INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2010; **84**:1303–1343
Published online 28 September 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.2943

Adaptive dynamic cohesive fracture simulation using nodal perturbation and edge-swap operators

Glaucio H. Paulino[1,*,†], Kyoungsoo Park[1], Waldemar Celes[2] and Rodrigo Espinha[2]

[1]*Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Ave., Urbana, IL 61801, U.S.A.*
[2]*Tecgraf/PUC-Rio Computer Science Department, Pontifical Catholic University of Rio de Janeiro, Rua Marquês de São Vicente 225, Rio de Janeiro, RJ, 22450-900, Brazil*

SUMMARY

Dependence on mesh orientation impacts adversely the quality of computational solutions generated by cohesive zone models. For instance, when considering crack propagation along interfaces between finite elements of **4k** structured meshes, both extension of crack length and crack angle are biased according to the mesh configuration. To address mesh orientation dependence in **4k** structured meshes and to avoid undesirable crack patterns, we propose the use of nodal perturbation (NP) and edge-swap (ES) topological operation. To this effect, the topological data structure TopS (*Int. J. Numer. Meth. Engng* 2005; **64**: 1529–1556), based on topological entities (node, element, vertex, edge and facet), is utilized so that it is possible to access adjacency information and to manage a consistent data structure in time proportional to the number of retrieved entities. In particular, the data structure allows the ES operation to be done in constant time. Three representative dynamic fracture examples using ES and NP operators are provided: crack propagation in the compact compression specimen, local branching instability, and fragmentation. These examples illustrate the features of the present computational framework in simulating a range of physical phenomena associated with cracking. Copyright © 2010 John Wiley & Sons, Ltd.

Received 22 January 2009; Revised 5 April 2010; Accepted 17 April 2010

KEY WORDS: PPR; potential-based model; topological data structure; **4k** mesh; nodal perturbation; edge-swap; dynamic fracture; extrinsic cohesive zone model

1. INTRODUCTION AND OPENING REMARKS

Dynamic fracture phenomena have been investigated by means of several methods, including numerical techniques, in conjunction with non-linear cohesive zone models. In the finite element method, a cohesive zone (or non-linear fracture process zone) is approximated by cohesive surface elements, which contain the constitutive relationship of fracture surfaces. Cohesive surface elements

*Correspondence to: Glaucio H. Paulino, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Ave., Urbana, IL 61801, U.S.A.
†E-mail: paulino@uiuc.edu

Table I. Geometrical and topological considerations for cohesive zone model simulations.

Finite element mesh	Geometrically	Topologically
4k	Structured	Structured
Nodal perturbation	Unstructured	Structured
Edge swap	Structured	Unstructured
Nodal perturbation and edge swap	Unstructured	Unstructured

can be inserted either before or during computational simulation. The former approach requires a cohesive surface network within the crack path domain [1]. In this case, a crack propagation criterion is naturally incorporated with the constitutive relation, leading to the *intrinsic cohesive zone model*. On the other hand, cohesive surface elements can be adaptively inserted during the computational simulation whenever and wherever they are necessary [2]. This model requires an external criterion for the insertion of cohesive surface elements, leading to the *extrinsic cohesive zone model*.

In finite element analysis, **4k** structured meshes are widely utilized while possessing several features of interest. They are easily generated, even for irregular domains, and are convenient to perform mesh refinement with high-quality discretization. As illustrated by Table I, a regular **4k** mesh is structured both in the topology sense and in the geometry sense. In fact, being topologically structured is one of the major advantages of **4k** meshes. The mesh can be represented by a hierarchical topological subdivision, which is suitable for mesh refinement and coarsening. Simple, efficient and effective topological operators can be applied in order to locally adapt the mesh [3, 4].

On the other hand, structured meshes may represent a drawback. For instance, the structured geometry may impose mesh bias in the case of crack propagation simulation, even for highly refined meshes. To address this problem, Papoulia *et al.* [5] have proposed the use of pinwheel-based meshes for dynamic crack propagation problems. Because the pinwheel-tiling possesses the 'isoperimetric property', any curve in the two-dimensional (2D) domain can be represented by edges of a pinwheel-based mesh, as the element size tends to zero (i.e. in the limit sense). However, elements used in numerical analysis are of finite size, generation of pinwheel mesh is non-standard and the aspect ratio of the resulting mesh, although bounded, still needs improvements [6]. Therefore, an alternative approach (based on revisited **4k** meshes) is presented in this paper.

The discrepancy between the mathematical path of a crack and the path represented by a set of mesh edges can be measured by comparing the associated path lengths. Convergence of path length is meaningful because the energy needed to create a crack is related to its length [5].

Nevertheless, besides length convergence, one should also analyze crack path deviation, especially for regular meshes. In quantitative terms, path deviation can be estimated by the Hausdorff distance. Given two arbitrary curves P and Q, with p and q denoting points on those curves, the Hausdorff distance is defined as [7]

$$H(P, Q) = \max(h(P, Q), h(Q, P)) \tag{1}$$

where

$$h(P, Q) = \max_{p \in P} \left[\min_{q \in Q} [\text{dist}(p, q)] \right] \tag{2}$$

DOI: 10.1002/nme

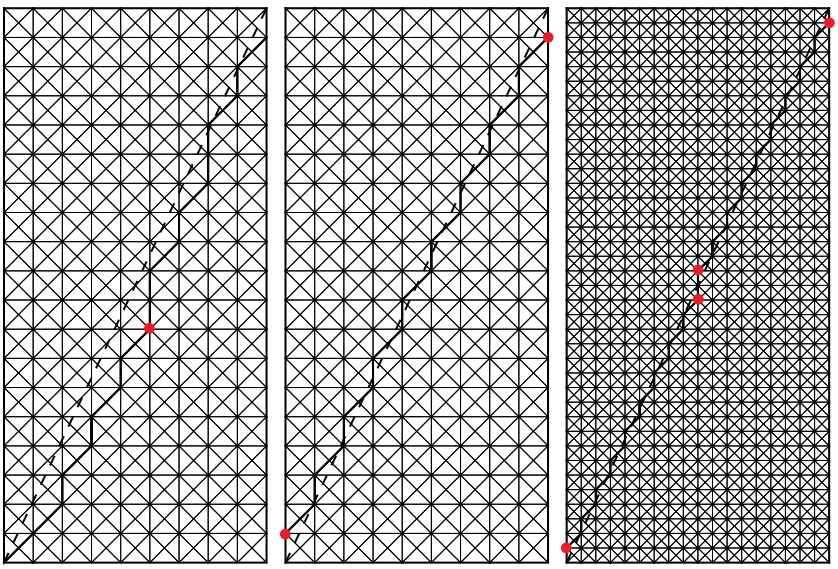


Figure 1. Discrepancy between a mathematical path (thick dashed line) and a discrete path (thick solid line). The aspect ratio of the rectangular domain (**4k** structured mesh) is 1: 2.11.

By considering P as the discretized path, where p are the vertices along that path, and Q the mathematical path, represented by a line segment, and considering that the endpoints are coincident in both discretized and mathematical paths, the Hausdorff distance to compute crack path deviation reduces to:

$$H(P, Q) = h(P, Q) \tag{3}$$

Figure 1 illustrates the importance of analyzing both length convergence and crack path deviation. The locations of the maximum deviation are described in solid circles. Figure 1(a), obtained from reference [5], illustrates the discrepancy between a mathematical path and a path represented by a **4k** structured mesh, where the Hausdorff distance ($H(P, Q)$) is 0.12156. With the same mesh discretization, the Hausdorff distance can be reduced even further by selecting another discretized shortest path, as shown in Figure 1(b), i.e. $H(P, Q) = 0.04757$. Moreover, Figure 1(c) demonstrates that the mesh refinement of the **4k** structured mesh leads to further reduction of the Hausdorff distance, i.e. $H(P, Q) = 0.02378$.

However, there is always a finite error between the mathematical length and the length represented by a **4k** structured mesh. For instance, Papoulia *et al.* [5] stated that '*no matter how much the mesh is refined, the jagged path will always be approximately 8% longer than the mathematical path*' for standard **4k** structured meshes. To further illustrate, a simple geometric problem is constructed within the 1×1 square domain ($OABC$), shown in Figure 2(a). The point P is located between the upper left corner (point A) and the upper right corner (point B). The exact distance (ℓ_{exact}) from the origin (point O) to the point P is $\sqrt{1+x^2}$, i.e. *mathematical length*, where x is a distance between the point A and the point P. Similarly, the path ($\overline{OP}$) can be also represented by a finite element mesh under the assumption that the path is only described by element boundaries

DOI: 10.1002/nme

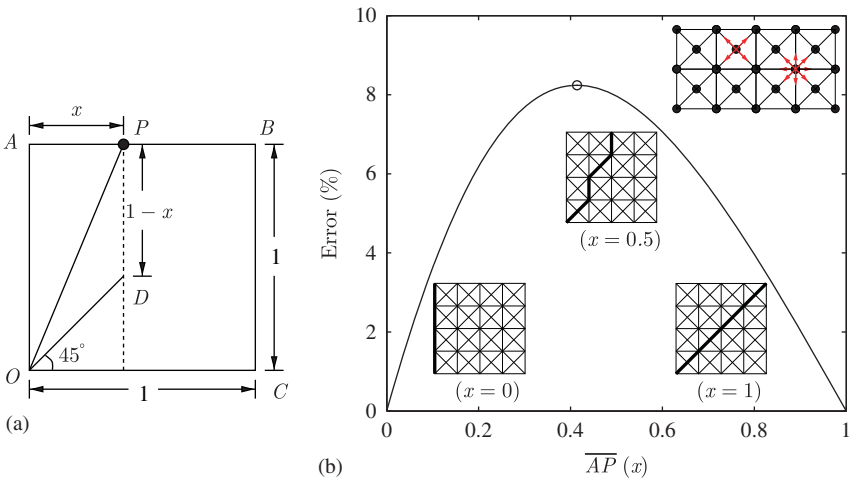


Figure 2. (a) Crack path test and (b) error between the *mathematical* length ($\overline{OP}$) and the *discrete length* represented by the **4k** structured mesh. The maximum error occurs at $x=\sqrt{2}-1$ (angle $\angle COP=67.5^\circ$).

(i.e. the path cannot go through the elements). If one utilizes **4k** structured meshes, the path ($\overline{OP}$) consists of piece-wise linear lines with angles of 45 and 90°. Therefore, the length represented by a **4k** structured finite element mesh (ℓ_{FE}) is the summation of the length $\overline{OD}$ ($\sqrt{2}x$) and the length $\overline{DP}$ ($1-x$) (see Figure 2(a)), i.e. *discrete length*. Figure 2(b) demonstrates the relative error ($|\ell_{\text{FE}}-\ell_{\text{exact}}|/\ell_{\text{exact}}$) of the discrete length represented by the **4k** structured finite element mesh with respect to the location of the point P ($\overline{AP}(x)$). The **4k** structured mesh is able to represent the exact path when $x=0$ or $x=1$, although it provides error up to 8.24 % for the intermediate range. The maximum error occurs when the distance $\overline{AP}(x)$ is equal to $x=\sqrt{2}-1$ and the angle ($\angle COP$) of 67.5°, which is the mean value of 45 and 90°. Note that the error is not relevant to the Hausdorff distance, and thus the mesh refinement does not reduce the error for **4k** structured mesh.

Another feature of **4k** structured meshes is that some internal nodes have eight adjacent edges, whereas others have four adjacent edges, as shown in the insert of Figure 2(b). In other words, when cohesive crack propagation is considered, some nodes have eight potential directions at 45° increments, whereas others have four potential directions at 90° increments. A node that has eight potential directions can have maximum error of 22.5° for the selection of a crack propagation direction, whereas a node that has four potential directions can have maximum error of 45°. Consequently, computational simulations using a **4k** structured mesh with cohesive elements may lead to undesirable (e.g. zigzag) crack patterns.

In order to reduce the error between the mathematical length and a length represented by edges of a **4k** mesh, and to alleviate zigzag crack patterns, this paper proposes the use of nodal perturbation (NP) and/or edge-swap (ES) topological operator. The original **4k** meshes are geometrically and topologically structured. The NP leads to geometrically unstructured meshes, whereas the ES provides locally unstructured meshes (topologically). Thus, both the NP and the ES operator result in geometrically and topologically unstructured meshes, as summarized in Table I. As proof-of-concept, the effects of NP and ES operator on representation of crack length, angle and path

DOI: 10.1002/nme

deviation are investigated in this work. *Computational experiments demonstrate that*, *for practical levels of mesh refinement*, *discrepancy between mathematical and discretized paths reduces more rapidly for modified* **4k** *meshes* (*as proposed in this paper*) *than for pinwheel-based meshes*.

This paper investigates dynamic mixed-mode crack propagation, microbranching, and fragmentation by means of **4k** meshes with NP and/or ES operator. A topological data structure, called TopS [4, 8], is employed to maintain adjacency relationships under adaptive insertion of cohesive elements and ES operation. A unified potential-based model, called PPR [9], is used, which provides a consistent constitutive relationship for mixed-mode fracture. This paper is organized as follows. First, to provide context, related work is briefly reviewed. Next, geometrical improvement of **4k** meshes is addressed by introducing NP (Section 3), whereas topological improvement is addressed by employing ES operator (Section 4). Path convergence is quantified by comparing **4k** meshes with pinwheel-based meshes in Section 5. The client–server approach of the topological data structure TopS is employed in conjunction with application programming interfaces (APIs) and callback functions in Section 6. Sections 7 presents the computational simulation framework and the potential-based PPR model including the unloading/reloading relationship. Section 8 provides three numerical examples: compact compression specimen (CCS) tests, microbranching experiments and fragmentation. Some remarks on topology of **4k** meshes and pinwheel-based meshes are given in Section 9. Finally, Section 10 summarizes the key findings of the paper.

2. RELATED WORK

Dynamic fracture problems have been investigated by employing several computational simulation frameworks. For instance, Swenson and Ingraffea [10] used finite elements for linear elasto-dynamic mixed-mode crack propagation problems in conjunction with remeshing techniques. Abraham *et al.* [11] employed molecular dynamics simulations with parallel computing to investigate the dynamic instability of a crack tip. Klein *et al.* [12] utilized the virtual internal bond model [13], which connects microscopic behavior with macroscopic behavior, to investigate dynamic fracture and microbranching with a meshfree method.

Moes and Belytschko [14] and Wells and Sluys [15] represented traction-free crack and cohesive zone by using discontinuous enrichment functions in the context of the extended/generalized finite element method (X-FEM/GFEM). Duarte *et al.* [16] represented through-the-thickness three-dimensional (3D) Y-shaped branch cracks for arbitrary background finite element meshes with high-order enrichment functions. However, non-polynomial enrichment functions in the X-FEM/GFEM lead to additional computational costs in numerical integration [17]. Moreover, according to Bishop [18], '*once crack branching and crack coalescence phenomena appear, the prospect of modeling a multitude of arbitrary three-dimensional intersecting cracks quickly becomes untenable*'. In addition, calculations based on a level set discontinuity tracking method [19] have difficulties in representing complicated crack patterns, for example, micro-crack branching that exists around a major crack in dynamic fracture problems. To circumvent this problem, Song and Belytschko [20] introduced a set of discrete crack segments to represent 2D complicated dynamic fracture patterns. Linder and Armero [21, 22] embedded discontinuities such as strain jumps at the element level, and extended the embedded discontinuities to the so-called 'T-shaped' discontinuities for dynamic branching problems.

As indicated earlier, a fracture process zone can be approximated by cohesive surface elements, which are adaptively inserted during computation. Ortiz and Pandolfi [23] developed irreversible

DOI: 10.1002/nme

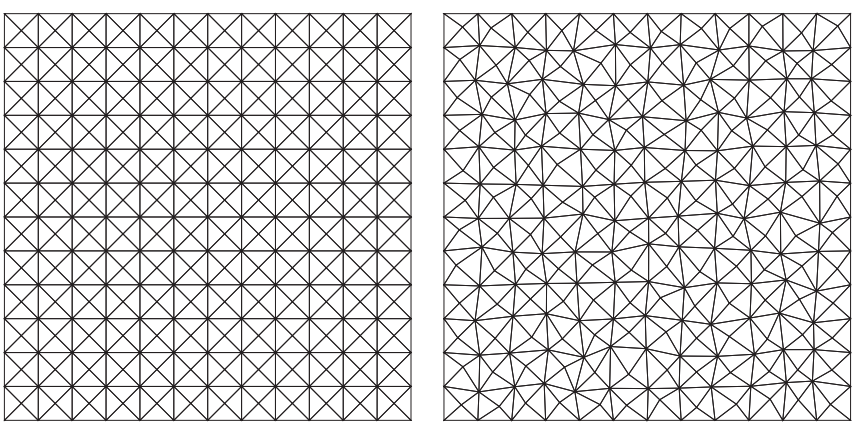
extrinsic cohesive elements for 3D dynamic fracture problems. Papoulia *et al.* [5, 24] addressed time continuity at the time of cohesive surface element activation, and errors in representing cracks in a finite element mesh. Zhou and Molinari [25] and Molinari *et al.* [26] selected cohesive strength based on a modified weakest link Weibull distribution while addressing mesh dependency and energy convergence.

