field stresses and traction-free faces, as illustrated in Figure 10(a), was computed in [48]. We provide a computer program to evaluate these fields [51]. The corresponding stress intensity factors can be found in [40]. We use this solution to construct a loading history that is applied as Dirichlet boundary conditions over the boundary of a square-shaped domain, as illustrated in Figure 10(b). The loading history ensures that the crack follows a circular arc.

Figure 10(b) shows a square-shaped domain Ω with a preexisting crack of radius $R = 2$ and angular span $\vartheta_0 = \pi/8$. Hence, the initial crack length is $\ell_0 = R\vartheta_0$. Let $u_{\text{exact}}(\vartheta, \sigma_x, \sigma_y)$ denote Muskhelishvili's exact solution for the displacement field for the problem illustrated in Figure 10(a) with $\tau_{xy} = 0$. The arguments ϑ, σ_x and σ_y of u_{exact} denote its dependence on the angular span of the crack and the applied far-field stresses, respectively. Dirichlet boundary conditions are applied over the entire boundary ($\Gamma_g = \partial\Omega$) with the load given by the following:

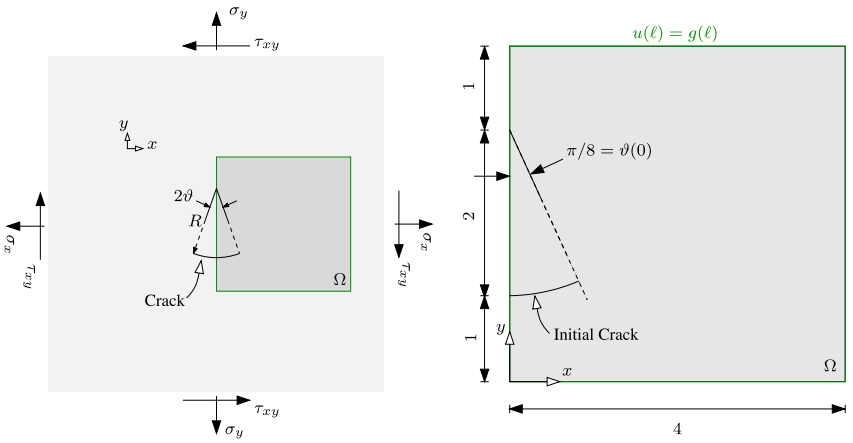
$$\overline{g}(\ell) = u_{\text{exact}}(\ell/R, \sigma_x(\ell/R), \sigma_y(\ell/R)) \quad \text{on } \partial\Omega. \tag{25}$$

In Equation (25), the span of the crack is specified via its length by $\vartheta = \ell/R$, while the stresses $\sigma_x(\vartheta), \sigma_y(\vartheta)$ are given by the following:

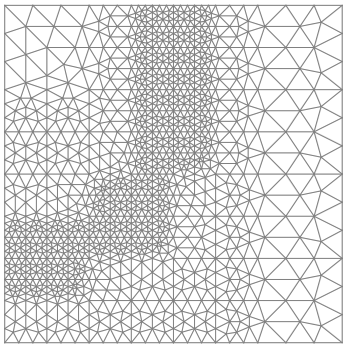
$$\sigma_x(\vartheta) = \frac{(15 + 20\cos(\vartheta) - 3\cos(2\vartheta))\sec(\vartheta/2)}{16\sqrt{\pi R \sin\vartheta}} \tag{26a}$$

$$\sigma_y(\vartheta) = \sigma_x(\vartheta)\left[\frac{16}{3(\cos(2\vartheta) - 5) - 20\cos(\vartheta)} + 1\right]. \tag{26b}$$

DOI: 10.1002/nme



(a) Circular arc crack problem (b) Modeled domain



(c) Universal mesh

Figure 10. Domain and boundary conditions.

The stresses $\sigma_x(\vartheta)$ and $\sigma_y(\vartheta)$ in Equation (26) yield a loading $\overline{g}(\ell)$ in Equation (25) that ensures $K_I(\ell) = 1$ and $K_{II}(\ell) = 0$ for any $\ell \in [0, \pi R)$. Hence, a crack of initial length ℓ_0 that propagates along a circular arc under $\overline{g}(\ell)$ in Equation (26) satisfies (5) and (7), namely, $K_I(\ell) = 1$ and $K_{II}(\ell) = 0$ for all $\ell \in (\ell_0, \pi R)$. So together with $C(\ell) = 1$ for all ℓ, they form a solution to problem **P**.

5.1.1. Simulation setup. The analyses were carried out with four progressively refined universal meshes. The coarsest one is shown in Figure 10, and its refinements were constructed by recursively subdividing each triangle into four self-similar ones. Triangles in the vicinity of the (predicted) crack path had approximately equal size h_{crack}, which was set to be smaller than the mesh size $h = \max_{K \in \mathcal{T}_h} \text{diam}(K)$. The ratio $\Delta a / h_{\text{crack}} \approx 2$ was kept constant over all simulations. We observed oscillations in the crack path when $\Delta a / h_{\text{crack}} < 1$, so as heuristics, we chose to utilize larger ratios.

For the mesh shown in Figure 10, $h_{\text{crack}}/R \approx 0.0425$. The final solution shown in Figure 11 is the result of 15 crack-tip increments. The number of increments doubles when the mesh size is halved. For completeness, we mention that the parameters used in the meshing algorithm were $N_{\text{relax}} = 8, N_{\text{steps}} = 4, N_{\text{iter}} = 4$. The interaction integral was computed with radius $\mathfrak{R}/R = 1/4$ (Section 6). We assume a state of plane strain and a Poisson ratio of $\nu = 0.3$, although any value $-1 < \nu < 0.5$ yields the same crack path and loading history.

5.1.2. Convergence of crack paths. Figure 11 shows snapshots of the evolution of the crack and the meshes conforming to it computed using the universal mesh in Figure 10. It also shows the crack paths computed with progressively refined universal meshes. It is evident that they are converging

DOI: 10.1002/nme

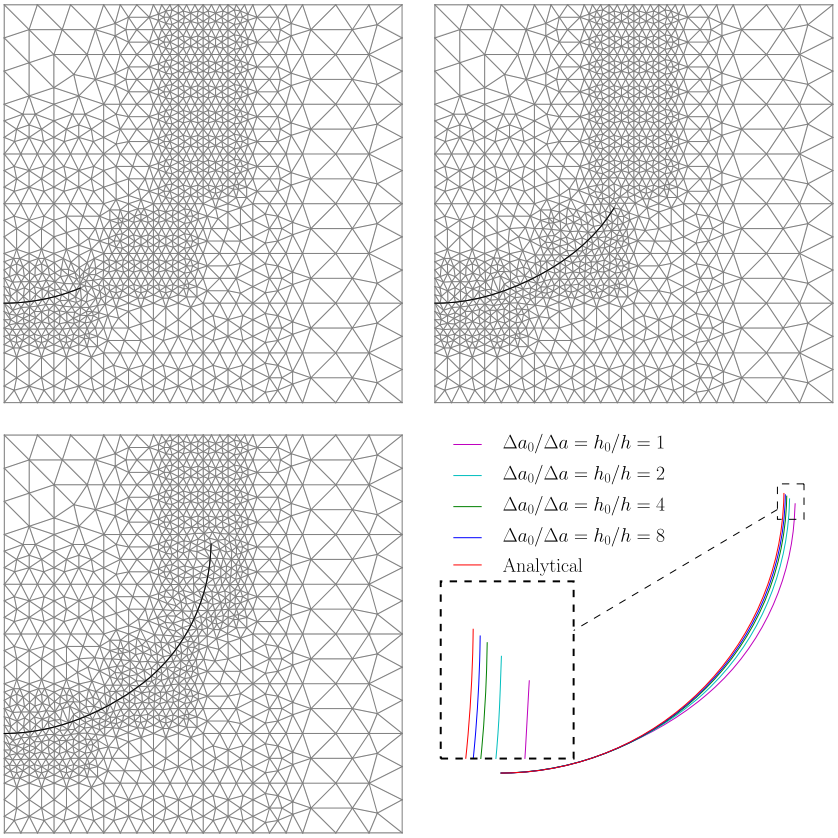


Figure 11. Snapshots of the evolution of the crack, and the mesh conforming to it, for the analysis with the coarsest universal mesh. Also shown are the computed crack paths for each level of refinement and the exact or analytical crack path.

to the exact one as $h \searrow 0$ (and hence $\Delta a \searrow 0$). In the following, we quantitatively examine the convergence.

