INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2012; **92**:1–35
Published online 17 August 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.3163

Adaptive mesh refinement and coarsening for cohesive zone modeling of dynamic fracture

Kyoungsoo Park[1,2], Glaucio H. Paulino[1,*,†], Waldemar Celes[3] and Rodrigo Espinha[3]

[1]*Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Ave., Urbana, IL 61801, U.S.A.*
[2]*School of Civil and Environmental Engineering, Yonsei University, 134 Sinchon-dong, Seodaemun-gu, Seoul, Korea*
[3]*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

Adaptive mesh refinement and coarsening schemes are proposed for efficient computational simulation of dynamic cohesive fracture. The adaptive mesh refinement consists of a sequence of edge-split operators, whereas the adaptive mesh coarsening is based on a sequence of vertex-removal (or edge-collapse) operators. Nodal perturbation and edge-swap operators are also employed around the crack tip region to improve crack geometry representation, and cohesive surface elements are adaptively inserted whenever and wherever they are needed by means of an extrinsic cohesive zone model approach. Such adaptive mesh modification events are maintained in conjunction with a topological data structure (TopS). The so-called PPR potential-based cohesive model (*J. Mech. Phys. Solids* 2009; **57**:891–908) is utilized for the constitutive relationship of the cohesive zone model. The examples investigated include mode I fracture, mixed-mode fracture and crack branching problems. The computational results using mesh adaptivity (refinement and coarsening) are consistent with the results using uniform mesh refinement. The present approach significantly reduces computational cost while exhibiting a multiscale effect that captures both global macro-crack and local micro-cracks. Copyright © 2012 John Wiley & Sons, Ltd.

Received 9 October 2009; Revised 21 November 2010; Accepted 19 January 2011

KEY WORDS: adaptive mesh refinement (AMR); adaptive mesh coarsening (AMC); adaptive mesh refinement and coarsening (AMR&C); PPR potential-based model; topological data structure (TopS); dynamic cohesive fracture; extrinsic cohesive zone model

1. INTRODUCTION

Adaptive mesh refinement (AMR) schemes have been utilized to control and minimize error in the finite element analysis [1–4]. Researchers further applied mesh adaptation schemes to several engineering problems while reducing computational cost. For linear elastic problems, mesh around a crack tip region can be regenerated by a recursive spatial decomposition algorithm [5–7] when the crack propagates. For elasto-dynamic problems, Wiberg *et al.* [8] discussed error estimation and adaptivity (h-adaptivity, p-adaptivity) of spatial and time discretization. Belytschko and Tabbara [9] adaptively refined finite element meshes (h-adaptivity) by subdividing a rectangular element into four, and introducing the L_2 projection of strains for error estimation in solving elasto-dynamic problems. Molinari and Ortiz [10] performed AMR and coarsening for dynamic-plastic problems. The mesh refinement scheme made use of Rivara's longest-edge propagation path bisection

*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

algorithm [11], whereas the coarsening scheme was based on the elimination of elements by edge-collapse. Cavin *et al.* [12, 13] presented automatic grid refinement for linear quasi-static problems, and developed space–time automatic refinement for one-dimensional linear dynamic problems.

From a broader perspective, AMR techniques can be a good candidate for multiscale numerical methods. For example, Bryan [14] utilized the technique in cosmology in order to represent the evolution of a cluster of galaxies in high-resolution. Trangenstein [15] investigated multi-component flow in porous media. Other multiscale numerical methods have been developed in conjunction with various techniques, for instance, variational multiscale method [16], multi-scale enrichment method [17], and multi-scale finite element method [18]. Notice that Hou and Wu [18] emphasized the importance of boundary conditions between coarse and fine scales for accuracy of numerical methods. However, AMR techniques do not suffer from accuracy of boundary conditions because the finite element mesh can be fully compatible between coarse and fine meshes. Nevertheless, the major challenge in adaptive finite element analysis is to transfer field variables between two discretizations, and to maintain a data structure to retrieve adjacency relationships efficiently.