The *extrinsic cohesive zone model* with cohesive surface elements is the choice for the current study. This approach leads to spontaneous crack initiation, branching and fragmentation for both 2D [2, 25, 27] and 3D [23, 28, 29] fracture problems. Thus, one can describe, for example, both macroscopic cracks and microscopic cracks within the same computational framework. In this study, the NP and the ES operators are introduced in order to reduce errors in representing crack path, as discussed in the following sections.

3. TOWARD UNSTRUCTURED GEOMETRY

In order to reduce the maximum error of the discrete length that exists in the **4k** structured mesh, the internal nodes of a mesh are randomly perturbed (or shaken), with respect to an NP factor and a given mesh quality metric. The NP factor is the ratio of the perturbed distance to the minimum distance between the central node of the **4k** pattern and the outer edges of the pattern, whereas a mesh quality metric is employed for a Laplacian smoothing. The algorithm for the NP is explained in Section 6, and the effect of the NP factors on a **4k** structured mesh is illustrated in Figure 3. The NP factors used in this study are 0.0 (unperturbed), 0.1, 0.2 and 0.3 with a fixed mesh quality parameter (e.g. Lo's parameter [30]). The higher the NP factor, the more the randomness in the finite element mesh.

The mesh quality is estimated by the Lo's parameter [30], which is defined as $4\sqrt{3}A/\sum_{i=1}^{3}\ell_i{}^2$ where A is the area of a triangle and ℓ_i represents the length of edges. The parameter varies from 0 to 1. Lo's parameter is zero when the area of a triangle is equal to zero, and one for an equilateral triangle. Within a square domain of (1 : 1), a **4k** mesh grid of 40 × 40 is constructed,



(a) (b)

Figure 3. Effect of the nodal perturbation (NP) factor on a finite element mesh: (a) NP = 0.0 (unperturbed) and (b) NP = 0.3.

DOI: 10.1002/nme

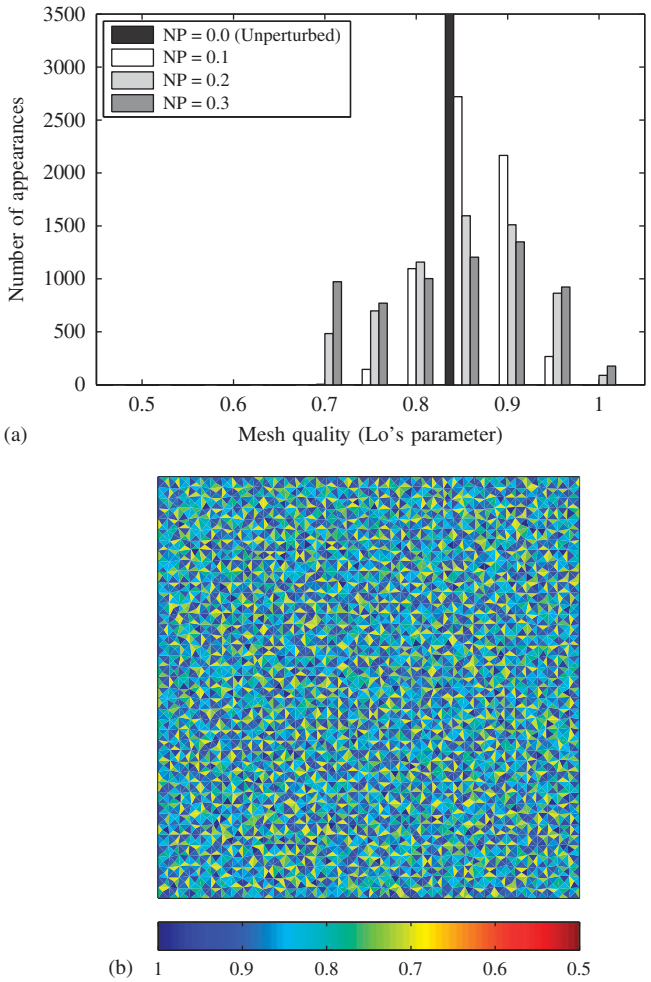


Figure 4. Mesh quality estimation: (a) histogram and (b) Lo's parameter for each element considering NP = 0.3. Notice that all 6400 elements in the unperturbed mesh (NP = 0.0) have Lo's parameter of 0.866.

resulting in 6400 triangular elements, with the NP factors of 0.0 (unperturbed), 0.1, 0.2 and 0.3. The minimum mesh quality parameter is selected as 0.7, which is generally acceptable for finite element simulations. The histogram of the mesh quality with respect to the NP factors is illustrated in Figure 4(a). Lo's parameter of all elements is 0.866 for the zero NP factor (NP = 0.0). The increase of the NP factor leads to a broader distribution of Lo's parameter. Figure 4(b) demonstrates the spacial distribution of Lo's parameter for each element in the 40×40 **4k** mesh with NP = 0.3.

In the following subsections, the effects of the NP factor are investigated by two examples: crack length convergence and crack angle convergence. The crack length convergence investigation compares a mathematical length with a length represented by edges of a finite element mesh, whereas the crack angle convergence investigation compares an arbitrarily given angle with a geometrically averaged angle.

DOI: 10.1002/nme

3.1. Crack length convergence

For the crack length convergence investigation, the exact distance (ℓ_{exact}) between two points is compared with the shortest length along edges of finite element meshes (ℓ_{FE}). One can find the shortest geometrical crack length between two arbitrary points by using, for instance, the Dijkstra's algorithm [31]. In this study, we employ two cases: the worst and the best cases in a structured **4k** mesh for the representation of a crack geometry. For the worst case, we have a 1×2.4 rectangular domain so that a given problem provides approximately the maximum error of a crack length in the **4k** structured mesh. The exact crack length from the lower left corner $(x, y) = (0, 0)$ to the top right corner $(x, y) = (1, 2.4)$ is 2.6, and the angle of the path is $\tan^{-1}(2.4) = 67.38^\circ$, which is close to the mean value of 45° and 90°. The rectangular region is discretized into 5×12, 10×24, 20×48 and 40×96 finite element meshes whose element sizes (i.e. the longest edge in an element) are 0.2, 0.1, 0.05 and 0.025, respectively. The internal nodes of each mesh are perturbed with NP = 0.0 (unperturbed), 0.1, 0.2 and 0.3.

The effect of the NP factors on the discrete crack length is investigated with the mesh grid of 10×24 (element size = 0.1). One hundred meshes are generated and tested for each of the NP factors, and all the results are plotted using white circles, as shown in Figure 5(a). The mean and the standard deviation of error are plotted in solid line and dashed-dot lines, respectively. When the NP factor is equal to zero (NP = 0.0), the shortest length in the finite element mesh is always 2.814 and its error is 8.24%. However, the increase in the NP factor results in a broader distribution of error level and lower averaged error, as shown in Figure 5(a). Next, with a NP factor of 0.3 (NP = 0.3), the influence of element sizes on the discrete crack length is investigated and the results are plotted in Figure 5(b). The larger the element size used, the wider is the range of error level. Moreover, notice that *mesh refinement reduces both maximum and averaged error level*, as shown in Figure 5(b).

For the best case scenario, a **4k** structured mesh is able to represent the mathematical length along the 0, 45 and 90° directions. Thus, a 2.4×2.4 square domain is discretized into a 24×24 **4k** mesh. The mathematical length between the lower left corner $(0, 0)$ and the upper right corner $(2.4, 2.4)$ is compared with the shortest length represented by edges of a **4k** mesh with the NP factors of 0.0 (unperturbed), 0.1, 0.2 and 0.3. Similarly, 100 meshes are generated for each NP factor. All the results are plotted using white circles, and the averaged error and the standard deviation are described with solid line and dashed-dot lines, respectively, as shown in Figure 6. The increase of NP leads to the increase of the error in crack length because a **4k** structured mesh (i.e. no NP) is able to precisely represent a straight line along the 45° direction. However, the additional error along the 45° direction, introduced by the NP, is less than the error reduction along the 67.38° direction due to the NP (cf. Figure 5(a)).

3.2. Crack angle convergence

For the crack angle convergence study, an arbitrarily given angle (θ) is compared with a geometrically obtained angle (θ_{FE}). An averaged geometric angle (θ_{FE}) is obtained as follows. First, from a source point, one locally searches the closest edge direction with respect to a given angle (θ). Next, along the edge direction, one moves the source point to the adjacent point. One repeats the previous procedure until the source point reaches the domain boundary. Then, an averaged geometric angle is approximated by connecting the first source point to the last source point.

DOI: 10.1002/nme

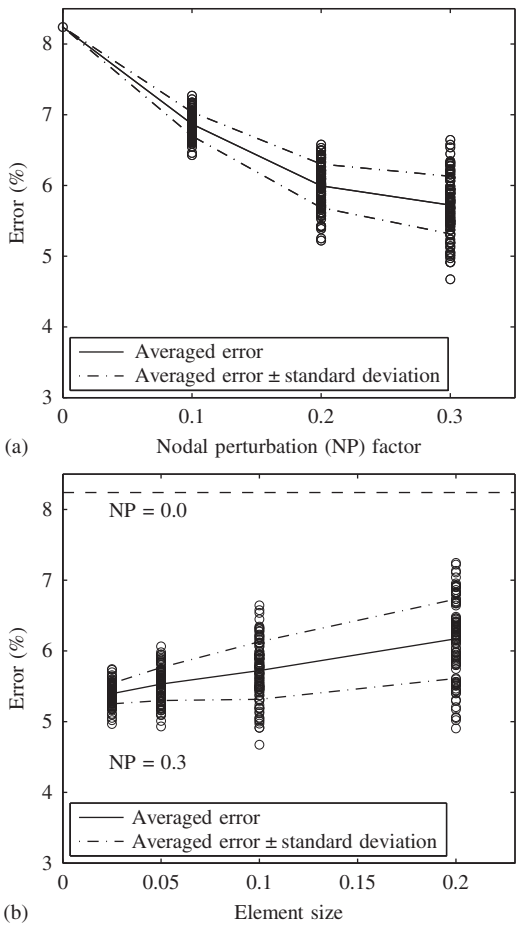


Figure 5. Error with respect to (a) nodal perturbation (NP) factor considering 10×24 mesh grid and (b) element size considering NP$=0.3$.

In this study, the arbitrarily angle (θ) is first selected as 67.38° within the 2.4×2.4 square domain (e.g. dashed line in Figure 7), which is the same as the angle in the previous length investigation. The domain is discretized into the 24×24 mesh grid, which leads to an element size of 0.1. With the NP factors of 0.0 (unperturbed), 0.1, 0.2 and 0.3, 200 meshes are generated for each NP factor. When the NP factor is equal to zero, the geometrically obtained angle is always 45°, as shown in Figure 7(a). This is because 45° is always the closest direction for the target angle ($\theta = 67.38^\circ$). However, the NP factor improves the results of crack angle convergence. For example, Figure 7(b) shows the result of crack angle convergence using the NP factor of 0.3 (NP$=0.3$) and the resulting averaged geometric angle of 59.7°. The histogram of the geometrically obtained angles (θ_{FE}) is plotted in Figure 8 with respect to the NP factors. The Gaussian curves are fitted to the histogram for each NP factor. The NP factor of 0.1 (NP$=0.1$) results in the most number of appearance of θ_{FE} around 61.6°, and the NP factors of 0.2 (NP$=0.2$) and 0.3 (NP$=0.3$) lead to

DOI: 10.1002/nme

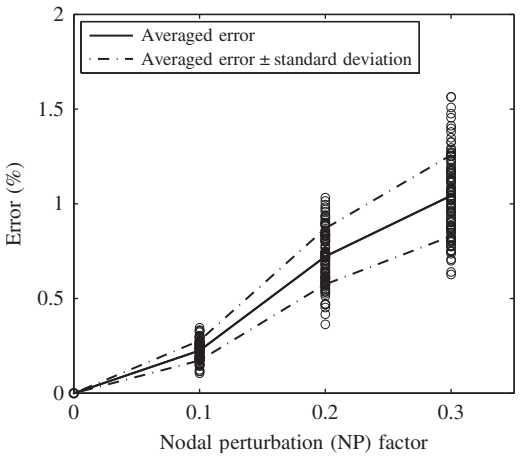


Figure 6. Error with respect to the nodal perturbation (NP) factor (along the 45° direction).

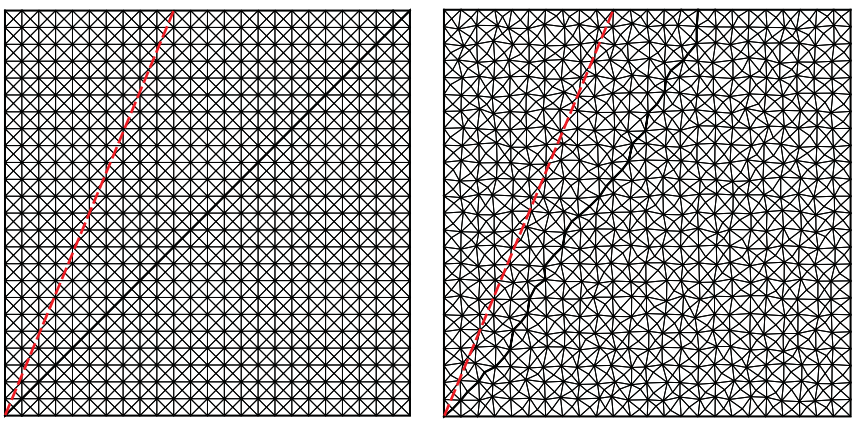


Figure 7. Representative results of crack angle convergence for (a) nodal perturbation (NP) factor of 0.0 (unperturbed) and (b) NP = 0.3.

the highest number of appearances around 59.7°. The distribution of the number of appearances demonstrates similar pattern (e.g. a bell-shape) across the NP factors. Thus, the NP factors ranging from 0.1 to 0.3 provide a similar degree of error level in this crack angle convergence example.

When the arbitrary angle (θ) is selected as 45°, the geometrically obtained angle (θ_{FE}) is always 45° for all NP factors (i.e. 0.1, 0.2 and 0.3). Therefore, the NP improves the results of crack angle convergence for the arbitrarily angle of 67.38° without introducing error in the geometrically obtained angle for the arbitrary angle of 45°.

DOI: 10.1002/nme

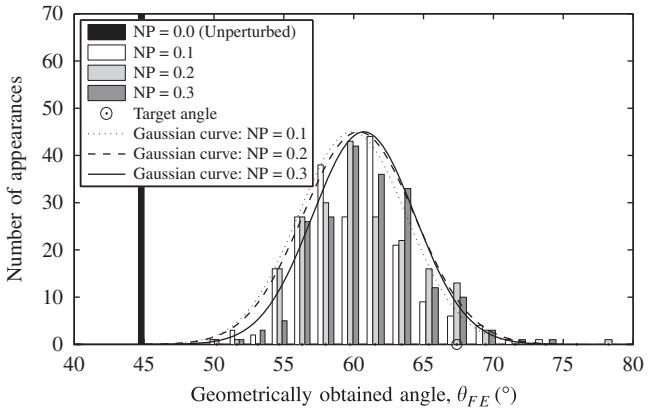


Figure 8. Number of appearances of geometrically obtained angles (θ_{FE}) for given nodal perturbation (NP) factors.

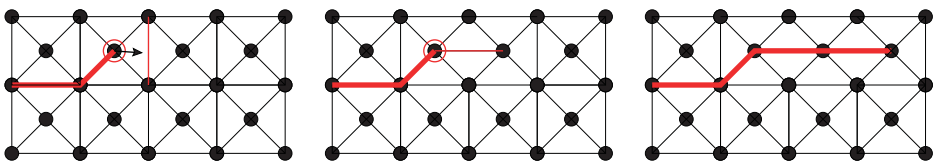


Figure 9. Three successive mesh instances showing a schematic description of the edge-swap (ES) operator.

4. TOWARD UNSTRUCTURED TOPOLOGY

In order to reduce undesirable crack patterns (e.g. zigzag) and to further improve the crack angle and length convergence, the ES topological operator is introduced. This operator requires local change of connectivity information. However, it does not introduce additional nodes, and therefore, the original geometry, with respect to nodal location, remains the same for **4k** structured meshes.