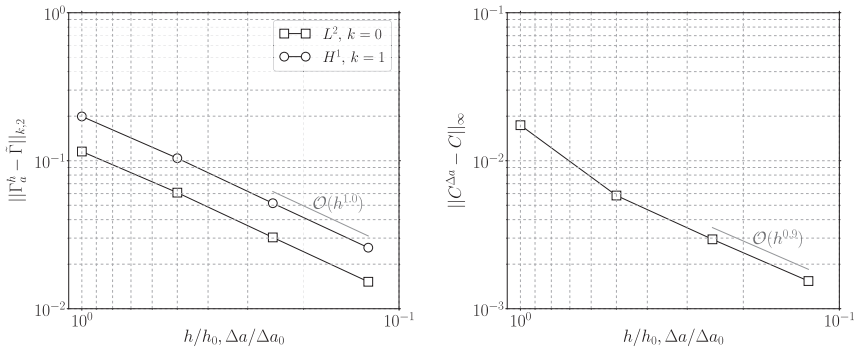
By maintaining $\Delta a / h_{\text{crack}}$ to be a constant, and by constructing progressively refined universal meshes by subdivision, cracks computed at the end of the loading program have identical final length $L = s_n - s_{-p}$. Then, let $\tilde{\Gamma}_a(s)\colon [s_{-p}, s_n] \to \mathbb{R}^2$ be an arc-length parametrization of the exact solution $\Gamma_a(L)$, a circular arc crack of length L, so that $\tilde{\Gamma}_a(s_{-p}) = \Gamma_a(L) \cap \partial\Omega$. We compare each computed discrete crack Γ_a^h with the exact one using one of the following Sobolev norms:

$$\left\| \Gamma_a^h - \tilde{\Gamma}_a \right\|_{k,2} = \sum_{i=0}^{k} \left(\int_{s_{-p}}^{s_n} \left| \frac{d^i \left(\Gamma_a^h - \tilde{\Gamma}_a \right)}{ds^i} \right|^2 ds \right)^{\frac{1}{2}}, \tag{27a}$$

$$\left\| \Gamma_a^h - \tilde{\Gamma}_a \right\|_{k,\infty} = \sum_{i=0}^{k} \max_{s \in [s_{-p}, s_n]} \left| \frac{d^i \left(\Gamma_a^h - \tilde{\Gamma}_a \right)}{ds^i} \right|. \tag{27b}$$

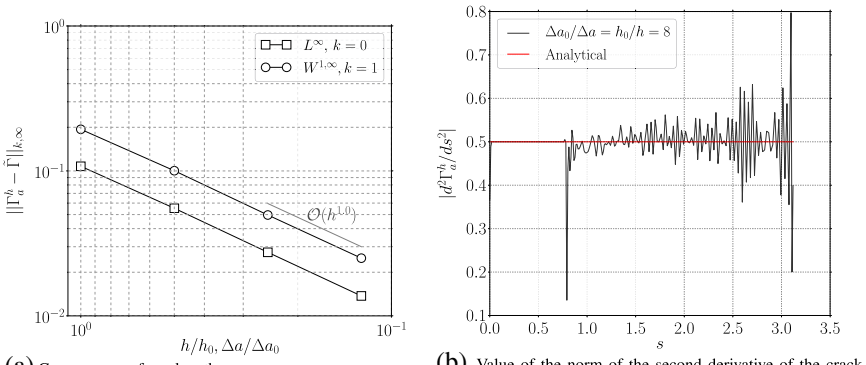
Figure 12(a) shows the L^2 and H^1 norms of the error, as defined in Equation (27a) with $k = 0$ and $k = 1$, respectively. The convergence curves in the ∞ norms are shown in Figure 13(a), as defined in Equation (27b) with $k = 0$ and $k = 1$, respectively. The errors in the path and its first derivative converge as $\mathcal{O}(h)$, in both families of norms. The analysis of problem **P** and its approximation has not yet been performed, so it is not possible to say whether these convergence rates could be expected in general situations. Figure 13(b) shows the norm of the second derivative as a function of s for $\Delta a = \Delta a_0/8$. It is worthwhile noting that although the norm of the second derivative deviates more pronouncedly from the analytical value near the two crack tips and the initial crack location, it

DOI: 10.1002/nme



(a) Convergence of crack path (b) Convergence of loading history

Figure 12. Convergence curves for a circular arc crack.



(a) Convergence of crack path

(b) Value of the norm of the second derivative of the crack path, $|d^2\Gamma_a^h/ds^2|$, for the finest value of Δa. The exact value is $1/R = 0.5$ in this case.

Figure 13. Convergence curves for a circular arc crack in the ∞ norm and value of the norm of the second derivative of the crack. The norm of the second derivative is oscillatory, but it approaches the exact value for most values of s, except near $s = 0, \ell_0, L$.

remains bounded elsewhere. Similar plots for other values of Δa (not shown here) suggest that this bound is uniform as $\Delta a \searrow 0$. This is important for the efficiency of the discrete evolution algorithm §4.4.

To evaluate the convergence of C^n to $C(\ell_0 + n\Delta a)$ as $\Delta a \to 0$, we define $C^{\Delta a}(\ell)$ to be the piecewise affine interpolant of the points $((\ell_0, C^0), \ldots, (L, C^{L/\Delta a}))$ in the interval $[\ell_0, L]$ and consider the $L^\infty[\ell_0, L]$ error:

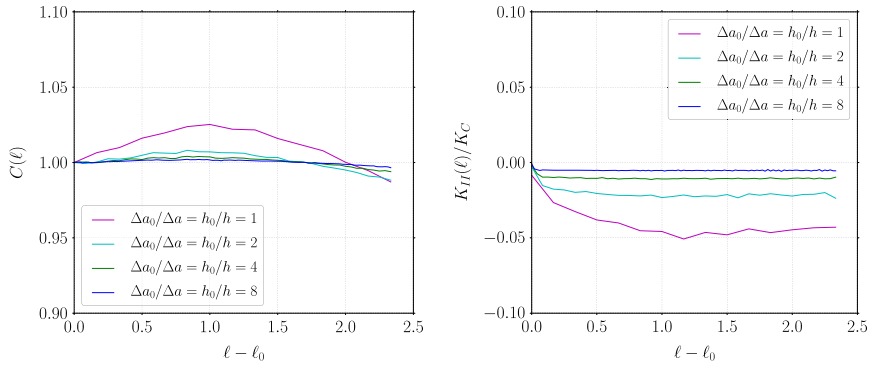
$$\|C^{\Delta a} - C\|_\infty = \max_{\ell_0 \leqslant \ell \leqslant L} |C^{\Delta a}(\ell) - C(\ell)|. \tag{28}$$

The value of the aforementioned error as a function of Δa is shown in Figure 12(b). It converges approximately as $O(h^{0.9})$. The actual values of $C^{\Delta a}(\ell)$ are shown in Figure 14(a), and they are all near the exact constant value of 1. Similarly, the computed values of $K_{II}(\ell)$ are shown in Figure 14(b), and they lie around the exact value 0. This shows that the asymptotic enforcement through Equation (12) has the expected effect.

5.2. Perforated plate

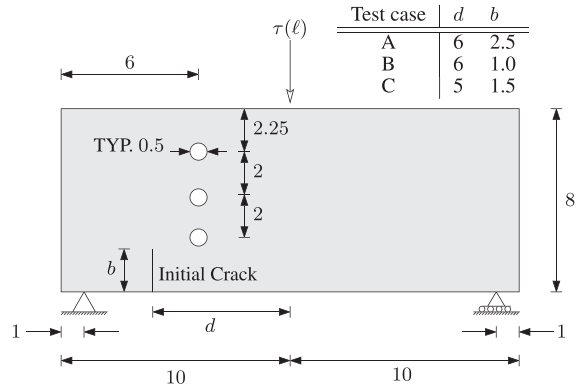
In the next example, we compute the evolution of an initially straight crack in a thin perforated plate during a three-point bending test, depicted in Figure 15(a). Experimental data on the resulting fractures for three combinations of initial crack length and distance of the crack from the perforations are available from [49]. The experiments were performed on PMMA plates. Correspondingly, we assume a Poisson ratio of $\nu = 0.35$ and a state of plane stress in our numerical simulations. Displacements are constrained at the two supports, and a point-load $\overline{\tau}$ is applied at the midpoint of the

DOI: 10.1002/nme

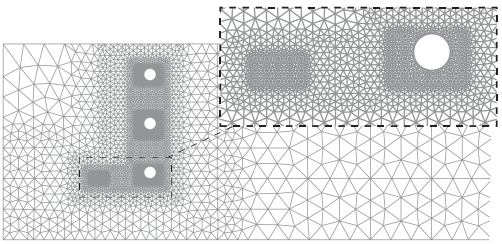


(a) Evolution of C (b) Evolution of K_{II}

Figure 14. Evolution of the values computed for $C(\ell)$ and $K_{II}(\ell)$. By design, they should approximate the exact values $C = 1$ and $K_{II} = 0$ for all ℓ.



(a) Modeled domain



(b) Universal mesh

Figure 15. Geometry for a plate with holes and the universal mesh adopted.

top face of the specimen. The rest of the boundary and the crack faces are traction free. The value of $|\overline{\tau}|$ is arbitrarily set to 1; its choice alters the expected value of $C(\ell)$ but not the crack path.

5.2.1. Simulation setup. The geometric parameters for three test cases A, B, and C are listed in the table adjoining Figure 15(a). For each case, we constructed universal meshes for the domain that are adequately refined near the holes to represent their geometry well and along the expected crack path. Figure 15(b) shows the coarsest universal mesh used for case A. A kink appears at the initial crack tip once the crack begins to propagate. This is manifested as a point where the discrete crack has a large curvature in its representation as a cubic spline. This necessitates additional refinement of the universal mesh near the initial tip. We are exploring the possibility of explicitly representing the kink by accommodating piecewise C^2 curves in the meshing algorithm. Similarly, a robust algorithm to

DOI: 10.1002/nme

deform a universal mesh to conform to triple junctions is essential for accommodating bifurcations and intersections of crack paths. Both questions remain outstanding at this time.