Non-linear crack tip behavior may be represented by several computational methods such as intrinsic cohesive zone models [19, 20], extrinsic cohesive zone models [21, 22], space–time finite element method [23], finite elements with embedded discontinuities [24, 25], and enrichment functions (i.e. generalized/extended finite element methods, GFEM/XFEM) [26–28]. For instance, when a cohesive surface network is inserted *a priori* within a potential crack path domain [19], the so-called intrinsic cohesive zone model results. The traction-separation relationships of the intrinsic cohesive zone model include an initial elastic range, which can provide significant artificial compliance, especially when cohesive surface elements are inserted in a large domain [29].

Alternatively, discontinuous fracture surfaces are described by utilizing discontinuous enrichment functions (e.g. branch functions, Heaviside function) in GFEM/XFEM [26, 27, 30, 31]. However, non-polynomial enrichment functions lead to additional computational costs involving, for example, numerical integration [32]. For transient problems in XFEM, time stepping schemes require special care because discontinuous enrichment functions are time dependent due to the position of moving discontinuities [33]. For instance, Remmers *et al.* [34] indicated that '*an element can be crossed by a discontinuity in such a way that one of the two resulting parts of the element becomes so small that the critical time increment for a stable solution procedure will become almost infinitesimal*'. In this general context, according to Bishop [35], '*once crack branching and crack coalescence phenomena appear*, *the prospect of modeling a multitude of arbitrary three-dimensional intersecting cracks quickly becomes untenable*'.

Among various computational methods, the extrinsic cohesive zone model is the choice of the current work. In the extrinsic cohesive zone models, cohesive surface elements are adaptively inserted on the basis of external crack initiation criteria whenever and wherever they are necessary. Thus, this approach does not lead to artificial compliance that exists in the intrinsic cohesive zone models. Previous extrinsic cohesive zone modeling approaches involved domain discretization using either a uniformly fine mesh or a fine mesh in a potential crack propagation region.

This paper deals with dynamic cohesive fracture problems. A fine mesh is utilized to capture micro-cracks and non-linear crack tip behavior, whereas a coarse mesh is employed in a far field from the crack tip region in conjunction with adaptive mesh refinement and coarsening (AMR&C) schemes. The mesh adaptation (refinement and coarsening) is performed systematically by keeping the topology of the evolving **4k** mesh with edge-split and vertex-removal operators. The mesh adaptation is incorporated with TopS [36, 37], which provides compact and efficient topological representation. The non-linear cohesive zone is represented by cohesive surface elements, and the Park–Paulino–Roesler (PPR) potential-based model [38] is utilized for the constitutive relationship of mixed-mode cohesive fracture. In addition, nodal perturbation and edge-swap operators are employed to reduce mesh bias and to improve crack patterns in **4k** meshes [39]. The computational framework is verified and validated by solving problems involving a predefined crack path, mixed-mode crack propagation, and crack branching instability.

This paper is organized as follows. Section 2 presents AMR schemes, interpolation of new nodes and energy balance, whereas Section 3 explains adaptive mesh coarsening (AMC) schemes and coarsening criterion. Finite element formulations, time integration, and the constitutive relationship

DOI: 10.1002/nme

of the PPR potential-based model are described in Section 4. Computational results of mode I fracture, mixed-mode fracture and crack branching problems are provided in Section 5. Finally, Section 6 summarizes the key findings of the paper.