Owing to the ES operator, all the internal nodes have the same number of potential directions. In **4k** structured meshes with the ES operator, the angle between two potential adjacent edges is initially 45°, and thus the maximum error of the crack direction is locally 22.5°. The ES operator in **4k** meshes, therefore, can lead to smoother crack patterns in computational simulations than without the ES operator. For example, if physics dictates a crack growth along the horizontal direction (Figure 9(a)), one can activate the ES topological operator (Figure 9(b)), and the crack patterns become smoother (e.g. Figure 9(c)). The ES operator has also been utilized in a randomized incremental algorithm for the construction of planar Voronoi diagrams and Delaunay triangulations [32].

In order to further investigate the influence of the ES operator, the crack length convergence and the crack angle convergence examples are also utilized in this section. For comparison purpose, the two examples are the same as the ones in Section 3.

DOI: 10.1002/nme

4.1. Crack length convergence

The 1×2.4 domain is discretized into a 10×24 grid. With the NP factors of 0.0, 0.1, 0.2 and 0.3, 100 meshes are generated for each NP factor. The shortest discrete length is obtained in conjunction with the ES operator. The results are described by white circles and the averaged error is plotted with solid line (Figure 10(a)). For comparison purpose, the averaged error without the ES operator (shown in Figure 5(a)) is plotted with dashed-dot line. According to Figure 10(a), the increase of the NP factor results in the decrease of the averaged error and a wider distribution of the shortest discrete length. The use of the ES operator reduces the averaged error (Figure 10(a)) compared with the case without the ES (Figure 5(a)). Additionally, within the same domain, the effect of element size (0.2, 0.1, 0.05, 0.025) on a discrete crack length is observed in conjunction with the ES operator and NP $= 0.3$. Figure 10(b) illustrates the error of discrete length with respect to element size. The mesh refinement leads to the decrease of the averaged error and a narrower distribution of the shortest discrete length. The ES operator further reduces the error compared with the case without the ES (Figure 5(b)).

4.2. Crack angle convergence

The 2.4×2.4 domain is discretized into 24×24 finite element mesh grid, and NP factors of 0, 0.1, 0.2 and 0.3 are applied to the finite element mesh. Similar to the investigation in Section 3.2, for a given angle ($\theta = 67.38^\circ$), an averaged geometrical angle (θ_{FE}) is obtained by searching locally for a direction, which is the closest angle of an edge to the given angle. When a node has four original possible directions, four additional directions, which are associated with the ES, are also considered for potential search directions. Figure 11 illustrates the number of appearances of the geometrically obtained angles (θ_{FE}) from the finite element meshes, considering the ES operator. The Gaussian distribution is fitted to the histogram, as shown in Figure 11. The highest number of appearance occurs at the angle of 71.6° for the NP factors of 0.1 and 0.2, whereas the highest number of appearance is at the angle of 69.4° for the NP factor of 0.3. These angles (with the ES) are closer to the target angle than the previous investigation (without the ES, shown in Figure 8). Moreover, the distribution of the geometrically averaged angles with the ES operator is narrower than the distribution without the ES operator (compare Figures 11 and 8).

5. ON PATH CONVERGENCE

Path convergence of **4k** meshes with NP and ES is quantified in conjunction with a computational experiment. First, the isoperimetric property is estimated by assessing length convergence of paths that connect pairs of vertices in the meshes to the corresponding straight lines. The results are compared with pinwheel-based meshes, which are known to possess the isoperimetric property in the limit sense [5, 33]. Then, the Hausdorff distance of the shortest path in each mesh, with respect to a straight line, is computed and compared.

A sample rectangular domain (Ω) with dimensions 2×1 is discretized by four types of finite element meshes (Ω_{FE}). The first mesh type (Type I) is obtained from a pinwheel-tiling by splitting triangles that have a hanging node [6] (e.g. Figure 12(a)). Pinwheel tiles are described by thick solid lines, whereas additional edges, which eliminate hanging nodes in pinwheel tiles, are illustrated by thin solid lines in Figure 12(a). The second type (Type II) is generated from a pinwheel-tiling (thick solid line in Figure 12(b)) by applying a triangular subdivision (thin solid line in Figure 12(b)).

DOI: 10.1002/nme

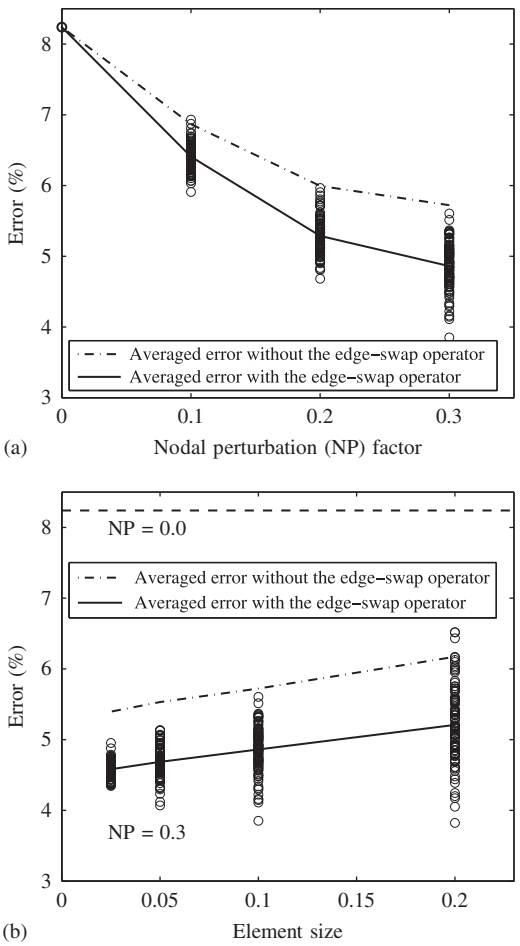


Figure 10. Error with respect to (a) the nodal perturbation factor and (b) the element size in conjunction with the edge-swap operator.

Notice that Figures 12(a) and (b) have the same pinwheel-tiling (i.e. same thick solid lines) but different ways to handle hanging nodes (i.e. different thin solid lines). Next, **4k** meshes are generated by using the NP factor of 0.3 without the ES operator (Type III), as shown in Figure 12(c). Finally, **4k** meshes with the NP (NP = 0.3) and the ES operators are employed (Type IV). Available ES operations in a **4k** mesh, for example, are described by the thick dashed line used for crack path representation (Figure 12(d)).

In order to quantify the isoperimetric property of these four mesh types, we utilize the *ρ-path deviation ratio* [6], which is defined as

$$\mathrm{dev}_{\rho}(\Omega_{\mathrm{FE}}) = \max\left(\frac{\ell_{\mathrm{FE}}(p,q)}{\|p-q\|} : p,q \in V(\Omega_{\mathrm{FE}}) \text{ and } \|p-q\| \geqq \rho\right) \tag{4}$$

DOI: 10.1002/nme

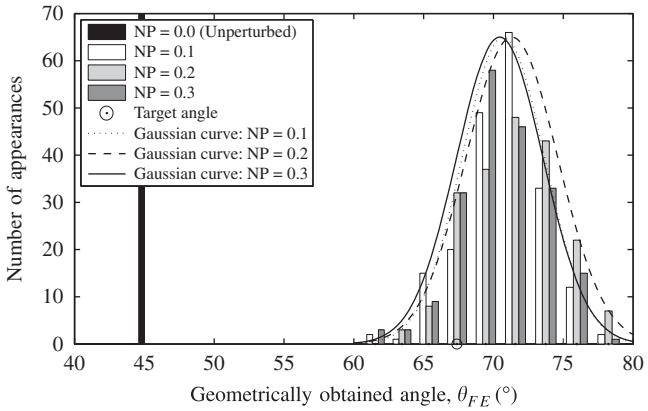


Figure 11. Number of appearances of geometrically obtained angles (θ_{FE}) with respect to the nodal perturbation (NP) factor in conjunction with the edge-swap (ES) operator.

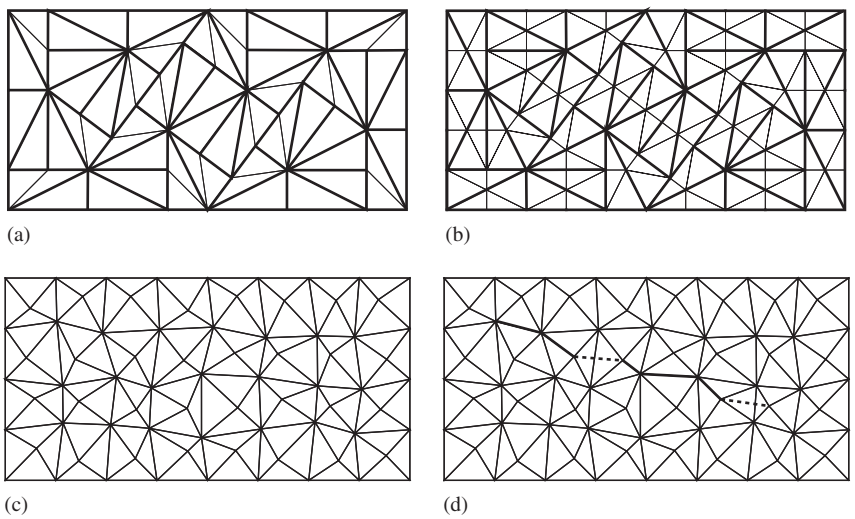


Figure 12. Mesh-type examples: (a) Type I: pinwheel (hanging nodes); (b) Type II: pinwheel (triangulation); (c) Type III: **4k** (NP); and (d) Type IV: **4k** (NP & ES).

where $V(\Omega_{\text{FE}})$ is the set of all vertices of a finite element mesh (Ω_{FE}) and $\|p-q\|$ is the distance from p to q. In addition, $\ell_{\text{FE}}(p,q)$ is the shortest path represented by edges of a finite element mesh (see Sections 3 and 4). A positive parameter ρ is selected as 1 so that the 2×1 domain contains a disk of diameter ρ, shown in Figure 13(a). The parameter ρ limits the range of the considered pairs of vertices (p,q) to the ones whose distance to each other is greater than or equal to ρ (Figure 13(b)). The ρ-path deviation ratio ($\text{dev}_\rho(\Omega_{\text{FE}})$) is the maximum relative error in representing a straight line by utilizing a finite element mesh (Ω_{FE}). Notice that Ganguly *et al.* [6]

DOI: 10.1002/nme

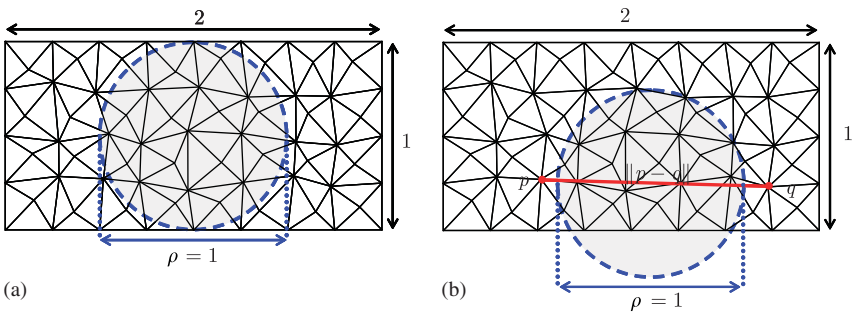


Figure 13. (a) Illustration of the parameter ρ in the rectangular domain and (b) constraint for the set of vertices (p, q) related to the parameter ρ.

Table II. ρ-path deviation ratio (dev_ρ) and the Hausdorff distance ($H(p,q)$) with respect to the number of elements.

Mesh type I			Mesh type II			Mesh type III			Mesh type IV	
# of elem.	dev_ρ	$H(p,q)$	# of elem.	dev_ρ	$H(p,q)$	# of elem. (grid)	dev_ρ	$H(p,q)$	dev_ρ	$H(p,q)$
16	1.3416	0.4	6	1.1441	0.3536	8 (2×1)	1.1700	0.3986	1.0964	0.4158
68	1.1948	0.3123	30	1.1749	0.2481	32 (4×2)	1.2143	0.2511	1.1115	0.2123
360	1.1843	0.0917	150	1.1755	0.12	128 (8×4)	1.1523	0.1926	1.0887	0.1672
1764	1.1264	0.0675	750	1.1156	0.1136	648 (18×9)	1.1062	0.1386	1.0700	0.0996
8880	1.0831	0.0552	3750	1.0827	0.0333	1568 (28×14)	1.0904	0.1031	1.0650	0.0958
44 292	1.0595	0.0414	18 750	1.0606	0.0442	3528 (42×21)	1.0817	0.0858	1.0581	0.0880
221 640	1.0441	0.0291	93 750	1.0456	0.0602	8712 (66×33)	1.0739	0.0459	1.0537	0.0509
			468 750	1.037	0.0309	18 432 (96×48)	1.0674	0.0519	1.0512	0.0501
						43 808 (148×74)	1.0619	0.0482	1.0476	0.0342
						93 312 (216×108)	1.0591	0.0427	1.0460	0.0323
						220 448 (332×166)	1.0557	0.0413	1.0438	0.0274
						468 512 (484×242)	1.0537	0.0242	1.0425	0.0276

performed the same computational experiment to quantify the isoperimetric property of a mesh based on a pinwheel-tiling.

For a **4k** mesh with the NP factor of 0.3, 12 **4k** mesh grids are utilized. For each grid, five randomly perturbed meshes are tested, and the averaged deviation ratio ($\text{dev}_\rho(\Omega_{\text{FE}})$) among the five meshes is taken. Then, the ρ-path deviation ratio of the **4k** mesh with the NP (Types III and IV) is compared with the other two mesh types that are based on a pinwheel-tiling (Types I and II). The ρ-path deviation ratio versus the number of elements is provided in Table II, and is plotted in Figure 14(a) on semi-logarithmic scale. The **4k** meshes with the NP and the ES (Type IV) perform better than the pinwheel-based meshes in representing a straight line with approximately up to approximately 0.1 million elements. In addition, in order to obtain the deviation ratio of 1.1, the required number of elements is approximately 100 for the **4k** meshes (Type IV), whereas it is 1000 for the pinwheel-based meshes (Types I and II), i.e. one order difference, in this example.

The Hausdorff distance between the shortest path in each mesh grid and the corresponding straight line is estimated based on the results above. For each **4k** mesh grid, the average Hausdorff

DOI: 10.1002/nme

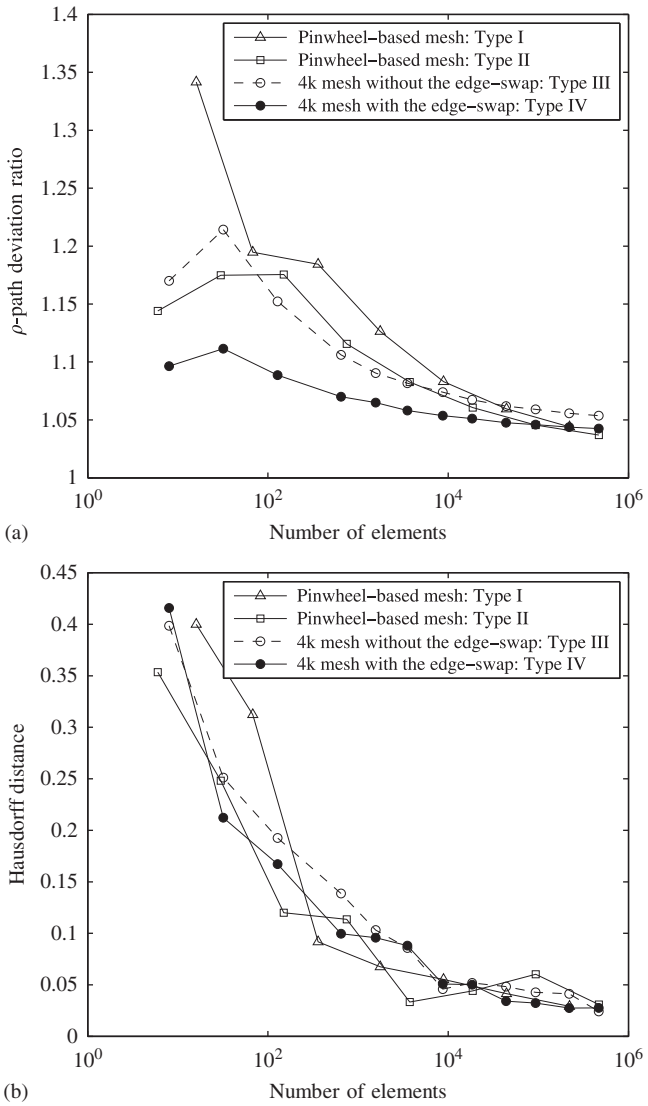


Figure 14. Quantification of the isoperimetric property in conjunction with (a) the ρ-path deviation ratio and (b) the Hausdorff distance.

distance of the shortest paths of the five perturbed meshes is taken. Then, it is compared with the Hausdorff distances of the shortest paths in the two other mesh types (Types I and II). The Hausdorff distance versus the number of elements is presented in Table II and plotted in Figure 14(b). A general decreasing trend is observed for the Hausdorff distance values as the number of elements increases for all the four mesh types.