For each of the three cases, simulations were performed with four levels of refinement for their respective universal meshes. Similar to the previous example, meshes were refined through self-similar subdivision. The ratio $\Delta a / h_{\text{crack}} \approx 2$ was maintained as a constant throughout, where h_{crack} denotes the size (diameter) of the largest triangle intersected by the crack. The coarsest meshes have $h_{\text{crack}} \approx 0.4$. Parameters N_{relax}, N_{steps} and N_{iter} in the meshing algorithm were all set to 4.

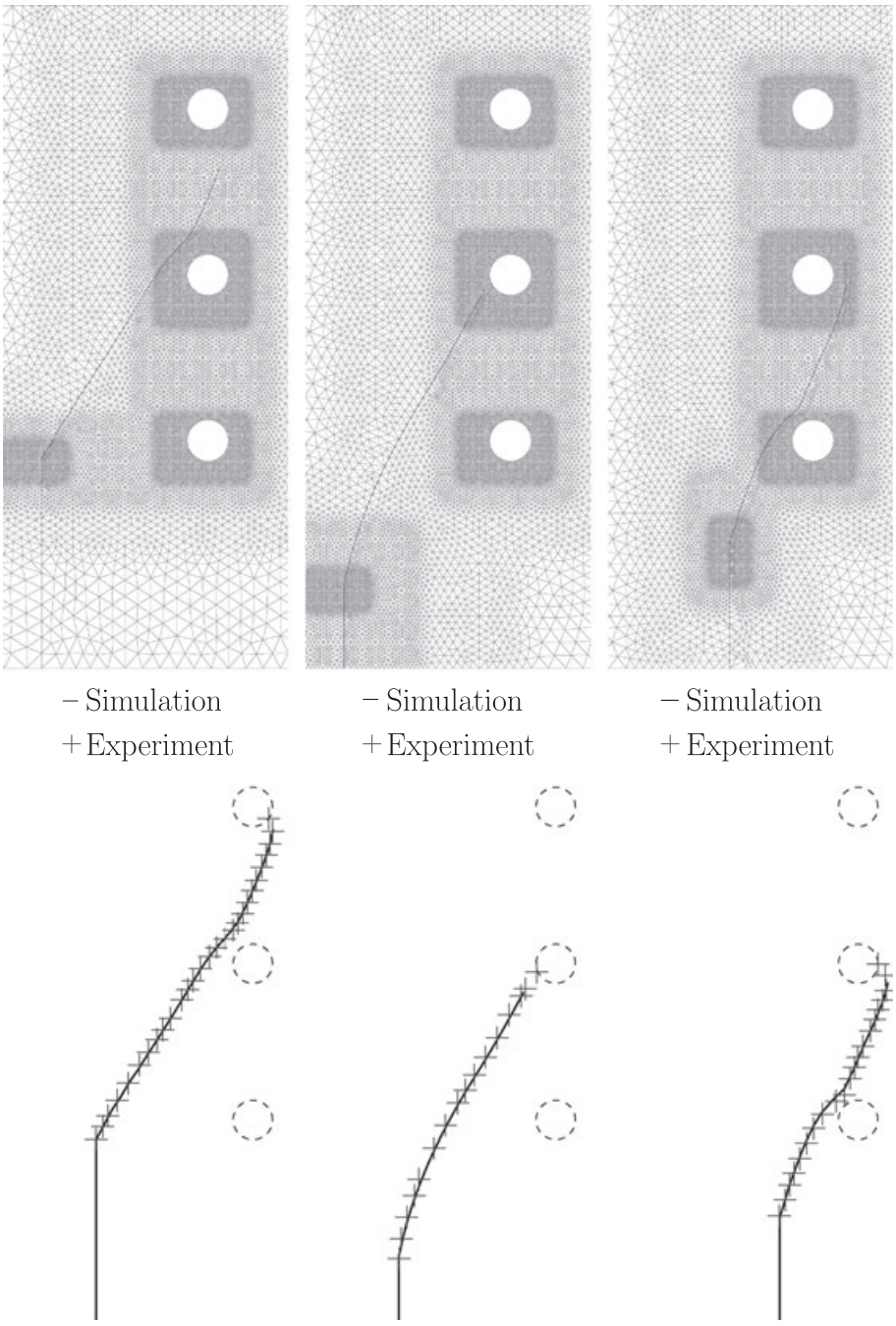
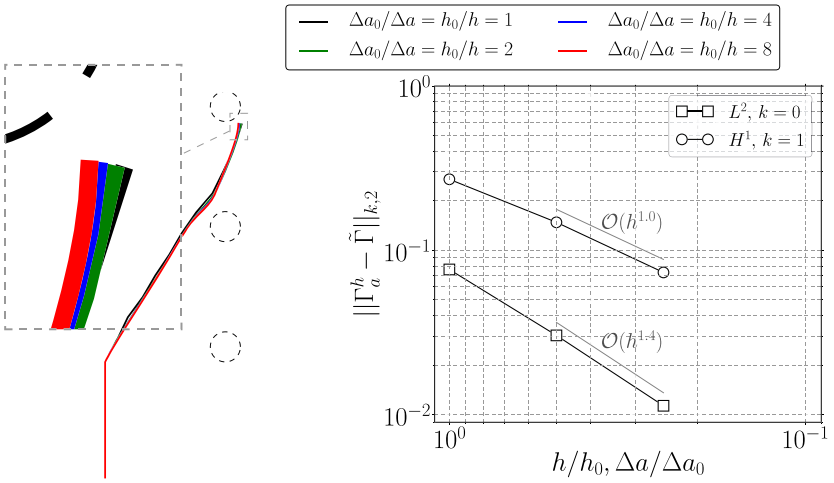
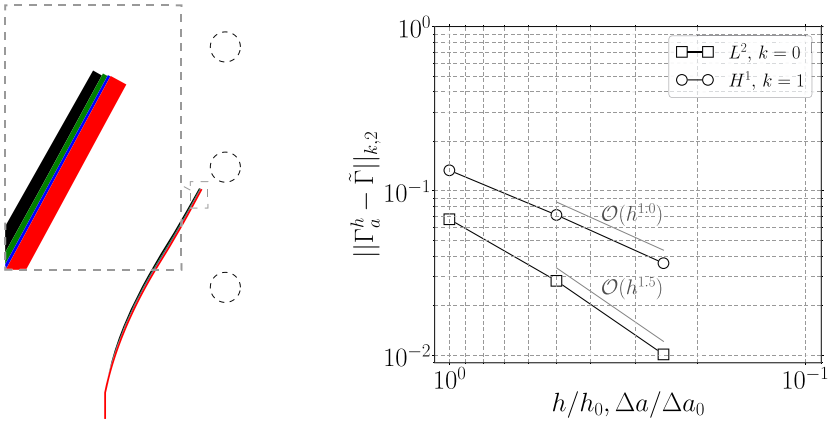


Figure 16. (Top row) Snapshots of the mesh for (left) case A, (middle) case B, and (right) case C. The bottom row shows the final crack path. Experimental results were digitized from [1, 49].

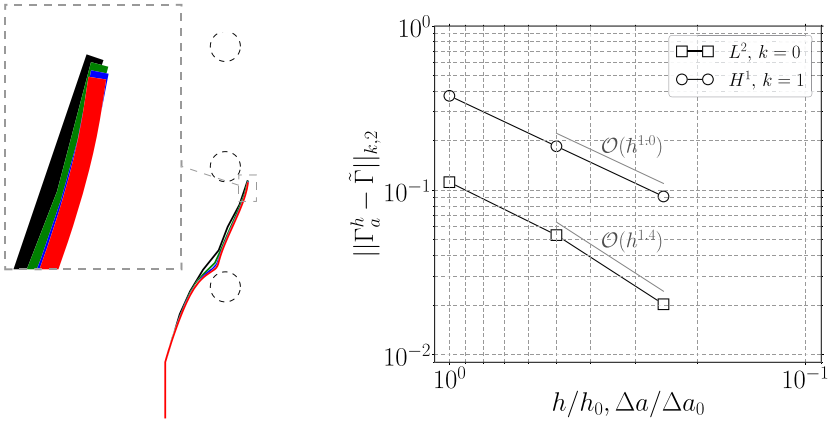
DOI: 10.1002/nme



(a) Case A paths (b) Case A convergence



(c) Case B paths (d) Case B convergence



(e) Case C paths (f) Case C convergence

Figure 17. Convergence of discrete paths. The errors were computed by adopting the discrete crack that resulted from the finest mesh (labeled as $\tilde{\Gamma}$) as a proxy for the exact solution.

DOI: 10.1002/nme

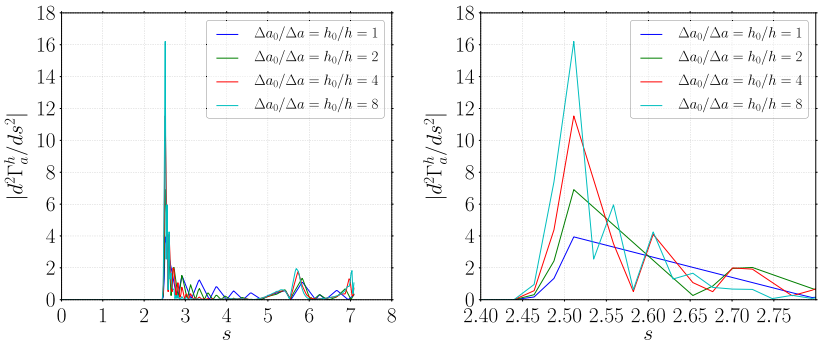
Simulations were terminated when the fracture approached any of the holes within a distance $4h_{\text{crack}}(= N_{\text{relax}}h_{\text{crack}})$, for h_{crack} in each one of the coarsest meshes, to avoid crack-boundary intersections. This choice resulted in about 10–15 loading steps for coarsest meshes of the different cases. The number of loading steps doubled with each refinement of the universal mesh. Hence, for a chosen case A, B, or C, the length of the curves S_a^h for the final discrete cracks computed with each of the four universal meshes were identical. This feature facilitates their comparison, which we undertake subsequently.

The radius used in computing the interaction integrals at the nth load increment was chosen as $\mathfrak{R} = \min\{0.75, d(\mathrm{a}_n, \partial\Omega)\}$. When the crack is sufficiently far away from the perforations, we have that $\mathfrak{R} = 0.75$. As the propagating crack tip approaches the holes, it is necessary to reduce the size of the region over which the interaction integrals are evaluated.

5.2.2. Convergence of crack paths and comparison with experiments. Figure 16 shows representative snapshots of the crack path for each of the three cases. The paths computed with the coarsest meshes possess visible oscillations, indicating the need for sufficient mesh refinement. The snapshots in Figure 16 are results computed by refining the coarsest meshes once. The figure also compares the computed paths with the experimental data obtained by digitizing the results in [1, 49]. The two agree well.

Figures 17(a), (c), and (e) plot the crack paths computed with each of the four universal meshes, revealing an apparent convergent behavior. Analytical solutions for crack paths are not known for any of the three cases, which precludes examining the convergence of computed paths. Instead, Figure 17(b), (d), and (f) compares the paths computed using three universal meshes with the fourth one computed using the most refined mesh, in the L^2 and H^1 norms. These plots do not imply convergence of computed paths to a limiting one. However, they do suggest that if computed paths converge, we can expect $\mathcal{O}(h)$ convergence of the L^2 and H^1 norms. We also observed that beyond the initial kink, the second derivative of each computed crack remains bounded irrespective of the universal mesh used; Figure 18 depicts an example corresponding to case A. This is an important feature of the solution; because for a fixed (and sufficiently large) ratio of Δa and h_{crack}, success of the meshing algorithm with a fixed universal mesh relies on the curvature of the crack remaining bounded.

Figure 19 shows the evolution of $K_{II}(\ell)/K_c$ and $C(\ell)$ as functions of $\ell - \ell_0$. By definition of problem **P**, $K_{II}(\ell) = 0$ for the exact path, except at the commencement of propagation. We observe that $K_{II}(\ell)$ of the computed paths indeed approach zero beyond the initial loading step as the universal mesh is refined, in all three test cases. Because $K_{II}(\ell_0^-) \neq 0$, an initial kink appears at $\ell = \ell_0$. The profile of $K_{II}(\ell)$ shows small deviations from zero when the crack turns sharply around the perforations and also that they decrease with the mesh size.

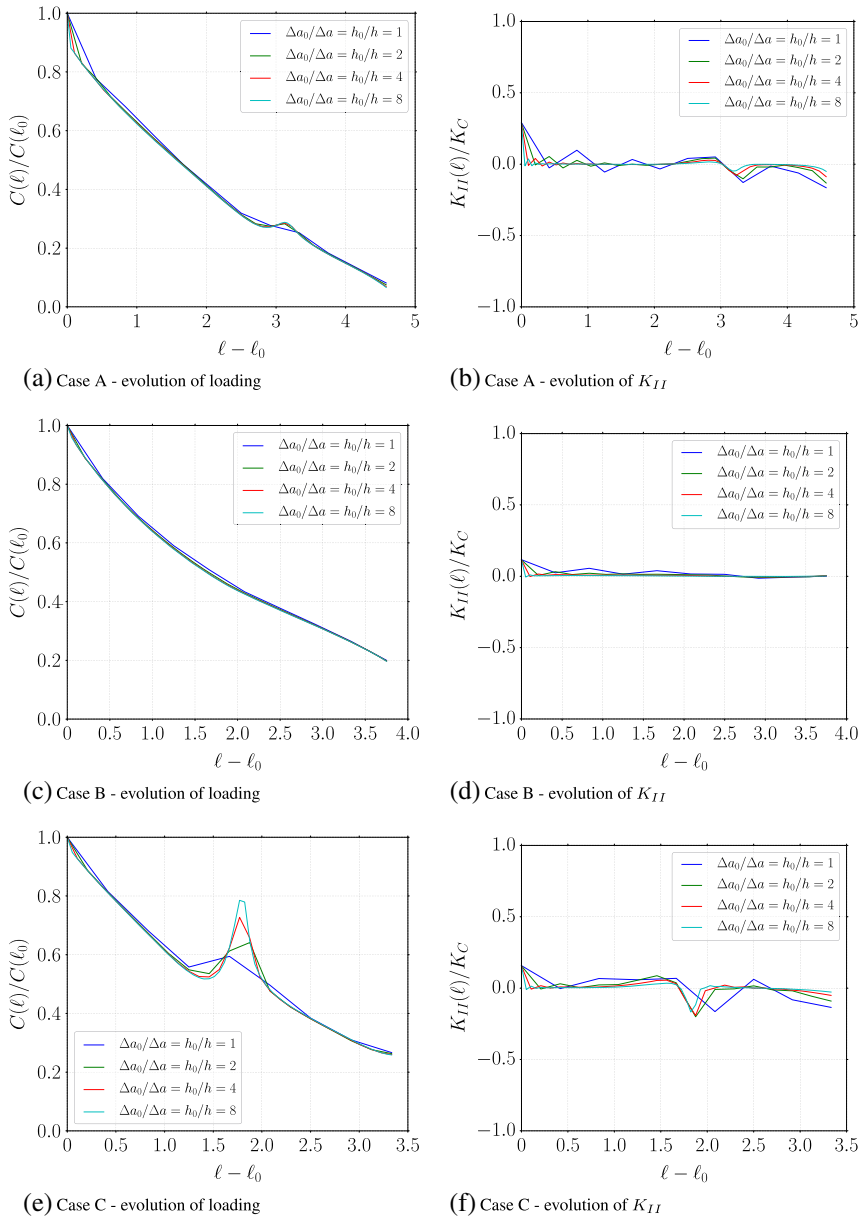


(a) Values of the second derivatives along the entire length of the crack

(b) Values of the second derivatives around $s = s_0$

Figure 18. Value of the norm of the second derivative of the discrete crack computed with each one of the four refinements, for case A, as a function of the parametric coordinate s. On the right-hand-side, we show the close up at $\ell = \ell_0$.

DOI: 10.1002/nme



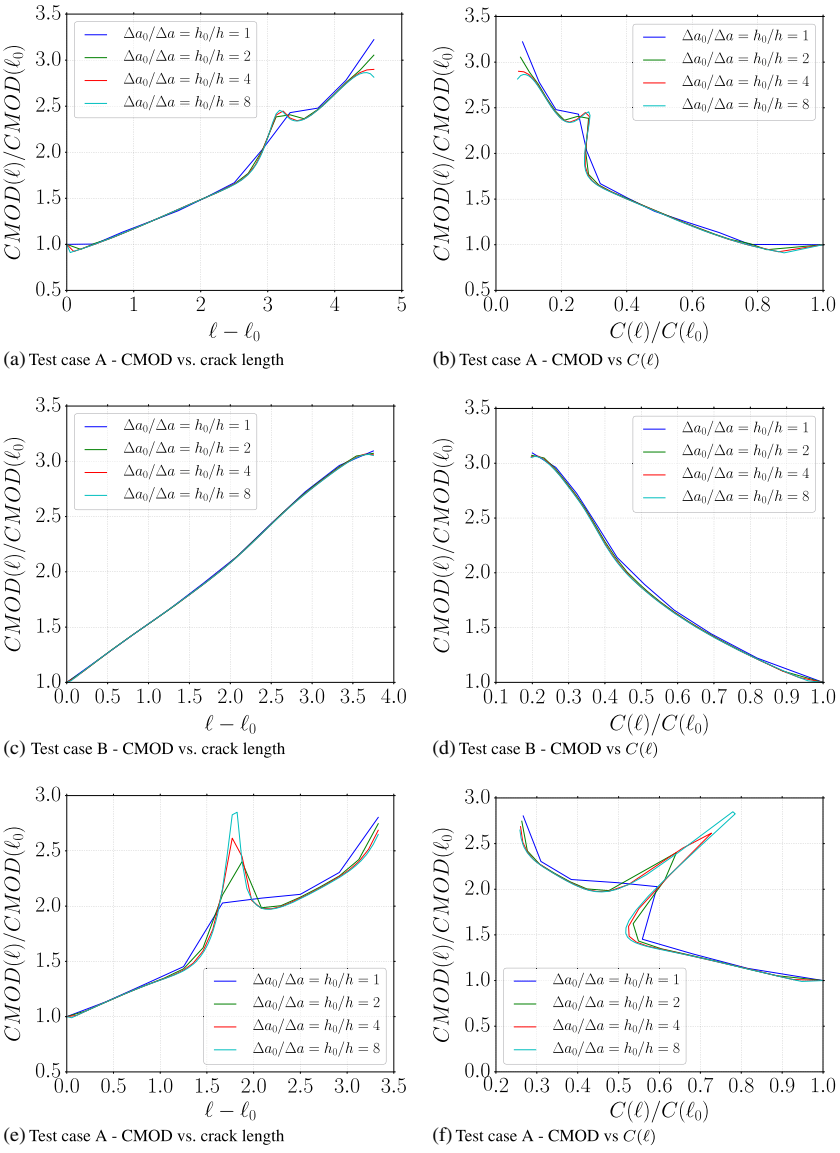
(a) Case A - evolution of loading
(b) Case A - evolution of K_{II}
(c) Case B - evolution of loading
(d) Case B - evolution of K_{II}
(e) Case C - evolution of loading
(f) Case C - evolution of K_{II}

Figure 19. Evolution of $K_{II}(\ell)$ and $C(\ell)$. A clear convergence trend is observed. Some additional refinement is needed near some of the holes, and a discontinuity is observed in both K_{II} and C as the crack begins to grow, because of the appearance of an initial kink.