The achieved results show that a **4k** mesh with NP and ES can perform either similarly or better than pinwheel-based mesh for some practical grid sizes. Although the pinwheel-based mesh

DOI: 10.1002/nme

has the isoperimetric property (in the limit sense), the exact path cannot be fully represented in numerical analysis (based on discretization). This is because the isoperimetric property assures spatial convergence only in the limit, and, in finite element analysis, the number of elements is finite.

6. TOPOLOGICAL DATA STRUCTURE

For an adaptive analysis environment, especially for the extrinsic cohesive zone model, classical finite element mesh representation, which consists of a node table and element incidence, is not adequate because of the limited topological information to handle adjacency relationship [8, 34]. In order to obtain efficient handling of adjacency relationship, the present paper utilizes a topology-based data structure, called *TopS* [4, 8]. TopS is able to obtain adjacency information and to insert cohesive surface elements with a linear time scale [35]. In order to maintain a consistent data structure when modification events occur, the client–server approach is employed.

6.1. Client–server approach

The topological data structure TopS is employed here for finite element mesh representation. It provides the geometric and topological support needed by numerical analysis applications, including adaptive simulations such as the insertion of a cohesive element and the ES operator. One of the main advantages of using TopS to support an analysis application consists of decoupling a computational mechanics code from data representation and management codes. As a consequence, developers of analysis applications can focus on the mechanics, relying on an external library for mesh representation.

In this way, TopS is seen as a *service provider*: it provides all the functionality needed by the analysis application for mesh representation and management. The analysis application represents the *client* that uses the services provided by TopS. The client application invokes TopS using the API. The TopS API consists of a large set of functions that the analysis code uses to set up the model, to attach the necessary attributes and to query/update the stored information. TopS eventually needs to invoke functions, named *callbacks*, of the client side in order to notify the occurrence of events related to mesh modification. Examples of API and callback functions are illustrated in Appendix.

In summary, TopS is responsible for managing the geometry and topology information. When the client application modifies the mesh, TopS ensures geometry and topology consistency. However, TopS has no knowledge regarding the properties attached to the model. The client application is responsible for maintaining the consistency of property information. The properties are represented by attributes attached to the topological entities. Whenever the mesh is modified, the client application has to update the attributes accordingly.

6.2. Topological operators

The NP and ES operator are implemented within the framework of TopS. The NP algorithm works as follows. First, the internal nodes of the mesh are traversed randomly. For each of them, one computes its minimum distance to the corresponding opposite edges in the incident elements. The node is then displaced by the computed distance multiplied by a fixed perturbation factor (i.e. NP factor), along a randomly chosen direction. After applying the NP, we use a simple Laplacian

DOI: 10.1002/nme

smoothing technique [36] in order to improve the overall mesh quality. It is done by interactively averaging the coordinates of internal nodes. A mesh quality metric (e.g. Lo's parameter [30]) is used to check whether minimum quality parameters are achieved and thus determine the convergence of the algorithm. At each iteration step, all the elements of the mesh are visited. For each element that does not meet the required quality standards, the positions of its nodes are displaced by the average of the distance vectors to the corresponding neighbors on the incident edges. The displacement is computed with a simple arithmetic mean scaled by a fixed relaxation parameter set (ϕ) in the interval [0, 1]. This is given by the following expression:

$$\mathbf{p}_0^{t+1} = \mathbf{p}_0^t + \frac{\phi}{n} \sum_{i=1}^{n} (\mathbf{p}_i^t - \mathbf{p}_0^t) \tag{5}$$

where $\mathbf{p}_0^t$ is the position of the current node at step t, $\mathbf{p}_i^t$ is the position of its corresponding ith neighbor node and n is the number of adjacent edges.

The ES algorithm works by simply removing the two triangles adjacent to a given edge and inserting two new other triangles with the appropriate nodal incidence. The callback mechanism of TopS is used to notify the application of the changed entities, such that the attached attributes can be correctly transferred.

7. EXTRINSIC COHESIVE MODEL APPROACH

In this section, numerical schemes of adaptive dynamic cohesive fracture simulation are presented. Next, the constitutive relationship across fracture surfaces is explained.

7.1. Numerical scheme

The weak form of the finite element formulation is obtained by the principle of virtual work. The summation of the virtual strain energy and the virtual kinetic energy is equal to the sum of the virtual work done by external traction ($\mathbf{T}_{\text{ext}}$) and by cohesive traction ($\mathbf{T}_{\text{coh}}$),

$$\int_{\Omega_0} (\delta\mathbf{E} : \mathbf{S} + \delta\mathbf{u} \cdot \rho_0 \ddot{\mathbf{u}})\, \mathrm{d}\Omega_0 = \int_{\Gamma_0} \delta\mathbf{u} \cdot \mathbf{T}_{\text{ext}}\, \mathrm{d}\Gamma_0 + \int_{\Gamma_0} \delta\boldsymbol{\Delta} \cdot \mathbf{T}_{\text{coh}}\, \mathrm{d}\Gamma_0 \tag{6}$$

where Ω_0 and Γ_0 are domain and boundary in the reference configuration, respectively. The second Piola–Kirchhoff stress ($\mathbf{S}$) and the Green deformation strain ($\mathbf{E}$) are utilized. The superposed dots in $\ddot{\mathbf{u}}$ denotes the second time derivatives, $\mathbf{u}$ is displacement vector, $\boldsymbol{\Delta}$ is displacement jump (or separation) across fracture surfaces and ρ_0 is the material density. The weak form is discretized on the basis of the initial configuration, i.e. the total Lagrangian formulation.

The time integration scheme is implemented by the central difference method, i.e. explicit method [37, 38]. The algorithm outline of the extrinsic cohesive zone model is described in Algorithm 1. First, the displacement ($\mathbf{u}$), velocity ($\dot{\mathbf{u}}$) and acceleration ($\ddot{\mathbf{u}}$) vectors are initialized, and the current displacement vector is obtained from the previous time step information. Then, one checks the insertion of cohesive elements based on an external criterion such as the maximum hoop stress [39], the minimum strain energy density [40] and the loss of material stability [19]. In the current study, cohesive elements are adaptively inserted when the averaged normal traction is greater than the cohesive strength ($\sigma_{\max}$).

DOI: 10.1002/nme

Algorithm 1 Explicit time integration for the extrinsic cohesive zone model.

Initialization: displacements ($\mathbf{u_0}$), velocity ($\dot{\mathbf{u}}_\mathbf{0}$), acceleration ($\ddot{\mathbf{u}}_\mathbf{0}$)
for $n=0$ to $n_{\max}$ **do**
  Update displacement: $\mathbf{u}_{n+1}=\mathbf{u}_n+\Delta t\dot{\mathbf{u}}_n+\Delta t^2/2\ddot{\mathbf{u}}_n$
  Check if cohesive elements need to be inserted
  Update acceleration: $\ddot{\mathbf{u}}_{n+1}=\mathbf{M}^{-1}(\mathbf{R}^{\text{ext}}_{n+1}+\mathbf{R}^{\text{coh}}_{n+1}-\mathbf{R}^{\text{int}}_{n+1})$
  Update velocity: $\dot{\mathbf{u}}_{n+1}=\dot{\mathbf{u}}_n+\Delta t/2(\ddot{\mathbf{u}}_n+\ddot{\mathbf{u}}_{n+1})$
  Update boundary conditions
end for

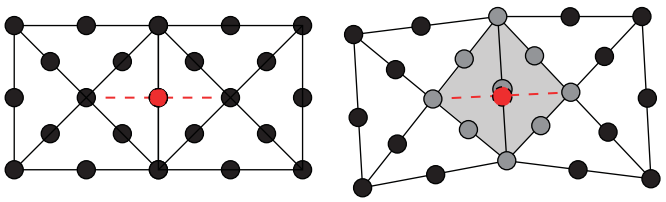


Figure 15. Location of the mid-point in the edge-swap operator: (a) **4k** structured mesh and (b) **4k** perturbed mesh.

The ES operator is adaptively requested during computational simulation when a cohesive element is necessary to be inserted along a swapped edge. When the ES is applied to a **4k** structured mesh, a mid-node of the swapped edge remains at the same location for six-node triangular elements (Figure 15(a)). When the ES operator is utilized in a perturbed mesh, the mid-node of the swapped edge is relocated so that the edge becomes a straight line (Figure 15(b)). Owing to the modification of the nodal coordinates, nodal quantities of the relocated node are updated through the interpolation of nodal quantities. The original **4k** path (e.g. shaded quadrilateral) is mapped into a 2×2 square domain (e.g. $\xi-\eta$ coordinates) by using the nine Lagrange shape functions with the nine gray solid nodes, i.e. $\mathbf{x}=\sum_{i=1}^{9}\mathbf{N}_i(\xi,\eta)\mathbf{x}_i$ (see Figure 15(b)). Then, the relocated nodal point in $\xi-\eta$ coordinates is calculated, and the nodal quantities, such as displacement, velocity and acceleration, are interpolated from the nine gray nodes by using the nine Lagrange shape functions.

After the insertion of cohesive elements, the external nodal force vector ($\mathbf{R}^{\text{ext}}_{n+1}$) and the internal nodal force vector ($\mathbf{R}^{\text{int}}_{n+1}$) are evaluated. The constitutive relationship of volumetric elements is assumed to be linear elastic. The cohesive force vector ($\mathbf{R}^{\text{coh}}_{n+1}$) is obtained by integrating the traction along cohesive surface elements. The traction is first set to be equivalent to the traction along the edge at the time of the cohesive element insertion. In the following time steps, the cohesive traction is evaluated on the basis of the potential-based constitutive model, which is explained in the following section. Alternatively, one can enforce $\mathbf{R}^{\text{coh}}_{n+1}$ at the time of activation so that the acceleration of a duplicated node is equal to an original node [24]. However, this approach can result in the violation of the conditions of $\text{sgn}(T_\text{n})=\text{sgn}(\Delta_\text{n})$ and $\text{sgn}(T_\text{t})=\text{sgn}(\Delta_\text{t})$, called traction locking [24]. The lumped mass matrix ($\mathbf{M}$) is obtained by considering diagonal terms of the consistent mass matrix and scaling them to preserve the total mass [41], which produces positive

DOI: 10.1002/nme

lumped masses. Then, the nodal acceleration, velocity and boundary conditions are updated for the current time step.

7.2. Constitutive relationship

For the constitutive relationships of cohesive elements, the unified potential-based model [9], called the PPR model, is implemented. The PPR model is able to represent different fracture energies (ϕ_n, ϕ_t) and cohesive strengths $(\sigma_{max}, \tau_{max})$ in each fracture mode. The potential for the extrinsic cohesive zone model is expressed as

$$\Psi(\Delta_n, \Delta_t) = \min(\phi_n, \phi_t) + \left[\Gamma_n\left(1-\frac{\Delta_n}{\delta_n}\right)^{\alpha} + \langle \phi_n - \phi_t\rangle\right]\left[\Gamma_t\left(1-\frac{|\Delta_t|}{\delta_t}\right)^{\beta} + \langle \phi_t - \phi_n\rangle\right] \tag{7}$$

where $\langle\cdot\rangle$ denotes the *Macaulay bracket*, and Δ_n, Δ_t are normal and tangential separations, respectively. The energy constants are expressed as

$$\Gamma_n = (-\phi_n)^{\langle \phi_n - \phi_t\rangle/(\phi_n-\phi_t)}, \quad \Gamma_t = (-\phi_t)^{\langle \phi_t - \phi_n\rangle/(\phi_t-\phi_n)} \quad (\phi_n \neq \phi_t) \tag{8}$$

for the different fracture energies. If the fracture energies are the same, one obtains the energy constants,

$$\Gamma_n = -\phi_n, \quad \Gamma_t = 1 \quad (\phi_n = \phi_t) \tag{9}$$

The shape parameters (α, β) are defined with respect to the shape of a softening curve. The PPR model can provide a convex softening $(\alpha, \beta \gg 2)$, a concave softening $(\alpha, \beta \ll 2)$ or a linear softening $(\alpha, \beta \approx 2)$.

The gradient of the potential leads to the normal and tangential tractions along the fracture surfaces,

$$\begin{aligned} T_n(\Delta_n, \Delta_t) &= -\alpha\frac{\Gamma_n}{\delta_n}\left(1-\frac{\Delta_n}{\delta_n}\right)^{\alpha-1}\left[\Gamma_t\left(1-\frac{|\Delta_t|}{\delta_t}\right)^{\beta} + \langle \phi_t - \phi_n\rangle\right] \\ T_t(\Delta_n, \Delta_t) &= -\beta\frac{\Gamma_t}{\delta_t}\left(1-\frac{|\Delta_t|}{\delta_t}\right)^{\beta-1}\left[\Gamma_n\left(1-\frac{\Delta_n}{\delta_n}\right)^{\alpha} + \langle \phi_n - \phi_t\rangle\right]\frac{\Delta_t}{|\Delta_t|} \end{aligned} \tag{10}$$

Notice that the value of tangential traction at $\Delta_t = 0$ exists in the limit sense [9], i.e.

$$\begin{aligned} \lim_{\Delta_t \to 0^+} T_t(\Delta_n, \Delta_t) &= -\beta\frac{\Gamma_t}{\delta_t}\left[\Gamma_n\left(1-\frac{\Delta_n}{\delta_n}\right)^{\alpha} + \langle \phi_n - \phi_t\rangle\right] \\ \lim_{\Delta_t \to 0^-} T_t(\Delta_n, \Delta_t) &= \beta\frac{\Gamma_t}{\delta_t}\left[\Gamma_n\left(1-\frac{\Delta_n}{\delta_n}\right)^{\alpha} + \langle \phi_n - \phi_t\rangle\right] \end{aligned} \tag{11}$$

which corresponds to a feature of the extrinsic cohesive zone models.

The above normal and tangential tractions are defined in a cohesive interaction (softening) region. The normal softening region is a rectangular domain associated with the normal final crack opening width (δ_n) and the conjugate tangential final crack opening width $(\bar{\delta}_t)$. Similarly, the tangential softening region is a rectangular domain associated with the conjugate normal final

DOI: 10.1002/nme

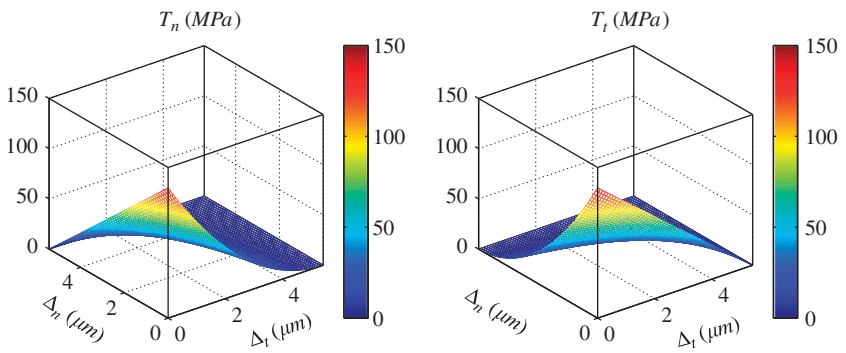


Figure 16. Gradient of the PPR potential with $\phi_n = \phi_t = 352\,\mathrm{N/m}$, $\sigma_{max} = \tau_{max} = 129.6\,\mathrm{MPa}$, $\alpha = \beta = 2$.

crack opening width ($\bar{\delta}_n$) and the tangential final crack opening width (δ_t). The final crack opening widths are expressed as

$$\delta_n = \alpha\phi_n/\sigma_{max}, \quad \delta_t = \beta\phi_t/\tau_{max} \tag{12}$$

which satisfy the boundary conditions of $T_n(\delta_n, \Delta_t) = 0$ and $T_t(\Delta_n, \delta_t) = 0$, respectively. The conjugate final crack opening widths ($\bar{\delta}_n, \bar{\delta}_t$) are given by

$$\bar{\delta}_n = \delta_n - \delta_n\left(\frac{\langle\phi_n - \phi_t\rangle}{\phi_n}\right)^{1/\alpha}, \quad \bar{\delta}_t = \delta_t - \delta_t\left(\frac{\langle\phi_t - \phi_n\rangle}{\phi_t}\right)^{1/\beta} \tag{13}$$

which satisfy the conditions of $T_t(\bar{\delta}_n, \Delta_t) = 0$ and $T_n(\Delta_n, \bar{\delta}_t) = 0$, respectively. Thus, when separation is outside of the normal softening region, the normal cohesive interaction is set to be zero. The normal cohesive interaction is continuous (i.e. no truncation) along the boundary of the normal softening region (i.e. $\Delta_n = \delta_n$ and $\Delta_t = \bar{\delta}_t$), because the normal cohesive interaction satisfies the boundary conditions of ($T_n(\delta_n, \Delta_t) = 0$ and $T_n(\Delta_n, \bar{\delta}_t) = 0$). Similarly, when separation is outside of the tangential softening region, T_t is set to be zero. The tangential cohesive interaction is continuous along the boundary of the tangential softening region (i.e. $\Delta_n = \bar{\delta}_n$ and $\Delta_t = \delta_t$), because the tangential cohesive interaction satisfies the boundary conditions of ($T_t(\bar{\delta}_n, \Delta_t) = 0$ and $T_t(\Delta_n, \delta_t) = 0$).