The loading history $C(\ell)$ displays an overall convergent behavior as well. The results suggest that further mesh refinement is needed near some of the perforations. Notice that the initial kink induces a discontinuity in the value of $C(\ell)$ at $\ell = \ell_0$.

The experiments in [49] were performed by controlling the crack mouth opening displacement (CMOD). Consequently, the experimental paths possess a monotonically nondecreasing CMOD. In contrast, our numerical simulations were performed by controlling the crack length. The resulting CMOD values are plotted in Figure 20. They show that the CMOD does not increase monotonically with crack advancement in test cases A and C. In particular, we observe a closing of the crack mouth when the crack turns sharply near the perforations. This suggests short periods of unstable crack growth in the CMOD-controlled experiments when the crack approaches the perforations. A

DOI: 10.1002/nme



(a) Test case A - CMOD vs. crack length

(b) Test case A - CMOD vs $C(\ell)$

(c) Test case B - CMOD vs. crack length

(d) Test case B - CMOD vs $C(\ell)$

(e) Test case A - CMOD vs. crack length

(f) Test case A - CMOD vs $C(\ell)$

Figure 20. Evolution of crack mouth opening displacement (CMOD).

monotonic CMOD in the experiments corresponds to a discontinuous loading history. On the other hand, our crack length-controlled numerical simulations result in kink-free paths with nonmonotonic CMOD but continuous loading histories.

5.3. Sheared beam

Figure 21 shows the geometry of a cantilever beam subjected to a shear end load $|\overline{\tau}| = 1$. For comparison with results from [50], we assume a plane strain condition and a Poisson ratio of $\nu = 0.25$.

Similar to the previous examples, we constructed a universal mesh for the cracked domain that is shown in Figure 21 and three refinements of it by subdivision. The coarsest mesh had $h_{\text{crack}} \approx 0.5$, and the ratio $\Delta a / h_{\text{crack}} \approx 2$ was maintained as a constraint for simulations. Parameters in the meshing algorithm were chosen to be $N_{\text{relax}} = 8$, $N_{\text{steps}} = 4$, and $N_{\text{iter}} = 4$. The radius for computing the interaction integral was set to $\Re = \min\{0.75, d(\mathrm{a}_n, \partial\Omega)\}$. The simulation with the coarsest universal mesh includes 20 loading steps.

DOI: 10.1002/nme

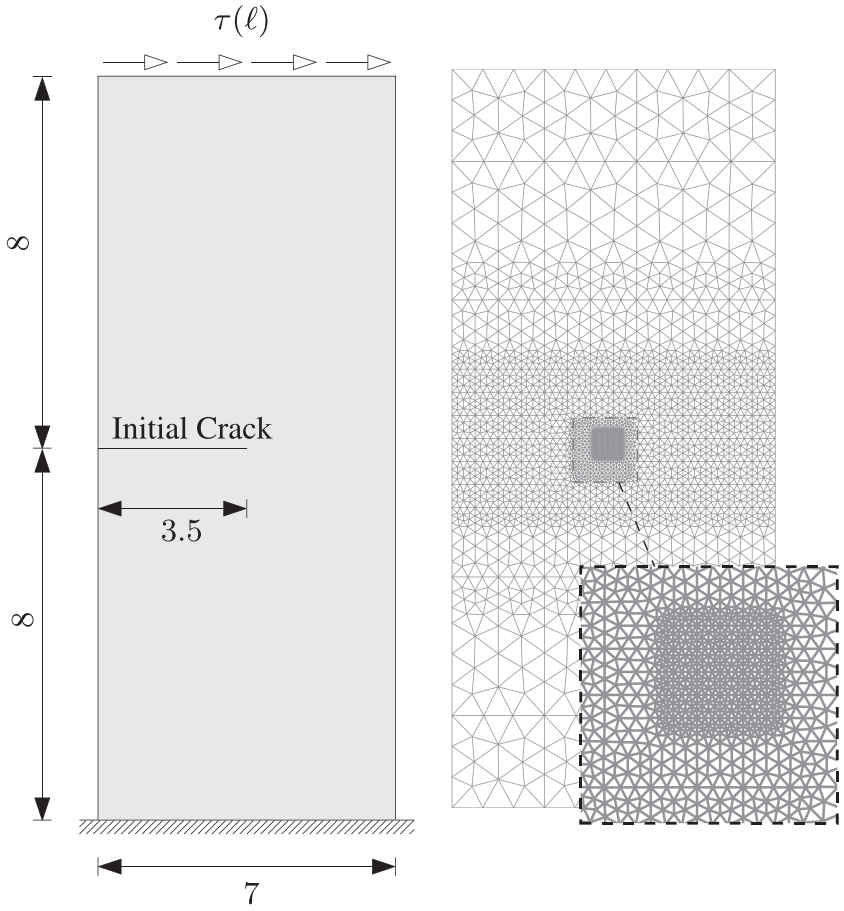


Figure 21. Geometry for a (left) clamped beam with a shear loading and (right) the corresponding universal mesh.

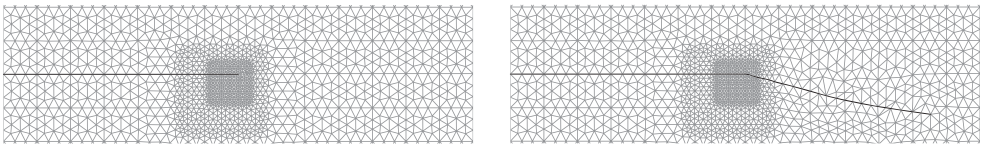


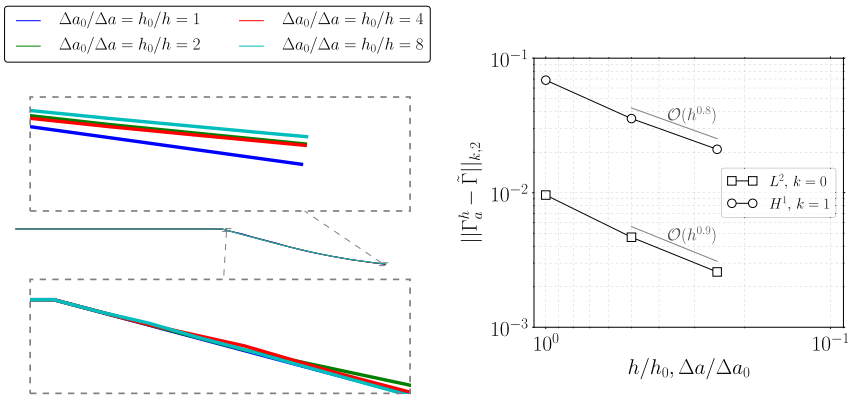
Figure 22. Evolution of the crack and conformed universal mesh.

Figure 22 shows snapshots of the crack path and the conforming mesh around it at two different instants. The paths computed with the four meshes are shown in Figure 23(a) and are qualitatively similar to those in [50]. We observe that the paths seem to converge with refinement of the universal mesh. Figure 23(b) examines the distance between the path computed with the most refined mesh and the paths computed with the coarser meshes. Their differences decreases as $\mathcal{O}(h^{0.9})$ and $\mathcal{O}(h^{0.8})$ in the L^2 and H^1 norms, respectively. Finally, Figure 24 shows that the loading history $C(\ell)$ appears to converge and that $K_{II}(\ell)$ approaches zero, except at the initial kink.

5.4. Stiff inclusion

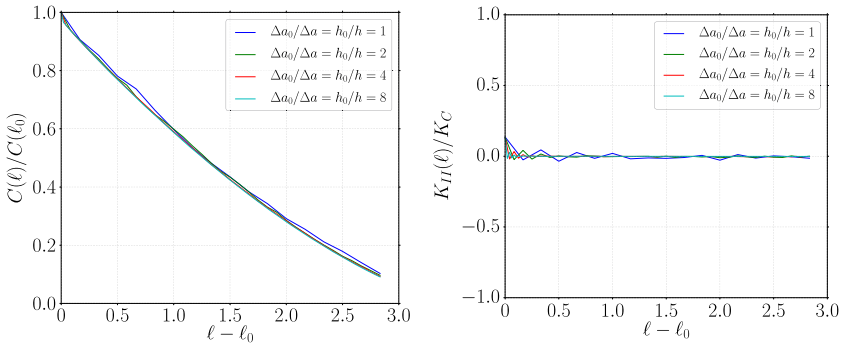
In this final example, we compute the evolution of an initially straight crack in a domain containing a stiff circular inclusion. The geometry of the domain and the inclusion are depicted in Figure 25, along with the prescribed Dirichlet boundary conditions and shear load. The material of the inclusion is assigned an identical Poisson ratio as the matrix ($\nu = 0.3$) but has a Young's modulus equal to

DOI: 10.1002/nme



(a) Crack paths (b) Convergence of error norms

Figure 23. Convergence of the crack path.