The gradient of the PPR potential is plotted in Figure 16. The mode I fracture energy and cohesive strength are selected as 352 N/m and 129.6 MPa, respectively. These values are utilized for dynamic fracture simulation of PMMA [1, 27]. The mode II fracture energy and cohesive strength are assumed to be the same as the mode I fracture parameters in this figure. The shape parameters (α, β) are selected as 2, which provides approximately linear softening in each fracture mode.

The unloading/reloading relationship is defined in conjunction with the softening behavior so that the model describes permanent dissipation [42]. The coupled unloading/reloading model is developed on the basis of a coupled loading history index,

$$\eta(\Delta_n, \Delta_t) = \sqrt{\Delta_n^2 + \Delta_t^2} \tag{14}$$

DOI: 10.1002/nme

Based on the loading history index (η), the unloading/reloading criterion is evaluated by introducing the maximum loading history index, defined as

$$\eta_{\max} = \sqrt{{\Delta_{n_{\max}}}^2 + {\Delta_{t_{\max}}}^2} \tag{15}$$

where $\Delta_{n_{\max}}$ and $\Delta_{t_{\max}}$ are the maximum normal and tangential separations in a loading (or separation) history, respectively. When the maximum loading history index ($\eta_{\max}$) is smaller than the (current) loading history index ($\eta > \eta_{\max}$), the current state of separations represents the loading condition, i.e. *softening* occurs. If $\eta_{\max}$ is greater than η ($\eta < \eta_{\max}$), material locally experiences the *unloading/reloading* condition.

Within the unloading/reloading region, the traction-separation relationships are expressed as

$$T_{\mathrm{n}}^{\upsilon}(\Delta_{\mathrm{n}}, \Delta_{\mathrm{t}}) = T_{\mathrm{n}}(\Delta_{\mathrm{n}}^{\upsilon}, \Delta_{\mathrm{t}}^{\upsilon}) \left(\frac{\eta}{\eta_{\max}}\right)^{\alpha_{\upsilon}}, \quad T_{\mathrm{t}}^{\upsilon}(\Delta_{\mathrm{n}}, \Delta_{\mathrm{t}}) = T_{\mathrm{t}}(\Delta_{\mathrm{n}}^{\upsilon}, \Delta_{\mathrm{t}}^{\upsilon}) \left(\frac{\eta}{\eta_{\max}}\right)^{\beta_{\upsilon}} \tag{16}$$

where

$$\Delta_{\mathrm{n}}^{\upsilon} = \Delta_{\mathrm{n}} \frac{\eta_{\max}}{\eta}, \quad \Delta_{\mathrm{t}}^{\upsilon} = \Delta_{\mathrm{t}} \frac{\eta_{\max}}{\eta} \tag{17}$$

The variables $\Delta_{\mathrm{n}}^{\upsilon}$ and $\Delta_{\mathrm{t}}^{\upsilon}$ satisfy the condition of $\sqrt{{\Delta_{\mathrm{n}}^{\upsilon}}^2 + {\Delta_{\mathrm{n}}^{\upsilon}}^2} = \eta_{\max}$, and the exponents α_{υ} and β_{υ} describe the shape of unloading/reloading surface along the radial direction. When an exponent value is equal to one, the unloading/reloading function is linear along the radial direction. If an exponent is greater or smaller than one, the shape is convex or concave, respectively. The normal and tangential unloading/reloading relations (e.g. black solids in Figure 17) are obtained by the multiplication $(\eta/\eta_{\max})^{\alpha_{\upsilon}}$ and $(\eta/\eta_{\max})^{\beta_{\upsilon}}$ of $T_{\mathrm{n}}(\Delta_{\mathrm{n}}^{\upsilon}, \Delta_{\mathrm{t}}^{\upsilon})$ and $T_{\mathrm{t}}(\Delta_{\mathrm{n}}^{\upsilon}, \Delta_{\mathrm{t}}^{\upsilon})$ (e.g. gray solids in Figure 17), which interpolates the normal and tangential tractions from the unloading/reloading boundaries to the origin, respectively. Owing to $0 \leqslant (\eta/\eta_{\max}) \leqslant 1$, the normal cohesive traction ($T_{\mathrm{n}}^{\upsilon}$) is monotonic and bounded between 0 and $T_{\mathrm{n}}(\Delta_{\mathrm{n}}^{\upsilon}, \Delta_{\mathrm{t}}^{\upsilon})$, and the tangential cohesive traction ($T_{\mathrm{t}}^{\upsilon}$) is also monotonic and bounded between 0 and $T_{\mathrm{t}}(\Delta_{\mathrm{n}}^{\upsilon}, \Delta_{\mathrm{t}}^{\upsilon})$ within the softening region.

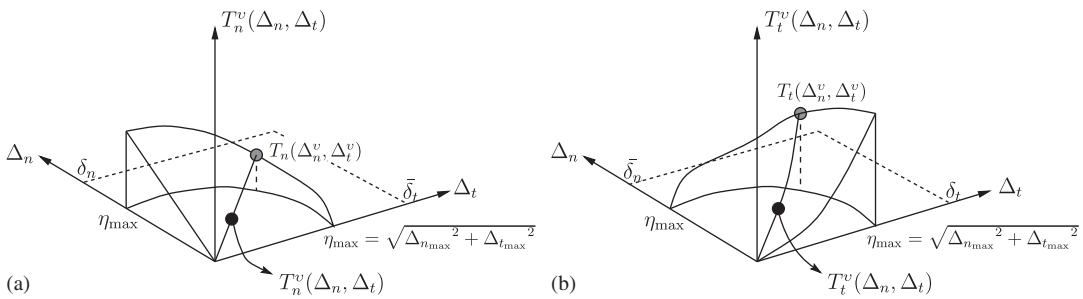


Figure 17. Schematics of the coupled unloading/reloading model: (a) normal interaction, and (b) tangential interaction.

DOI: 10.1002/nme

8. EXAMPLES

This section provides three examples: CCS tests, microbranching experiments and fragmentation simulations. The NP is applied to a finite element mesh before computational simulation, whereas the ES operators are adaptively requested during computation.

8.1. CCS test

The CCS test was developed by Rittel *et al.* [43] to investigate mixed-mode dynamic crack initiation in PMMA. The CCS is fractured by applying the impact with a Hopkinson bar. The geometry and the boundary conditions of the CCS are shown in Figure 18. The incident pulse is applied on the left side of the specimen by an incident bar whose diameter is 16.5 mm. The magnitude of the pulse is approximately 2.2 kN, and the pulse lasts about 350 μs. They observed that a crack initiated by forming angle at a kink of about 45°, and the dynamic stress intensity factor reached an experimental value of the fracture toughness at time $t = 96$–$126\,\mu$s. Previously, Papoulia *et al.* [5] utilized cohesive surface elements with reduced dimension to investigate convergence of a crack initiation angle. Menouillard *et al.* [44] simulated the CCS test with the Heaviside enrichment in conjunction with linear elastic fracture mechanics (LEFM) approach.

In the present computational simulation, plane strain condition is employed, and the impact pulse is applied as an acceleration on the traction boundary along the lower left segment 16.5 mm long. The elastic modulus of PMMA is 5.76 GPa, and the Poisson ratio is 0.42 [43]. The mode I fracture energy (ϕ_n) and the normal cohesive strength (σ_{max}) are 352.3 N/m and 129.6 MPa, respectively. In the absence of mode II fracture parameters, they are assumed to be the same as the mode I parameters. The shape parameters (α, β) are equal to two so that the softening model provides a linear softening relationship. Figure 19(a) illustrates the finite element mesh of the specimen. The mesh around the crack tip is generated by utilizing the curvilinear coordinate transformation and the conformal mapping [5] so that the mesh provides many alternative crack initiation directions. Figure 19(b) shows the zoom of the mesh around a crack tip, and Figure 19(c) demonstrates the perturbed mesh with the NP factor of 0.3. The number of nodes is 10 908 and the number of elements is 5345 at the initial discretization. A cohesive element is inserted at

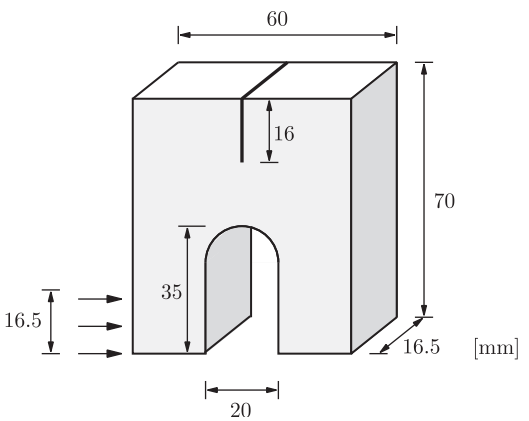


Figure 18. Geometry of compact compression specimen (CCS) tests and its boundary conditions.

DOI: 10.1002/nme

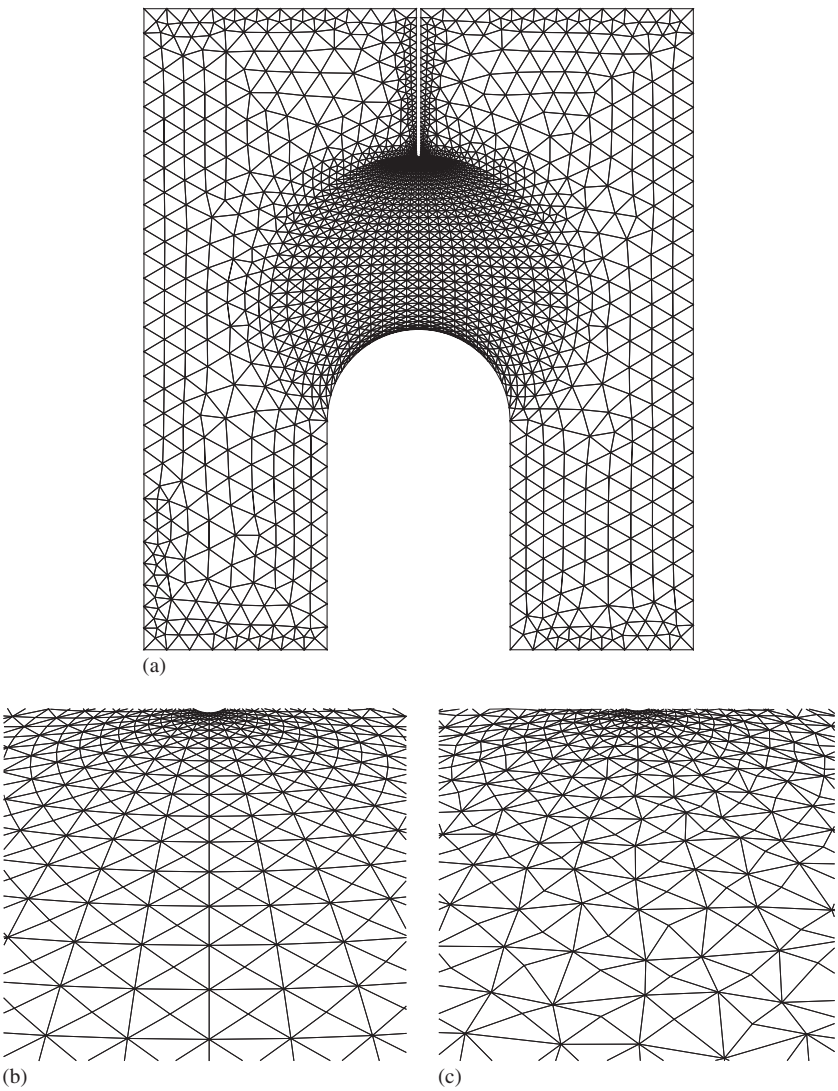


Figure 19. Finite element mesh of the compact compression specimen: (a) full specimen; (b) detail with NP = 0.0; and (c) detail with NP = 0.3.

80 μs, and the cohesive element experiences complete failure at 90 μs. Averaged crack velocity is approximately 110 m/s. The crack path of the computational simulation is illustrated in Figure 20. During simulation, 19 ES operations are requested, and the crack path is generally smooth because of the use of the ES operator. Additionally, the crack initiation angle is close to 45°, which corresponds to the experimental observation. After the initiation, the crack propagation direction changes along the vertical direction, which is similar to the previous computation by Menouillard *et al.* [44].

DOI: 10.1002/nme

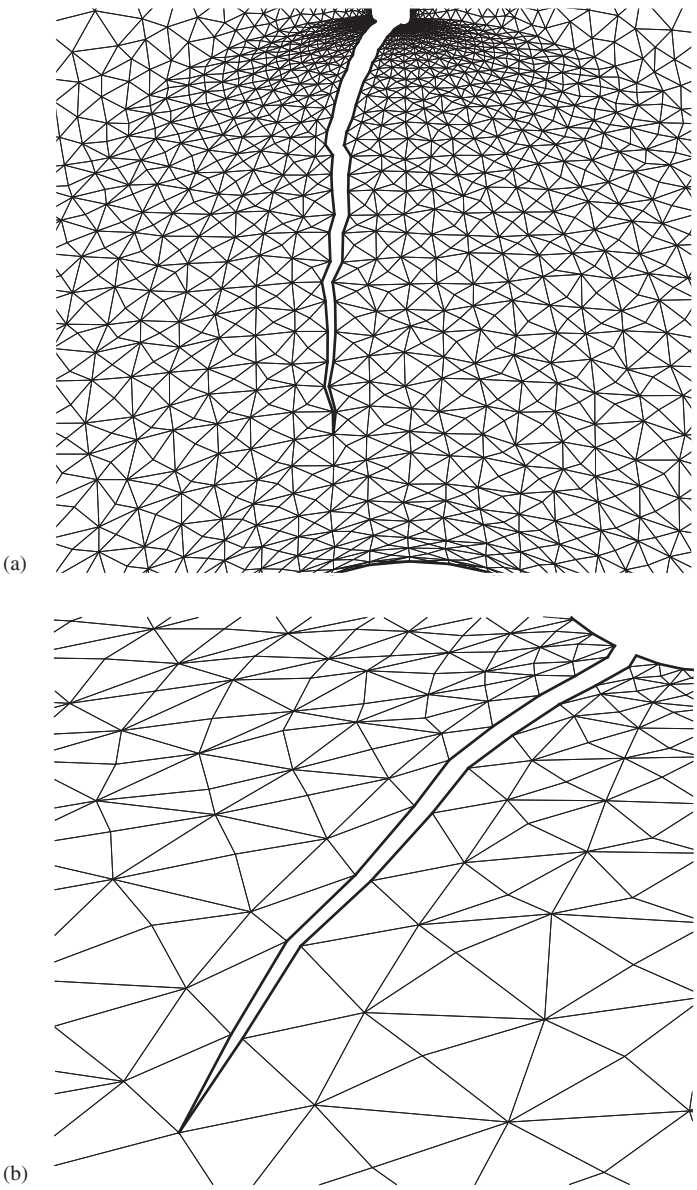


Figure 20. Crack path of CCS simulation results.

8.2. Microbranching experiments and simulations

Microbranching experiments were performed by Sharon *et al.* [45, 46] to investigate microbranching instability. Specimens are PMMA sheets having a thickness of either 0.8 or 3 mm, a width of 50–200 mm and a length of 200–400 mm. The initial stress of $\sigma_0 = 10$–18 MPa is applied by clamping the top and the bottom of a PMMA sheet. When a sharp crack is created by a razor

DOI: 10.1002/nme

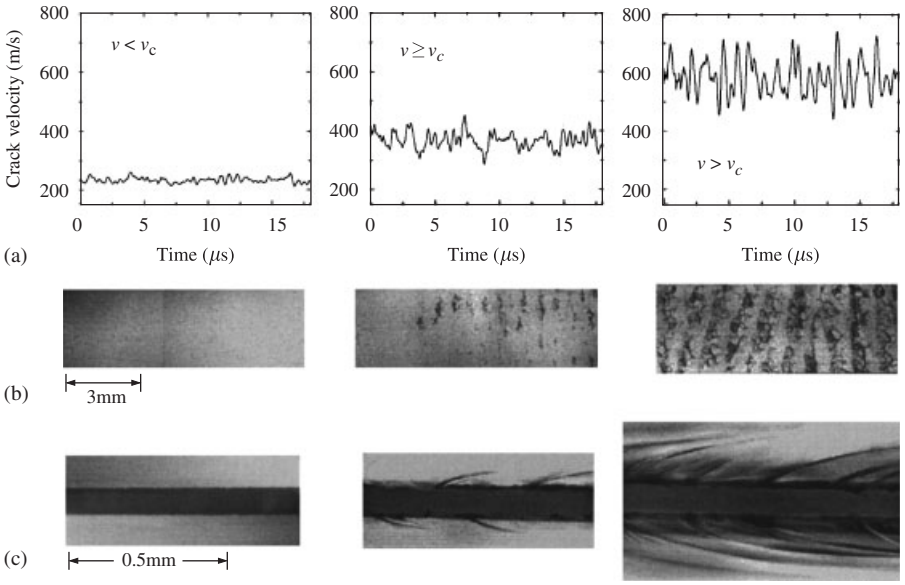


Figure 21. Observation of microbranching experiment: (a) crack velocity (v) versus time; (b) fractured surface; and (c) crack patterns with respect to different crack speeds ($v \sim 300\,\mathrm{m/s} < v_c$, $v \sim 400\,\mathrm{m/s} \sim v_c$, $v \sim 600\,\mathrm{m/s} > v_c$). The present figure is reproduced from Figure 4 of Sharon and Fineberg [46].

blade, a crack initiates and propagates. The higher energy input results in the higher crack velocity (v) and the more microbranching along the major crack, shown in Figure 21. The fracture surfaces are smooth when a crack velocity is lower than a critical velocity (v_c). When a crack velocity is greater than a critical value ($v > v_c$), one can observe that fracture surfaces are rough and more microbranching occurs. These physical phenomena are quantitatively investigated by Miller *et al.* [47] and Zhang *et al.* [27] in conjunction with cohesive surface elements. In addition, Ganguly [48] observed that unstructured mesh probably performs even better than the pinwheel mesh in replicating the branching pattern for high initial strain.

The geometry and boundary conditions for computational simulations are described in Figure 22. The domain size is reduced to 16 mm $\times$ 4 mm, and plane stress condition is employed with unit thickness. The 2D domain is initially discretized by the **4k** mesh grid of 192 $\times$ 48, and then the NP factor of 0.3 is applied. The ES operator is activated when it is necessary during computational simulation. The number of nodes is 74 257 and the number of elements is 36 864 at the initial discretization. The time step (Δt) is 0.002 µs, which is approximately one order lower than the characteristic time step [27, 49]. The initial strain ($\varepsilon_0 = 0.010$–0.015) is applied by imposing an initial displacement boundary condition within the domain. The material properties of PMMA are based on the properties listed in the reference [1]. The elastic modulus of PMMA is 3.24 GPa, the Poisson ratio is 0.35 and the density is $1190\,\mathrm{kg/m^3}$. The fracture parameters of PMMA were selected to be the same as those in the CCS simulation.

Figure 23 compares microbranching patterns with respect to different initial strains ($\varepsilon_0 = 0.01$, 0.012 and 0.015). The higher initial strain represents the more energy input, and thus one can expect

DOI: 10.1002/nme

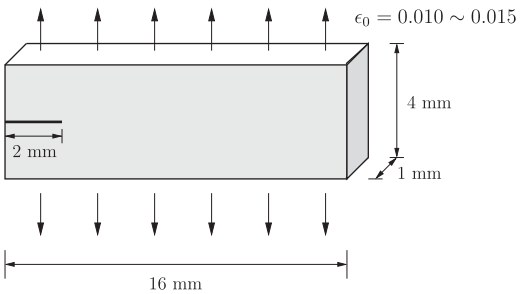


Figure 22. Schematics of geometry and boundary conditions for the microbranching experiments.

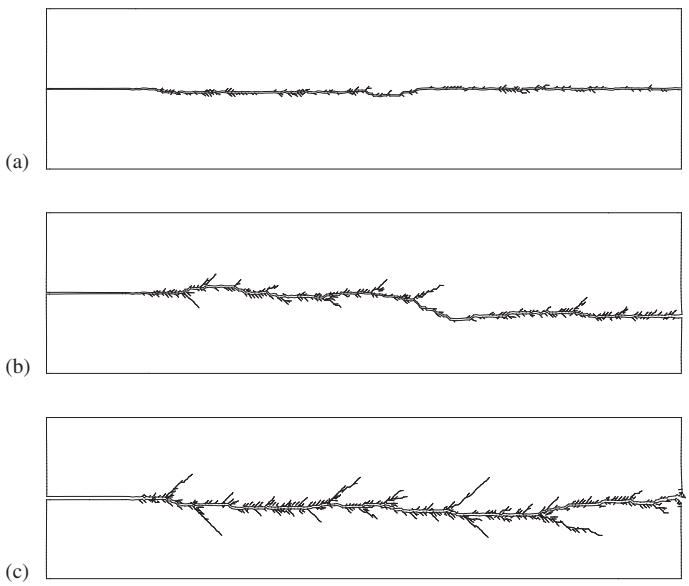


Figure 23. Branching patterns with respect to different initial strains: (a) $\varepsilon_0 = 0.01$; (b) $\varepsilon_0 = 0.012$; and (c) $\varepsilon_0 = 0.015$ for the **4k** mesh grid of 192×48.

more microbranching, which is captured in the experimental results. The simulation results also demonstrate that the overall length of microbranching increases with respect to the increase of the initial strain. The major crack propagates along the horizontal direction, although it demonstrates a little deviation from the center line for the initial strain of 0.012 (Figure 23(b)). The microbranching evolution with respect to time for the initial strain of 0.015 is illustrated in Figure 24. In addition, the crack velocity is estimated through the linear regression of time and crack tip position. Crack tip position is defined when the constitutive relationship of a cohesive element provides complete separation at all integration points. Figure 25 illustrates the crack velocity versus time with respect to the different initial strains. The higher initial strain leads to the higher averaged crack velocity and more fluctuation in the velocity, which qualitatively corresponds to experimental results.

DOI: 10.1002/nme

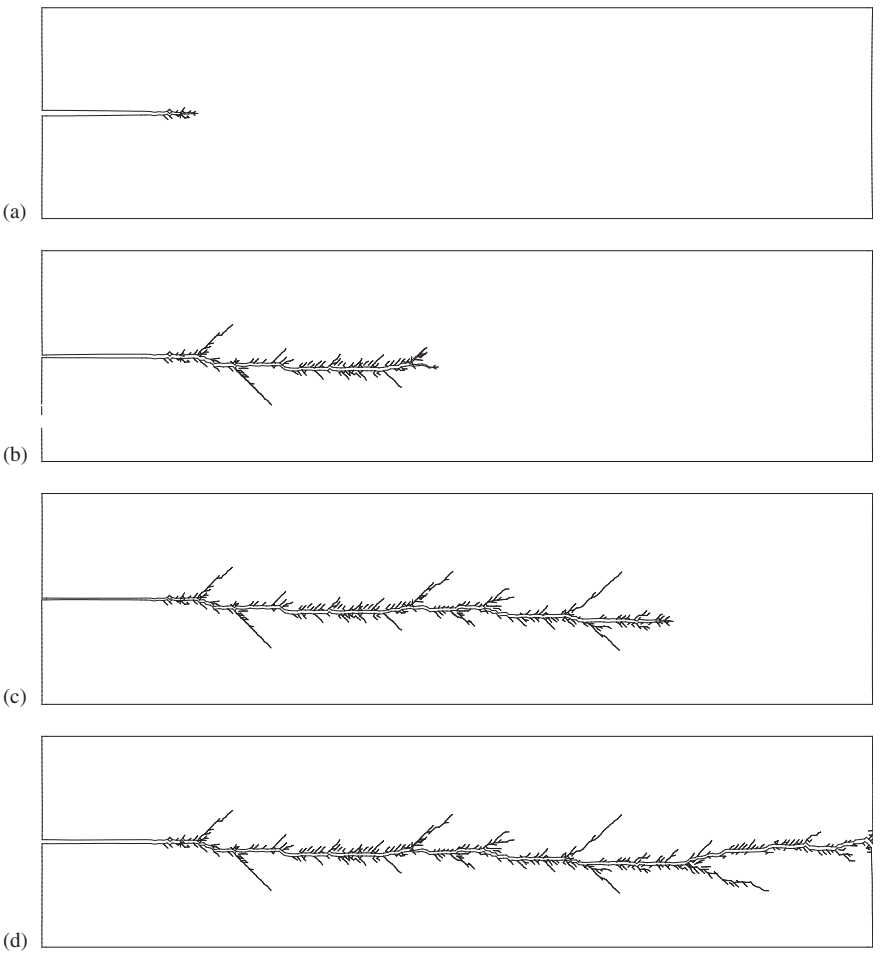


Figure 24. Microbranching evolution with respect to time for $\varepsilon_0 = 0.015$: (a) 2 μs; (b) 8 μs; (c) 14 μs; and (d) 20 μs.

Mesh refinement is performed with the **4k** mesh grid of 256×64, which leads to 131 777 nodes and 65 536 elements at the initial discretization. Figure 26 illustrates branching patterns with respect to different initial strains: (a) $\varepsilon_0 = 0.01$, (b) $\varepsilon_0 = 0.012$ and (c) $\varepsilon_0 = 0.015$. These microbranching patterns well correspond to the results with the **4k** mesh grid of 192×48. The crack velocities versus time and the averaged velocities are plotted in Figure 27 for each initial strain. The averaged velocities obtained from the 256×64 mesh grid are similar to the velocities obtained from the 192×48 mesh grid. Thus, the computational results are consistent under mesh refinement regarding crack patterns and velocities.

In order to investigate the effect of randomness and the ES operator on computational results, three finite element meshes are consecutively generated with the NP factor of 0.3, and all the nodal locations are stored. For each finite element mesh, the microbranching problem with the initial strain of 0.015 is simulated with the ES operator (Figure 28) and without the ES operator

DOI: 10.1002/nme

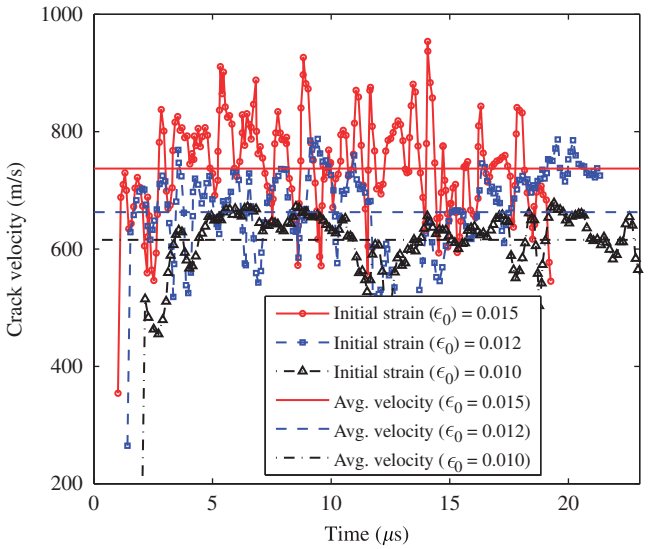


Figure 25. Crack velocity and averaged velocity versus time considering the **4k** mesh grid of 192×48 for applied initial strains (ε_0) of 0.01, 0.012 and 0.015.

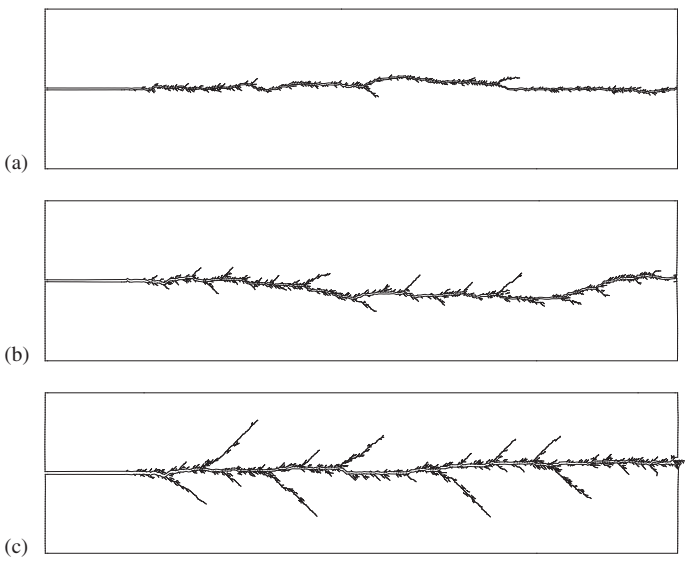


Figure 26. Branching patterns with respect to different initial strains for the mesh grid of 256×64: (a) $\varepsilon_0 = 0.01$; (b) $\varepsilon_0 = 0.012$; and (c) $\varepsilon_0 = 0.015$.

(Figure 29). Although the detailed microbranching patterns are different from each other because of the different discretizations, the overall crack patterns are all similar to each other. In addition, Figures 28 and 29 illustrate that one can obtain more consistent results with the ES operator than without the ES operator because the ES operator provides more potential crack propagation

DOI: 10.1002/nme

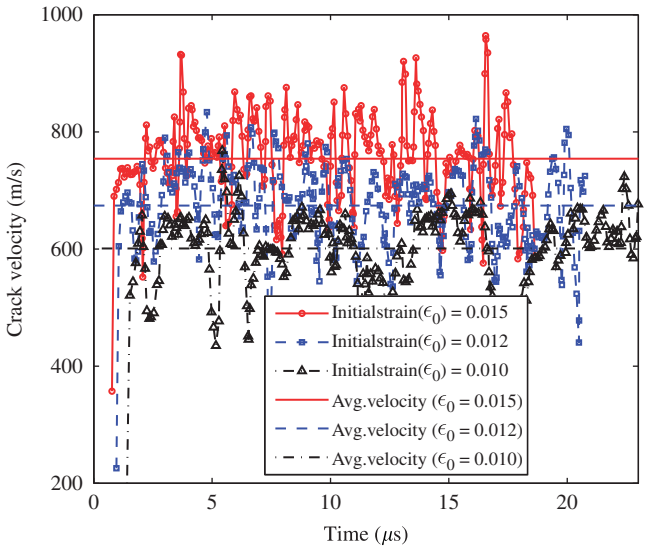


Figure 27. Crack velocity and averaged velocity versus time considering the **4k** mesh grid of 256×64 for applied initial strains (ε_0) of 0.010, 0.012 and 0.015.

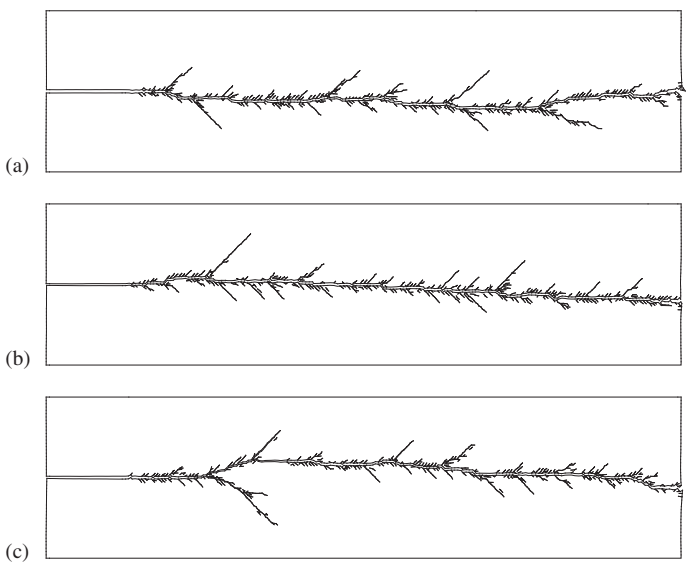


Figure 28. Three consecutive simulations for the applied initial strain of 0.015 with the edge-swap operator.

directions. Notice that 66, 70 and 70 ES operations are adaptively employed in Figures 28(a), (b) and (c), respectively. Moreover, the maximum deviation of the major crack from the straight center line is approximately 0.4 mm for the three results with the ES (Figure 28), whereas the maximum deviations are 0.4, 0.6 and 1.1 mm for the results without the ES (Figure 29).

DOI: 10.1002/nme

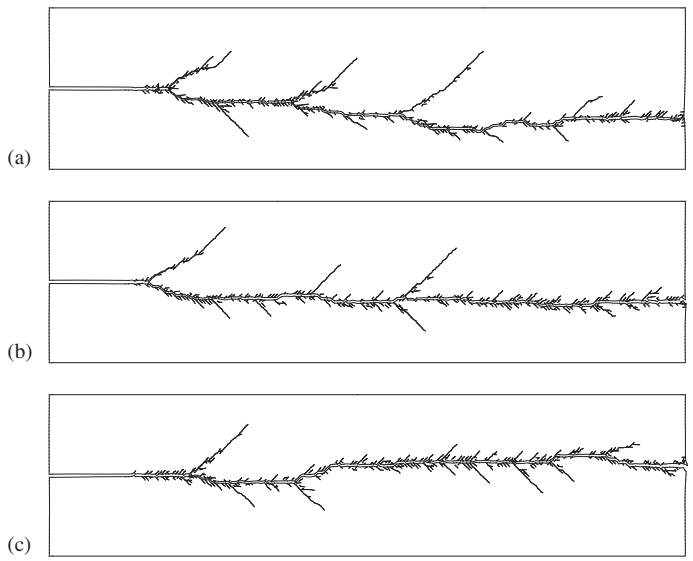


Figure 29. Three consecutive simulations for the applied initial strain of 0.015 without the edge-swap operator.