(a) Evolution of loading (b) Evolution of K_{II}

Figure 24. Evolution of $K_{II}(\ell)$ and $C(\ell)$. The presence of a kink at $\ell - \ell_0$ is evident from the nonzero value of $K_{II}(\ell_0)$. This induces a discontinuity in $C(\ell)$ at $\ell = \ell_0$.

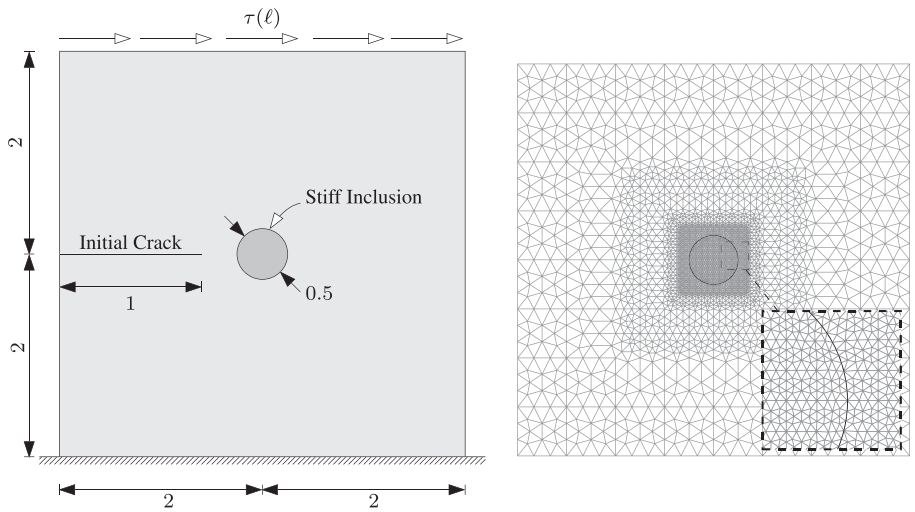


Figure 25. (Left) Geometry for a clamped beam with a shear loading and (right) a universal mesh constructed for the interface before inserting the initial crack, following the refinement strategy outlined in the text.

DOI: 10.1002/nme

1000 times that of the matrix. We assume a state of plane strain and that the displacement field is continuous across the interface (i.e., a perfectly bonded interface).

5.4.1. Adaptive refinement of the universal mesh. We utilize this example to demonstrate the possibility of periodically adapting the universal mesh during the evolution of the crack. As mentioned previously, such refinement serves dual purposes. First, as the crack develops small features and/or large curvatures, it is necessary to adapt the universal mesh if it does not satisfy the mesh size requirements in the meshing algorithm. Local estimates of the required mesh size can be found in

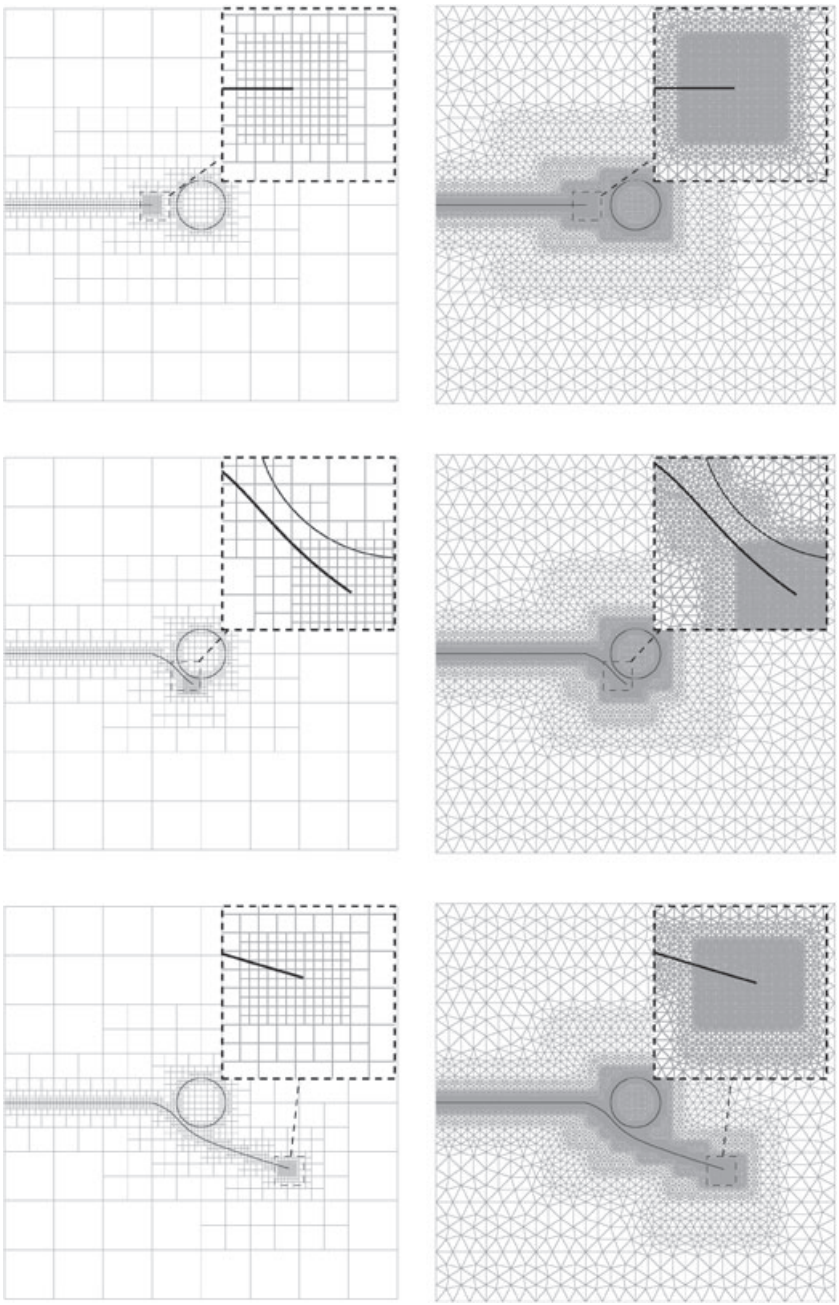


Figure 26. Evolution of the crack, the mesh, and the quadtree for several time steps of the sheared block with a stiff inclusion.

DOI: 10.1002/nme

[42]. Second, refinement in the vicinity of the propagating tip yields better approximations of the stress intensity factors and presumably a more accurate evolution of the crack.

We construct adaptively refined universal meshes from adaptively refined quadtrees covering the domain, which are then tiled with stencils of acute-angled triangles [47]. In fact, the universal meshes in all the previous examples were constructed in this way. Consequently, the resulting meshes satisfy the acute conditioning angle requirement in the meshing algorithm, irrespective of the location of the crack. Here, we additionally adapt the quadtree as the crack evolves.

In principle, a strategy to adapt the universal mesh should employ mesh size estimates required in the meshing algorithm and *a posteriori* error estimates from computed solutions to guide refinement. Instead, for simplicity, we choose an *ad hoc* strategy defined by the following rules:

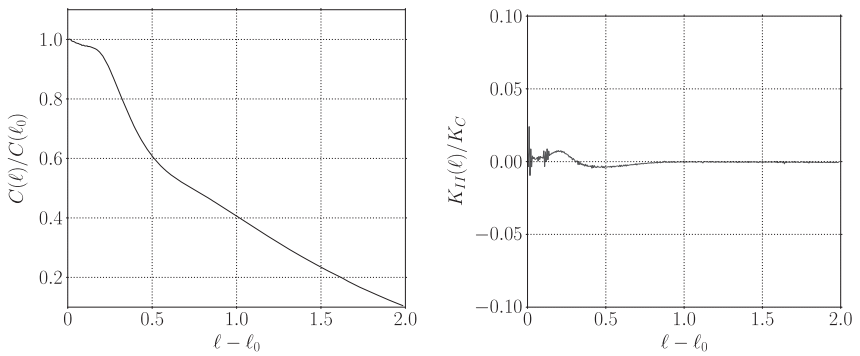
(i) Squares in the quadtree (also referred to as its nodes) away from the crack and the matrix-inclusion interface are assigned a size $\delta := 1/2$.
(ii) Squares in the vicinity of the crack and those near the matrix-inclusion interface have size $\delta/16$.
(iii) Squares contained by the ball of radius $\mathfrak{R}$ around the crack tip are prescribed an even smaller size of $\delta/32$. These squares include the region over which interaction integrals are computed.