The energy balance is a necessary condition to ensure the numerical stability [37]. Figure 30 illustrates the energy conservation with respect to time for the initial strain of 0.015. The total energy (E_{tot}) consists of the external work (E_{ext}), strain energy (E_{int}), kinetic energy (E_{kin}) and work done by fracture (E_{fra}). In this simulation, the total energy is a constant, which corresponds to the initial strain energy, i.e. no external work ($E_{\text{ext}} = 0$). After a crack initiates and propagates, the strain energy decreases whereas the kinetic energy and the fracture energy increase. The computational result of energy evolution demonstrates the total energy conservation, as shown in Figure 30.

8.3. Fragmentation simulations

Fragmentation of a thick cylinder due to an impact load is investigated. The geometry of the cylinder is shown in Figure 31(a). The inner radius is 80 mm, whereas the outer radius is 150 mm. Impact pressure is applied along the inner rim, and the impact pressure with respect to time is shown in Figure 31(b). The elastic modulus is 210 GPa and the Poisson ratio is 0.3 with a density of 7850 kg/m^3. The fracture energy is 2000 N/m and the cohesive strength is 850 MPa. Previously, Rabczuk and Belytschko [50] investigated the fragmentation problem by using a mesh-free method with cracking particles. Zhou and Molinari [25] addressed mesh dependency in fragmentation problems, and Song and Belytschko [20] solved this problem by using the cracking node method.

The cylindrical domain is discretized by a **4k** mesh grid of 20×160 with a cylindrical-polar transformation. The number of nodes is 25 920 and the number of elements is 12 800 at the initial discretization. Three finite element meshes are generated with the NP factors of 0.3, and the ES operators are adaptively employed during computation. Crack nucleation in the middle of the domain is prevented. Fragmentation patterns of three consecutive computational results are demonstrated in Figure 32. The numbers of the ES operations used in Figures 32(a)–(c) are 116,

DOI: 10.1002/nme

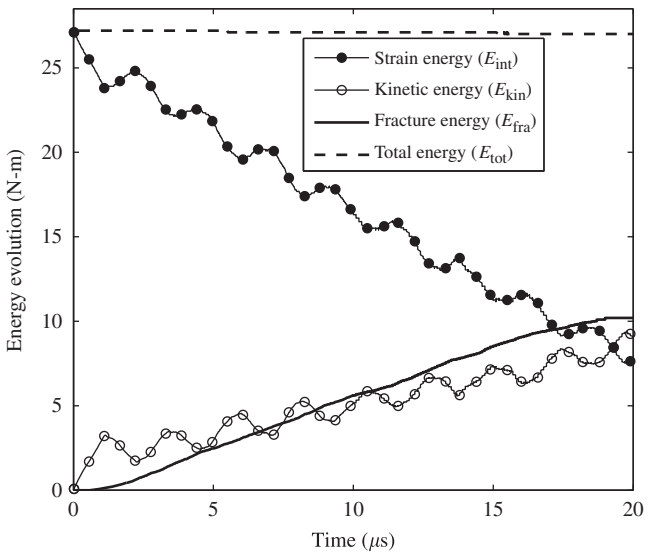


Figure 30. Energy evolution with respect to the time for $\varepsilon_0 = 0.015$.

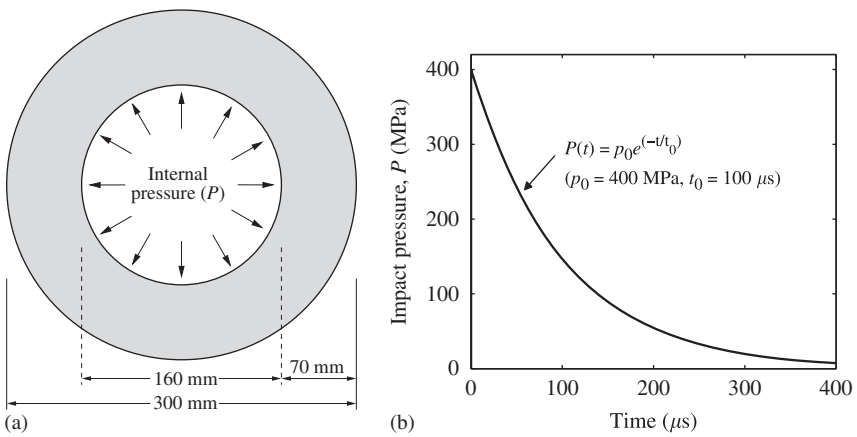


Figure 31. (a) Geometry of a thick cylinder and (b) applied impact pressure with respect to time.

139 and 149, respectively. The number of major fragments for each result is 24, 20 and 23, which correspond to the results by Song and Belytschko [20]. Figure 33 illustrates the fragmentation process with strain energy density at different times. Stress and strain are concentrated along the inner rim (Figure 33(a)) due to the internal pressure. Multiple cracks are initiated from the inner rim and propagate along the radial direction (Figures 33(b) and (c)). Some cracks are arrested during the fragmentation process, whereas the others propagate up to the outer rim with the formation of crack branching (Figures 33(d) and (e)). The number of major fragments is 24, as shown in Figure 33(f).

DOI: 10.1002/nme

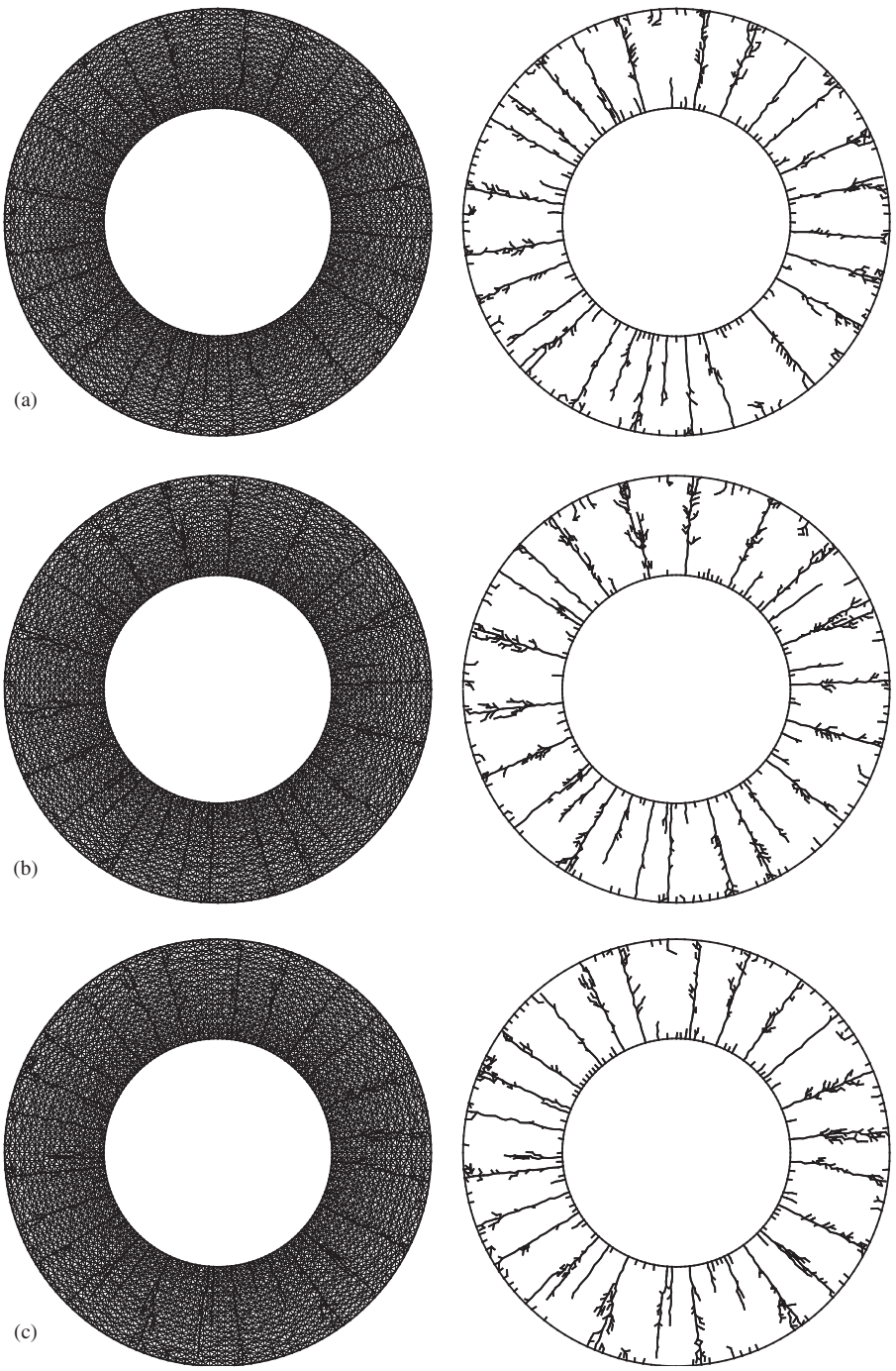


Figure 32. Fragmentation patterns of three consecutive computational results with the nodal perturbation factor of 0.3.

DOI: 10.1002/nme

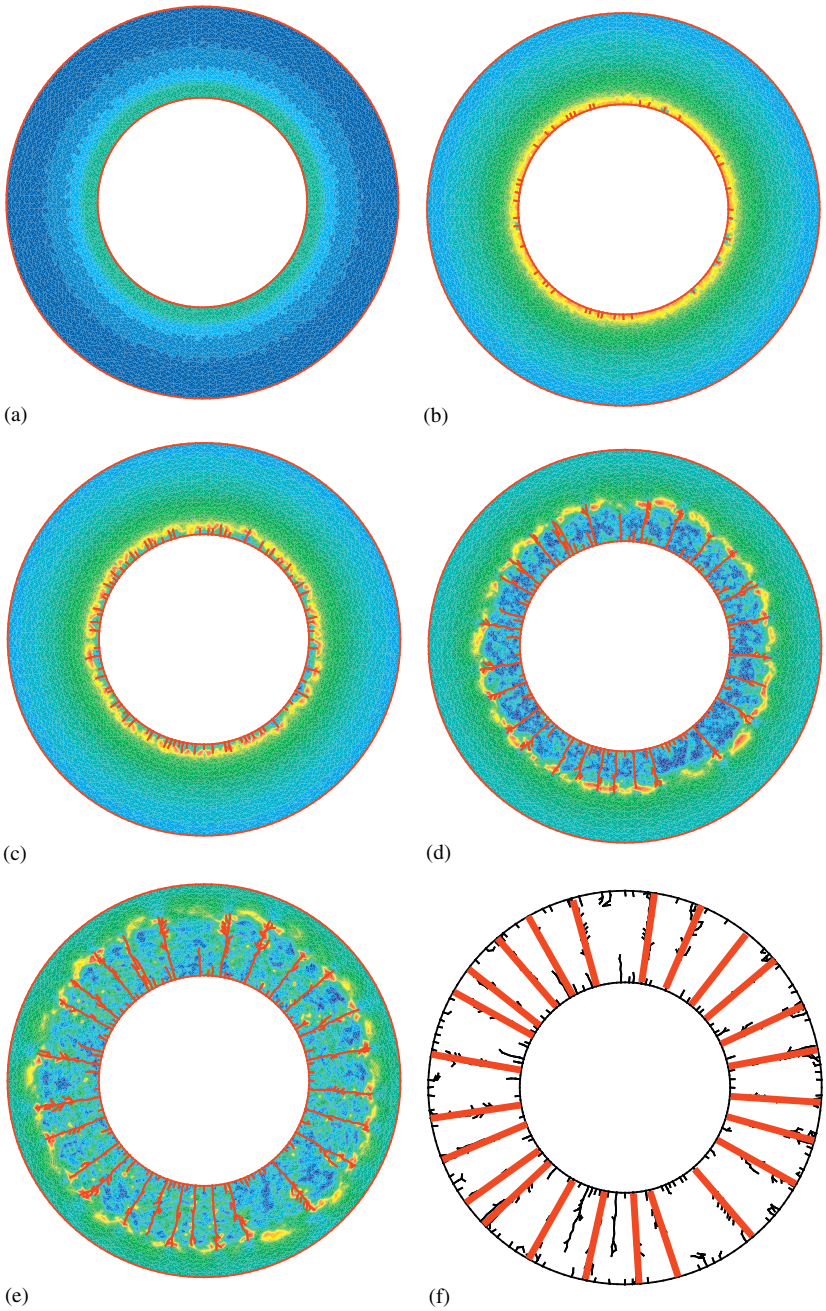


Figure 33. Fragmentation procedure with respect to time: (a) 26 μs; (b) 37 μs; (c) 39 μs; (d) 47 μs; (e) 54 μs; and (f) 64 μs. Part (f) displays 24 major fragments.

DOI: 10.1002/nme

9. SOME REMARKS ON **4k** AND PINWHEEL MESHES

A **4k** structured mesh is compared with a pinwheel-based mesh (i.e. 1 level). The pinwheel-based mesh consists of the pinwheel-tiling and triangles of the subdivision shown in Figure 34(a). The solid line in the pinwheel-based mesh is the original pinwheel-tiling, whereas the dashed line provides additional sub-triangles to eliminate hanging nodes in the pinwheel-tiling. Then, both the solid and the dashed lines are mapped into the **4k** structured mesh (Figure 34(b)). The edges of an individual pinwheel-tiling and its subdivision can be represented by a subset of the edges of a **4k** mesh, although a **4k** structured mesh has a different aspect ratio of triangles compared with a pinwheel-based mesh. However, notice that a set of pinwheel tiles (e.g. 2 levels) is not equivalent to a **4k** mesh, as illustrated in Figure 35. The topologies of the pinwheel tiles and the **4k** mesh are different along the line that has solid circles.

Reversely, edges of a **4k** mesh (Figure 36(a)) are mapped into edges of a pinwheel-based mesh (Figure 36(b)). In this case, one has to create additional edges (dashed lines) in a pinwheel-based mesh because a **4k** mesh has more edges (or directions) at a node than a pinwheel-based mesh, which may imply that **4k** meshes have richer topology. Notice that a finite element mesh whose internal nodes have more adjacent edges is always preferable for crack propagation simulation with cohesive surface element. As discussed previously, **4k** meshes with the ES operator provide eight

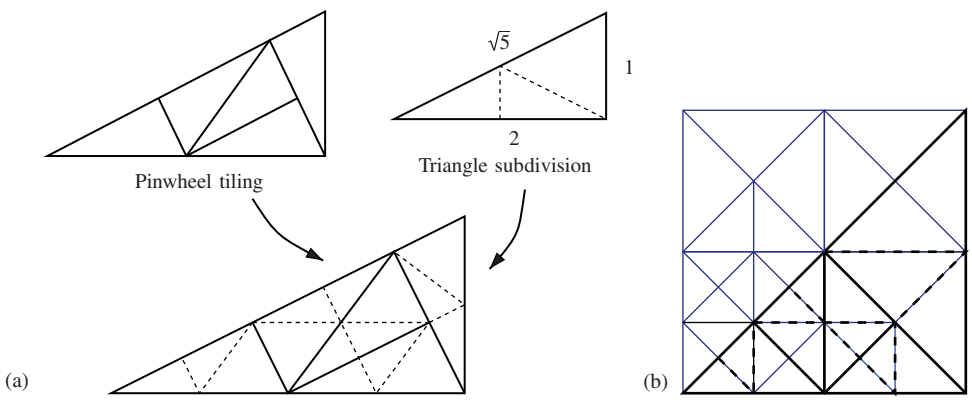


Figure 34. Comparison by mapping edges of pinwheel-based mesh into edges of **4k** structured mesh.

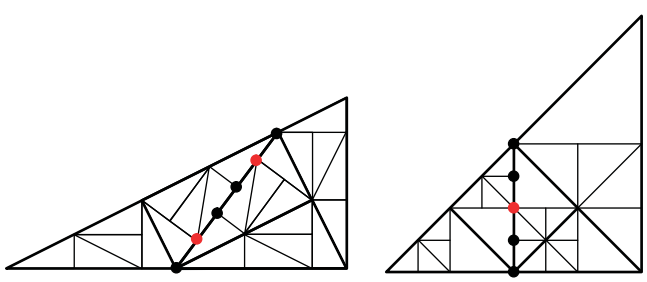


Figure 35. Comparison between a set of pinwheel tiles and a **4k** mesh.