As the crack evolves, squares in the quadtree are subdivided on the basis of the aforementioned criteria; sample snapshots of the quadtrees and the resulting universal meshes are shown in Figure 26. Two features are apparent from such a refinement strategy. First, it permits adaption of the universal mesh during each step of the evolution of the crack. Second, coarsening of the mesh is limited to a small region around the propagating tip. Neither of these features are essential and may be overcome with an alternate set of refinement criteria. For instance, it may suffice to periodically update the universal mesh rather than at each update of the crack tip.

The simulation setup was similar to the previous examples. The radius for computing the interaction integrals at the nth load step was chosen as $\mathfrak{R} := \min\{0.75, d(\mathrm{a}_n, \partial\Omega \cup \Gamma_{\text{int}})\}$ where Γ_{int} is the interface between the matrix and the inclusion. The computed evolution reveals that the crack never approaches too close to the boundary $\partial\Omega$ of the domain. Hence, the choice of $\mathfrak{R}$ effectively reduces to $\min\{0.75, d(\mathrm{a}_n, \Gamma_{\text{int}})\}$, similar to the definition in the example in Section 5.2 with the perforations in that example replaced by the inclusion here.

Deforming a universal mesh to accommodate the interface Γ_{int} has not been discussed here. We only mention that the requisite perturbations near the interface are for the most part identical to those around a crack discussed in Section 4, while omitting perturbations around the crack tips. Parameters N_{relax}, N_{iter}, and N_{steps} in the meshing algorithm to accommodate the crack and the interface were set to 8, 4, and 4, respectively.

We conclude this section noting the computed evolution of $K_{II}(\ell)$ and $C(\ell)$ in Figure 27. As observed in previous examples, we see that K_{II} is close to zero, except at the initial kink.



(a) Evolution of loading history (b) Evolution of K_{II}

Figure 27. Evolution of $K_{II}(\ell)$ and $C(\ell)$.

DOI: 10.1002/nme

6. CONCLUDING REMARKS

We formulated a specific class of linear elastic fracture mechanics problems to model quasistatic and kink-free propagation of cracks in two dimensions. Its numerical approximation relies on two new ideas: robustly deforming a universal mesh to accommodate the crack propagating through and accurately computing stress intensity factors for curvilinear cracks. The performance of the resulting numerical scheme was examined through detailed examples and revealed favorable comparisons with exact solutions and experimental data.

The numerical method was described in sufficient detail to facilitate a correct and efficient implementation. We have deliberately omitted comprehensive discussions of the meshing algorithm and the calculation of stress intensity factors, referring instead to [37, 38, 41–43] for an explanation of the rationale behind them, their analysis when possible, and extensive scrutiny through examples.

We also mention that the scheme is not limited to problems with smooth cracks. The latter restriction stems essentially from the meshing algorithm. We expect to create algorithms permitting piecewise smooth and bifurcated cracks in the near future. Preliminary results using extensions of the scheme for simulating thermal cracking and hydraulic fracturing problems appear to be promising.

We end the article noting that many challenges remain in the construction of models for fracture, their analysis (cf. [25, 28, 52]) and validation through experiments, especially in three dimensions. Numerical simulations play a crucial role in their study and their application to practical engineering problems and have increasingly engaged engineers, mechanicians, mathematicians, and physicists alike. The contributions of Ted Belytschko are especially noteworthy in this regard.

APPENDIX A: INTERACTION INTEGRAL FOR A CURVED CRACK

This section summarizes a method to compute the stress intensity factors through an interaction integral specially designed for a curved crack. More details can be found in [41]. The stress intensity factors corresponding to a displacement field u are obtained from the following:

$$K_{I,II} = \frac{\mathcal{I}\left[\boldsymbol{\beta}, \boldsymbol{\beta}^{\text{aux}}_{I,II}, \delta\boldsymbol{\gamma}\right]}{\eta K^{\text{aux}}_{I,II}},$$

where $\boldsymbol{\beta} := \nabla u$, $\boldsymbol{\beta}^{\text{aux}}_{I,II}$ is an auxiliary tensor field, $\delta\boldsymbol{\gamma}$ is an auxiliary vector field,

$$\eta = \frac{2\left(1-\nu^2\right)}{E(1-2\nu)}$$

and $\mathcal{I}$ is the interaction integral. This expression is valid in plane strain. In the sequel, we first define $\boldsymbol{\beta}^{\text{aux}}_{I,II}$ and $\delta\boldsymbol{\gamma}$ and then provide the definition of $\mathcal{I}$.

The auxiliary field $\boldsymbol{\beta}^{\text{aux}}_{I,II}$ is constructed so that

(i) If $\mathbb{C}$ is the elasticity tensor and $\boldsymbol{\sigma}^{\text{aux}}_{I,II} = \mathbb{C} : \boldsymbol{\beta}^{\text{aux}}_{I,II}$, then $\boldsymbol{\sigma}^{\text{aux}}_{I,II}\boldsymbol{n} = 0$ on the crack faces.
(ii) Near the crack tip, $\boldsymbol{\beta}^{\text{aux}}_{I,II}$ behaves like $\nabla u^{I,II}$, where $u^{I,II}$ is the displacement field of the mode I, II asymptotic solution.

More precisely, let (r, ϑ) be crack-tip polar coordinates such that $\vartheta_{\min}(r) \leqslant \vartheta < \vartheta_{\min}(r) + 2\pi$ for each $r > 0$, where $\vartheta_{\min}(r)$ is the angle at which the circle of radius r intersects the curvilinear crack. Then, $\boldsymbol{\beta}^{\text{aux}}_{I,II}(r, \vartheta)$ is given by the following:

$$\boldsymbol{\beta}^{\text{aux}}_{I,II}(r,\vartheta) = \sum_{k,l=1,2} \left[\boldsymbol{g}_k(0^+) \cdot \nabla u^{I,II}(r, \varphi(r,\vartheta)) \cdot \boldsymbol{g}_l(0^+)\right] \boldsymbol{g}_k(r) \otimes \boldsymbol{g}_l(r),$$

DOI: 10.1002/nme

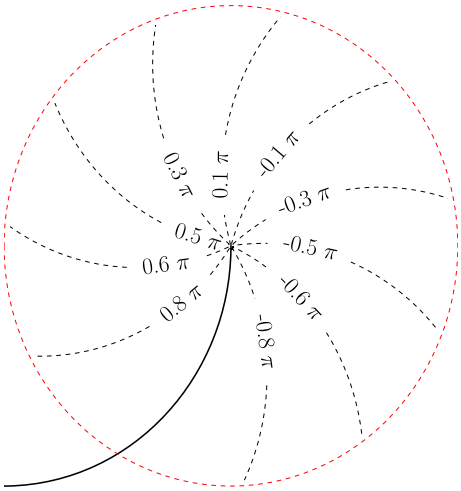


Figure A.1. Isolines of the function φ, which always lies between $-\pi$ and π.

where $(\boldsymbol{g}_1, \boldsymbol{g}_2)$ is a system of curvilinear right-handed orthonormal basis such that $\boldsymbol{g}_1(r)$ parallels the direction tangent to the crack at a distance of r away from the crack tip, pointing toward the crack tip. Here, $\varphi(r, \vartheta) = \vartheta - \vartheta_{\min}(r) - \pi$ so that the value of $\varphi(r, \vartheta)$ is always between $-\pi$ and π. In Figure A.1, values of φ are plotted for a circular arc crack geometry. For a straight crack, $\vartheta_{\min}(r) = -\pi$ for all $r > 0$, and $\varphi(r, \vartheta) = \vartheta$.

The vector field $\delta\boldsymbol{\gamma}$ only depends on r and is defined as $\delta\boldsymbol{\gamma}(r) = q(r)\boldsymbol{g}_1(r)$, where $q \in C^1([0, \infty))$ satisfies the following:

$$q(r) = \begin{cases} 1, & \text{if } r < \Re_I, \\ f(r), & \text{if } \Re_I < r < \Re, \\ 0 & \text{otherwise,} \end{cases}$$

with $f(r)$ being a cubic polynomial, $\Re$ is such that for any $r < \Re$, there exists a unique point on Γ_a with a distance of r away from the crack tip, and $\Re_I := \Re/4$.

The interaction integral for a curved crack is defined as follows:

$$\mathcal{I}[\boldsymbol{\beta}, \boldsymbol{\beta}^{\text{aux}}, \delta\boldsymbol{\gamma}] := -\int_{\boldsymbol{B}_{\Re}(\boldsymbol{x}_t)} \left[\left(\beta_{jk}\sigma^{\text{aux}}_{jk,i} - \beta^{\text{aux}}_{ji,k}\sigma_{jk} - \beta_{ji}\sigma^{\text{aux}}_{jk,k} \right) \delta\gamma_i + \overline{\Sigma}_{ij}\delta\gamma_{i,j} \right] dV$$

where indicial notation and the summation convention on repeated indices have been adopted, all components and partial derivatives are with respect to a Cartesian system of coordinates and associated basis, and

$$\overline{\Sigma}_{ij} := \frac{1}{2}\delta_{ij}\left(\sigma_{kl}\beta^{\text{aux}}_{kl} + \sigma^{\text{aux}}_{kl}\beta_{kl}\right) - \beta_{ki}\sigma^{\text{aux}}_{kj} - \beta^{\text{aux}}_{ki}\sigma_{kj}.$$

Interaction integrals are computed with numerical quadrature rules over the finite element mesh.