DOI: 10.1002/nme

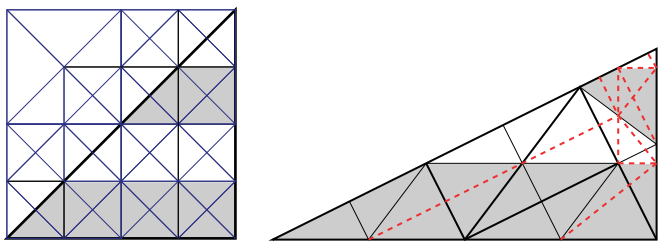


Figure 36. Comparison by mapping edges of **4k** structured mesh into edges of pinwheel-based mesh with additional dashed edges.

potential directions for all internal nodes. On the other hand, some internal nodes of pinwheel-based meshes have four, five or six directions, as shown in Figure 12(b). Therefore, **4k** meshes with the ES provide more (or the same) potential directions than pinwheel-based meshes for any internal node.

10. CONCLUSIONS

The potential-based PPR cohesive zone model with the topology-based data structure TopS leads to an effective computational framework to simulate physical phenomena associated with dynamic fracture, branching and fragmentation. The PPR model is implemented in conjunction with unloading/reloading relationships. The client–server architecture in TopS enables us to separate a computational mechanics code from a data representation and management code, and vice-versa. CCS tests, microbranching experiments and fragmentation problems are investigated by using the NP and the ES operator, which reduce mesh orientation dependence in **4k** structured meshes. Moreover, both the NP and the ES operator reduce discrepancy between mathematical and discretized paths more rapidly than the pinwheel-based meshes, for practical levels of mesh refinement. Simulation of the CCS test illustrates a smoother crack pattern with the ES operator. In the microbranching experiments, computational simulation provides consistent results in terms of microbranching patterns and crack velocity. The use of the ES operator reduces the maximum deviation of the major crack from a straight center line. In addition, the total energy is conserved during adaptive insertion of cohesive elements using ES operators. Finally, fragmentation of a thick cylinder is investigated. The fragmentation patterns and the number of fragments obtained from this study illustrate agreement with the previous computational results.

APPENDIX A: API AND CALLBACK FUNCTIONS

The client application creates a model and then modifies it by calling the functions provided by the TopS API. The TopS API is composed by a large set of functions that allow the client to create, destroy or modify the model and its topological entities. Whenever creating, destroying or modifying a mesh, the client application has to consistently manage the attributes attached to the affected entities. The TopS API provides functions that directly modify the mesh. When using

DOI: 10.1002/nme

these functions, the client can update the attributes associated with the affected topological entities directly, just after invoking the function. For example, let us consider a function to insert a new node in the model. Let us also consider that a node attribute (e.g. `MyNodeAttrib type`) has to be associated with each node of the model. Then, after inserting a new node, the client application is responsible for updating the node attribute accordingly. A typical excerpt of the client code in such a situation is (using the C API):

```
TopNode node;
MyNodeAttrib nAtt;
. . .
node = topModel_InsertNode (x , y , z );
nAtt = (MyNodeAttrib*) malloc (sizeof (MyNodeAttrib));
topNode_SetAttrib (model , node , nAtt);
. . .
```

In this case, managing the attributes is simple because the `topModel_InsertNode` function modifies a single entity in the mesh, the newly created node, and returns a reference to it. Hence, the client gets the returned reference and attaches the corresponding attribute.

However, in a few situations, the attribute managing cannot be done so straightforwardly because some API functions, when invoked, affect a variable number of topological entities. Let us consider the function that inserts a cohesive element in the mesh. The insertion of a cohesive element may require the duplication of nodes. Whether a node has to be duplicated or not depends on the topological classification of the fractured facet. The client application cannot know in advance how many and which nodes will be duplicated. In such cases, the callback mechanism is very handy, for instance, while executing the function that inserts a new cohesive element, whenever a node is duplicated, TopS calls back a function in the client application side. This client function, named callback function, is responsible for managing the attributes associated with the affected entities (in this case, the old and the newly created nodes).

APPENDIX B: REGISTERING AND CODING CALLBACK FUNCTIONS

A few API functions can lead to mesh modification events when they are executed. For instance, the function to insert a cohesive element (`InsertCohesive`) may provide the duplicate node event. The ES operator (`SwapEdge4K8`) results in a swap element and a swap edge. In order to be notified of the occurrence of the modification, the client application has to *register* the corresponding callback function. The client application may not be interested in all generated events, but it has to register callbacks for the events it needs to track, that is, for the events that affect entities that hold attributes. Note that the client application registers each callback once, and whenever the function that generated the corresponding event is invoked, more than one event may occur. For each event, the callback function is invoked by TopS.

To register the callback for the duplicate node, the swap element and the swap edge, the client application uses the following TopS API functions:

```
void topModel_SetDuplicateNodeCb (TopModel* m, TopDuplicateNodeCb cb, void* data);
void topModel_SetSwapElemCb(TopModel* m, TopSwapElemCb cb, void* data);
void topModel_SetSwapEdgeCb(TopModel* m, TopSwapEdgeCb cb, void* data);
```


DOI: 10.1002/nme

Besides the model, the function receives as input a pointer to the callback function and an extra pointer to data that will be passed as parameter to the callback function. If no extra data are needed, the client may pass the `NULL` value as this last parameter.

The callback function `cb` has to be a function with the following prototypes:

```
void DuplicateNodeCb
   (TopModel* m, TopNode old_n, TopNode new_n, void* data);
void TopSwapElemCb
   (TopModel* m, TopElement old_el[2], TopElement new_el[2], void* data);
void TopSwapEdgeCb
   (TopModel* m, TopEdge old_e, TopEdge new_e, void* data);
```

The `DuplicateNodeCb` receives as parameters the model, a reference to the node that was duplicated, a reference to the node that was just created and a pointer to the extra data passed when the callback was registered. The `TopSwapElemCb` receives as parameters the model, an array with the two old elements, an array with the two newly created elements (that replace the old ones) and a pointer to the extra data passed when the callback was registered. The `TopSwapEdgeCb` receives as parameters the model, the old edge, the newly created edge (that replaces the old one) and a pointer to the extra data passed when the callback was registered.

APPENDIX C: EXAMPLE

Let us consider an illustrative example on the use of callback functions. Let us also consider a client application that attaches attributes only to nodes and elements, using the user-defined types `MyNodeAttrib` and `MyElemAttrib`, respectively. If the client application invokes the `SwapEdge4k8` function, a callback function to deal with swap element events must be registered. Hence, the client programmer has to code the corresponding callback:

```
// This call back function simply copies the attributes attached to
// the old elements to the new elements

static void MySwapCb (TopModel* m, TopElement old_el [2],
                      TopElement new_el [2] , void* data)

// Retrieve the attributes attached to the old elements
MyElemeAttrib* old_att0 = topElement_GetAttrib (m, old_el[0]);
MyElemeAttrib* old_att1 = topElement_GetAttrib (m, old_el[1]);
// Create new attributes to the new elements
MyElemAttrib* att0 = (MyElemAttrib*)malloc(sizeof(MyElemAttrib));
MyElemAttrib* att1 = (MyElemAttrib*)malloc(sizeof(MyElemAttrib));
// Set the attribute to the new elements
topElement_SetAttrib (m, new_el[0], att0);
topElement_SetAttrib (m, new_el[1], att1);
// Copy the attribute from the first old element
memcpy (att0, old_att0, sizeof(MyElemAttrib));
memcpy (att1, old_att1, sizeof(MyElemAttrib));
```


DOI: 10.1002/nme

```
// Modify the attributes, if needed.
// The client can retrieve all adjacency relationships related to
// the newly created elements, but cannot retrieve the ones related to
// the old elements (the old elements no longer belong to the model)
. . .
// Free the old attributes
free (old_att0);
free (old_att1);
```

The client has to register the callback once when initializing TopS:

```
void topModel_SetSwapEdgeCb (model, MySwapCb, NULL);
```

Then, whenever the client application invokes the `topModel_SwapEdge4K8` function, the coded callback will be executed each time the swap element event occurs:

```
topModel_SwapEdge4K8 (model, edge);
```

REFERENCES

1. Xu XP, Needleman A. Numerical simulations of fast crack growth in brittle solids. *Journal of the Mechanics and Physics of Solids* 1994; **42**(9):1397–1434.
2. Camacho GT, Ortiz M. Computational modelling of impact damage in brittle materials. *International Journal of Solids and Structures* 1996; **33**(20–22):2899–2938.
3. Velho L, Gomes J. Variable resolution 4-k meshes: concepts and applications. *Computer Graphics Forum* 2000; **19**(4):195–212.
4. Celes W, Paulino GH, Espinha R. Efficient handling of implicit entities in reduced mesh representations. *Journal of Computing and Information Science in Engineering* 2005; **5**(4):348–359.
5. Papoulia KD, Vavasis SA, Ganguly P. Spatial convergence of crack nucleation using a cohesive finite-element model on a pinwheel-based mesh. *International Journal for Numerical Methods in Engineering* 2006; **67**(1):1–16.
6. Ganguly P, Vavasis SA, Papoulia KD. An algorithm for two-dimensional mesh generation based on the pinwheel tiling. *SIAM Journal on Scientific Computing* 2006; **28**(4):1533–1562.
7. Huttenlocher DP, Klanderman GA, Rucklidge WJ. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1993; **15**(9):850–863.
8. Celes W, Paulino GH, Espinha R. A compact adjacency-based topological data structure for finite element mesh representation. *International Journal for Numerical Methods in Engineering* 2005; **64**(11):1529–1556.
9. Park K, Paulino GH, Roesler JR. A unified potential-based cohesive model of mixed-mode fracture. *Journal of the Mechanics and Physics of Solids* 2009; **57**(6):891–908.
10. Swenson DV, Ingraffea AR. Modeling mixed-mode dynamic crack propagation using finite elements: theory and applications. *Computational Mechanics* 1988; **3**(6):381–397.
11. Abraham FF, Brodbeck D, Rafey RA, Rudge WE. Instability dynamics of fracture: a computer simulation investigation. *Physical Review Letters* 1994; **73**(2):272–275.
12. Klein PA, Foulk JW, Chen EP, Wimmer SA, Gao HJ. Physics-based modeling of brittle fracture: cohesive formulations and the application of meshfree methods. *Theoretical and Applied Fracture Mechanics* 2001; **37**(1–3):99–166.
13. Gao H, Klein P. Numerical simulation of crack growth in an isotropic solid with randomized internal cohesive bonds. *Journal of the Mechanics and Physics of Solids* 1998; **46**(2):187–218.
14. Moes N, Belytschko T. Extended finite element method for cohesive crack growth. *Engineering Fracture Mechanics* 2002; **69**(7):813–833.
15. Wells GN, Sluys LJ. A new method for modelling cohesive cracks using finite elements. *International Journal for Numerical Methods in Engineering* 2001; **50**(12):2667–2682.
16. Duarte CA, Reno LG, Simone A. A high-order generalized FEM for through-the-thickness branched cracks. *International Journal for Numerical Methods in Engineering* 2007; **72**(3):325–351.
17. Park K, Pereira JP, Duarte CA, Paulino GH. Integration of singular enrichment functions in the generalized/extended finite element method for three-dimensional problems. *International Journal for Numerical Methods in Engineering* 2009; **78**(10):1220–1257.

DOI: 10.1002/nme

18. Bishop JE. Simulating the pervasive fracture of materials and structures using randomly close packed Voronoi tessellations. *Computational Mechanics* 2009; **44**(4):455–471.
19. Belytschko T, Chen H, Xu J, Zi G. Dynamic crack propagation based on loss of hyperbolicity and a new discontinuous enrichment. *International Journal for Numerical Methods in Engineering* 2003; **58**(12):1873–1905.
20. Song JH, Belytschko T. Cracking node method for dynamic fracture with finite elements. *International Journal for Numerical Methods in Engineering* 2009; **77**(3):360–385.
21. 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.
22. Linder C, Armero F. Finite elements with embedded branching. *Finite Elements in Analysis and Design* 2009; **45**(4):280–293.
23. 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.
24. Papoulia KD, Sam CH, Vavasis SA. Time continuity in cohesive finite element modeling. *International Journal for Numerical Methods in Engineering* 2003; **58**(5):679–701.
25. Zhou F, Molinari JF. Dynamic crack propagation with cohesive elements: a methodology to address mesh dependency. *International Journal for Numerical Methods in Engineering* 2004; **59**(1):1–24.
26. Molinari JF, Gazonas G, Raghupathy R, Rusinek A, Zhou F. The cohesive element approach to dynamic fragmentation: the question of energy convergence. *International Journal for Numerical Methods in Engineering* 2007; **69**(3):484–503.
27. Zhang Z, Paulino GH, Celes W. Extrinsic cohesive modelling of dynamic fracture and microbranching instability in brittle materials. *International Journal for Numerical Methods in Engineering* 2007; **72**(8):893–923.
28. Ruiz G, Pandolfi A, Ortiz M. Three-dimensional cohesive modeling of dynamic mixed-mode fracture. *International Journal for Numerical Methods in Engineering* 2001; **52**(1–2):97–120.
29. Zhang Z. Extrinsic cohesive modeling of dynamic fracture and microbranching instability using a topological data structure. *Ph.D. Thesis*, University of Illinois, Urbana-Champaign, 2007.
30. Lo SH. Generating quadrilateral elements on plane and over curved surfaces. *Computers and Structures* 1989; **31**(3):421–426.
31. Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik* 1959; **1**:269–271.
32. Guibas LJ, Knuth DE, Sharir M. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica* 1992; **7**(4):381–413.
33. Radin C, Sadun L. The isoperimetric problem for pinwheel tilings. *Communications in Mathematical Physics* 1996; **177**(1):255–263.
34. Beall MW, Shephard MS. General topology-based mesh data structure. *International Journal for Numerical Methods in Engineering* 1997; **40**(9):1573–1596.
35. Paulino GH, Celes W, Espinha R, Zhang ZJ. A general topology-based framework for adaptive insertion of cohesive elements in finite element meshes. *Engineering with Computers* 2008; **24**(1):59–78.
36. Field DA. Laplacian smoothing and Delaunay triangulations. *Communications in Applied Numerical Methods* 1988; **4**(6):709–712.
37. Belytschko T, Liu WK, Moran B. *Nonlinear Finite Elements for Continua and Structures*. Wiley: New York, 2000.
38. Zhang Z, Paulino GH. Cohesive zone modeling of dynamic failure in homogeneous and functionally graded materials. *International Journal of Plasticity* 2005; **21**(6):1195–1254.
39. Ergodan F, Sih GC. On the crack extension in plates under plane loading and transverse shear. *Journal of Basic Engineering* 1963; **85**:519–527.
40. Sih GC. Strain–energy–density factor applied to mixed mode crack problems. *International Journal of Fracture* 1974; **10**(3):305–321.
41. Hughes TJR. *The Finite Element Method*: *Linear Static and Dynamic Finite Element Analysis*. Dover Publications: New York, 2000.
42. Park K. Potential-based fracture mechanics using cohesive zone and virtual internal bond modeling. *Ph.D. Thesis*, University of Illinois, Urbana-Champaign, 2009.
43. Rittel D, Maigre H. An investigation of dynamic crack initiation in PMMA. *Mechanics of Materials* 1996; **23**(3):229–239.
44. Menouillard T, Rethore J, Moes N, Combescure A, Bung H. Mass lumping strategies for X-FEM explicit dynamics: application to crack propagation. *International Journal for Numerical Methods in Engineering* 2008; **74**(3):447–474.

DOI: 10.1002/nme

45. Sharon E, Gross SP, Fineberg J. Local crack branching as a mechanism for instability in dynamic fracture. *Physical Review Letters* 1995; **74**(25):5096–5099.
46. Sharon E, Fineberg J. Microbranching instability and the dynamic fracture of brittle materials. *Physical Review B* (*Condensed Matter*) 1996; **54**(10):7128–7139.
47. Miller O, Freund LB, Needleman A. Energy dissipation in dynamic fracture of brittle materials. *Modelling and Simulation in Materials Science and Engineering* 1999; **7**(4):573–586.
48. Ganguly P. Spatial convergence of finite element cohesive interface networks. *Ph.D. Thesis*, Cornell University, 2006.
49. Pandolfi A, Krysl P, Ortiz M. Finite element simulation of ring expansion and fragmentation: the capturing of length and time scales through cohesive models of fracture. *International Journal of Fracture* 1999; **95**(1–4):279–297.
50. Rabczuk T, Belytschko T. Cracking particles: a simplified meshfree method for arbitrary evolving cracks. *International Journal for Numerical Methods in Engineering* 2004; **61**(13):2316–2343.

DOI: 10.1002/nme