ACKNOWLEDGEMENTS

Adrian J. Lew thanks support from the Office of Naval Research Young Investigator Award (grant number N000140810852), the National Science Foundation Career Award (grant number CMMI-0747089), and the National Science Foundation (grant number CMMI-1301396). Yongxing Shen thanks support from the Marie Curie Careet Integration Grant (European Commission, grant number FP7-PEOPLE-2011-CIG-CompHydraulFrac).

DOI: 10.1002/nme

REFERENCES

1. Bittencourt TN, Wawrzynek PA, Ingraffea AR, Sousa JL. Quasi-automatic simulation of crack propagation for 2D LEFM problems. *Engineering Fracture Mechanics* 1996; **55**:321–334.
2. Phongthanapanich S, Dechaumphai P. Adaptive Delaunay triangulation with object-oriented programming for crack propagation analysis. *Finite Elements in Analysis and Design* 2004; **40**:1753–1771.
3. Azócar D, Elgueta M, Rivara MC. Automatic LEFM crack propagation method based on local Lepp–Delaunay mesh refinement. *Advances in Engineering Software* 2010; **41**:111–119.
4. Miehe C, Gürses E. A robust algorithm for configurational-force-driven brittle crack propagation with *r*-adaptive mesh alignment. *International Journal for Numerical Methods in Engineering* 2007; **72**(2):127–155.
5. Zielonka M, Ortiz M, Marsden J. Variational *r*-adaption in elastodynamics. *International journal for numerical methods in engineering* 2008; **74**(7):1162–1197.
6. Fraternali F. Free discontinuity finite element models in two-dimensions for in-plane crack problems. *Theoretical and Applied Fracture Mechanics* 2007; **47**(3):274–282.
7. Angelillo M, Babilio E, Fortunato A. Numerical solutions for crack growth based on the variational theory of fracture. *Computational Mechanics* 2012; **50**(3):285–301.
8. Giacomini A, Ponsiglione M. Discontinuous finite element approximation of quasistatic crack growth in nonlinear elasticity. *Mathematical Models and Methods in Applied Sciences* 2006; **16**(01):77–118.
9. Negri M. A discontinuous finite element approach for the approximation of free discontinuity problems. *Advances in Mathematical Sciences and Applications* 2005; **15**:283–306.
10. Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
11. Organ D, Fleming M, Terry T, Belytschko T. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational Mechanics* 1996; **18**(3):225–235. DOI: 10.1007/BF00369940.
12. Batra RC, Ching HK. Analysis of elastodynamic deformations near a crack/notch tip by the meshless local Petrov–Galerkin (MLPG) method. *Computer Modeling in Engineering and Sciences* 2002; **3**(6):717–730.
13. Sladek J, Sladek V, Krivacek J, Zhang C. Meshless local Petrov–Galerkin method for stress and crack analysis in 3-D axisymmetric FGM bodies. *Computer Modeling in Engineering and Sciences* 2005; **8**(3):259–270.
14. Chen W, Chen H. On three-dimensional fracture mechanics analysis by an enriched meshless method. *Computer Modeling in Engineering and Sciences* 2005; **8**(3):177–190.
15. Simo JC, Oliver J, Armero F. An analysis of strong discontinuities induced by strain-softening in rate-independent inelastic solids. *Computational Mechanics* 1993; **12**(5):277–296.
16. Oliver J. Continum modelling of strong discontinuities in solid mechanics using damage models. *Computational Mechanics* 1995; **17**:49–61.
17. Linder C, Armero F. Finite elements with embedded strong discontinuities for the modeling of failure in solids. *International Journal for Numerical Methods in Engineering* 2007; **72**(12):1391–1433.
18. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 1999; **45**:601–620.
19. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 1999; **46**:131–150.
20. Ortiz M, Pandolfi A. Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *International Journal for Numerical Methods in Engineering* 1999; **44**(9):1267–1282.
21. Laborde P, Renard Y, Pommier J, Salaün M. High order extended finite element method for cracked domains. *International Journal for Numerical Methods in Engineering* 2005; **64**:354–381.
22. Sudhakar Y, Wall WA. Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods. *Computer Methods in Applied Mechanics and Engineering* 2013; **258**:39–54.
23. Shen Y, Lew A. An optimally convergent discontinuous-Galerkin-based extended finite element method for fracture mechanics. *International Journal for Numerical Methods in Engineering* 2010; **82**:716–755.
24. Negri M, Ortner C. Quasi-static crack propagation by Griffith's criterion. *Mathematical Models and Methods in Applied Sciences* 2008; **18**(11):1895–1925.
25. Francfort G, Marigo JJ. Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids* 1998; **46**(8):1319–1342.
26. Negri M. From rate-dependent to rate-independent brittle crack propagation. *Journal of Elasticity* 2010; **98**(2):159–187.
27. Karma A, Lobkovsky AE. Unsteady crack motion and branching in a phase-field model of brittle fracture. *Physical Review Letters* 2004; **92**:245–510.
28. Bourdin B, Francfort G, Marigo J. The variational approach to fracture. *Journal of Elasticity* 2008; **91**(1):5–148.
29. Miehe C, Welschinger F, Hofacker M. Thermodynamically consistent phase-field models of fracture: variational principles and multi-field FE implementations. *International Journal for Numerical Methods in Engineering* 2010; **83**(10):1273–1311.
30. Miehe C, Hofacker M, Welschinger FA. Phase field model for rate-independent crack propagation: robust algorithmic implementation based on operator splits. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(45):2765–2778.
31. Borden MJ, Verhoosel CV, Scott MA, Hughes TJ, Landis CM. A phase-field description of dynamic brittle fracture. *Computer Methods in Applied Mechanics and Engineering* 2012; **217**:77–95.

32. Amiri F, Millán D, Shen Y, Rabczuk T, Arroyo M. Phase-field modeling of fracture in linear thin shells. *Theoretical and Applied Fracture Mechanics* 2014; **69**:102–109.
33. Negri M. Convergence analysis for a smeared crack approach in brittle fracture. *Interfaces and Free Boundaries* 2007; **9**(3):307.
34. Negri M. A finite element approximation of the Griffiths model in fracture mechanics. *Numerische Mathematik* 2003; **95**(4):653–687.
35. Schmidt B, Fraternali F, Ortiz M. Eigenfracture: an eigendeformation approach to variational fracture. *Multiscale Modeling & Simulation* 2009; **7**(3):1237–1266.
36. Hunsweck MJ, Shen Y, Lew AJ. A finite element approach to the simulation of hydraulic fractures with lag. *International Journal for Numerical and Analytical Methods in Geomechanics* 2013; **37**(9):993–1015.
37. Rangarajan R, Lew AJ. Universal meshes: a method for triangulating planar curved domains immersed in nonconforming triangulations. *International Journal for Numerical Methods in Engineering* 2014; **98**(4):236–264.
38. Rangarajan R, Lew AJ. Parameterization of planar curves immersed in triangulations with application to finite elements. *International Journal for Numerical Methods in Engineering* 2011; **88**(6):556–585.
39. Sukumar N, Prévost JH. Modeling quasi-static crack growth with the extended finite element method, part I: computer implementation. *International Journal of Solids and Structures* 2003; **40**:7513–7537.
40. Cotterell B, Rice JR. Slightly curved or kinked cracks. *International Journal of Fracture* 1980; **16**:155–169.
41. Chiaramonte MM, Shen Y, Keer LM, Lew AJ. Computing stress intensity factors for curvilinear fractures 2014. In preparation.
42. Rangarajan R, Lew AJ. Analysis of a method to parameterize planar curves immersed in triangulations. *SIAM Journal on Numerical Analysis* 2013; **51**(3):1392–1420.
43. Rangarajan R, Lew AJ. Directional vertex relaxation for geometric mesh optimization. Preprint.
44. Rangarajan R. Universal meshes: a new paradigm for computing with nonconforming triangulations. *PhD Thesis*, Stanford University, 2012.
45. Bank RE, Smith RK. Mesh smoothing using *a posteriori* error estimates. *SIAM Journal on Numerical Analysis* 1997; **34**(3):979–997.
46. De Berg M, Van Kreveld M, Overmars M, Schwarzkopf OC. *Computational Geometry*. Springer: Berlin, 2000.
47. Bern M, Eppstein D, Gilbert J. Provably good mesh generation. *Journal of Computer and System Sciences* 1994; **48**:384–409.
48. Muskhelishvili NI. *Some Basic Problems of the Mathematical Theory of Elasticity: Fundamental Equations, Plane Theory of Elasticity, Torsion, and Bending (Translated from Russian)* (Second edn.) Noordhoff International Publishing: Leyden, The Netherlands, 1977.
49. Ingraffea AR, Grigoriu M. Probabilistic fracture mechanics: a validation of predictive capability. *Technical Report*, Cornell University, 1990.
50. Rao BN, Rahman S. An efficient meshless method for fracture analysis of cracks. *Computational Mechanics* 2000; **26**:398–408.
51. Chiaramonte MM. A Python implementation to compute the elastic fields around a circular arc crack, 2014. Available as supplementary information for the manuscript.
52. Chambolle A, Francfort GA, Marigo JJ. When and how do cracks propagate? *Journal of the Mechanics and Physics of Solids* 2009; **57**:1614–1622.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of this article at the publisher's web site.

DOI: 10.1002/nme