2. ADAPTIVE MESH REFINEMENT (AMR)

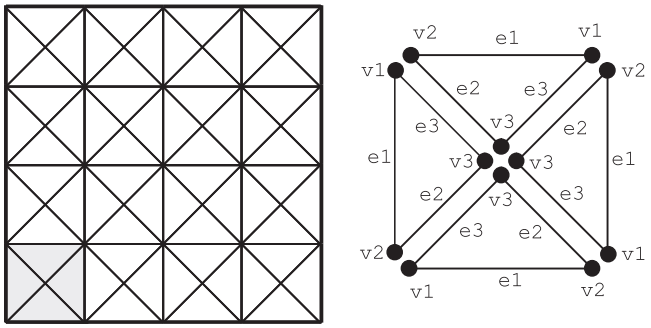
The AMR strategy requires an efficient and robust data structure in order to retrieve adjacency relationships. In this study, the topological data structure named TopS [36, 37] is used for representing **4k** meshes. Notice that TopS provides a linear time scale for obtaining adjacent information and inserting cohesive elements [40]. This data structure provides efficient element-to-element adjacency and representation of vertices and edges, which are convenient for the implementation of refinement and coarsening operations. A sample initial **4k** mesh and the convention used for the local incidences of elements are shown in Figure 1. The basic mesh is composed of a uniform set of right isosceles triangles and internal vertices with degree 4 or 8. Element incidence is defined such that the third local vertex (`v3`) of each triangle is associated with the right angle and corresponds to a global vertex of degree 4. The first local edge (`e1`) is opposite the third vertex (`v3`).

In the following subsection, mesh refinement schemes of **4k** meshes are explained in conjunction with edge-split operators. Next, AMR criterion, interpolation schemes of new nodal quantities, and energy balance are presented.

2.1. Mesh refinement schemes

Mesh refinement is implemented by using an edge-split operator. The operator subdivides an edge into two new other edges. In order to maintain the **4k** structure, internal edge splits are defined on specific clusters of two triangles (as in the literature [41]) and boundary edges. Figure 2 illustrates the two possible cases of the edge split. An interior edge split replaces the two adjacent triangles with four triangles (Figure 2(a)), whereas a boundary edge split replaces a triangle with two triangles (Figure 2(b)).

In a **4k** mesh, the edge split can only be applied to the first local edge (`e1`), described in Figure 1(b), of the adjacent triangles. When the edge-split operator is requested for the other edges (`e2` or `e3`), a recursive procedure is introduced in order to transform the local sub-mesh into one of the patterns in Figure 2. The procedure works by splitting edges of neighboring triangles until an allowable pattern is achieved, i.e. recursive edge-split procedure, as illustrated in Figure 3. When the dashed edge in Figure 3(a) is to be split, the edge does not fit in any of the patterns in Figure 2. Then, other edges are split (Figure 3(b)) in order to create an allowable pattern for the edge's split



(a) (b)

Figure 1. (a) Sample initial **4k** mesh where thin lines indicate interior edges and thick lines indicate boundary edges and (b) local nodal incidences (topological entities) of triangles of a region of the **4k** mesh.

DOI: 10.1002/nme

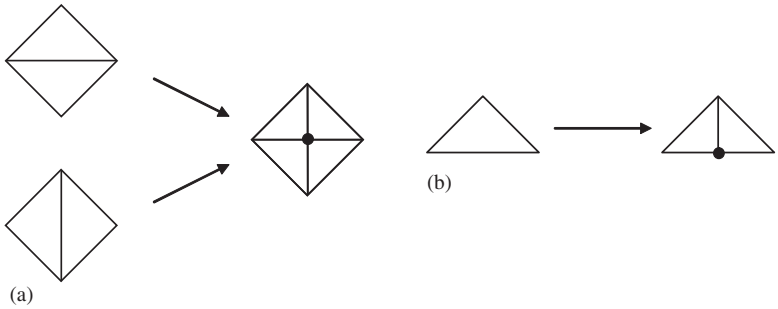


Figure 2. Local mesh patterns for which edge splits are allowed: (a) mesh around an interior edge and (b) boundary edge.

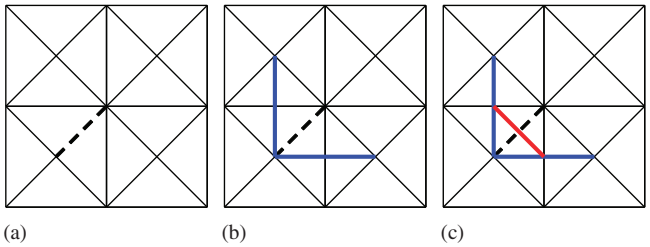


Figure 3. Recursive edge split procedure: (a) request for an edge-split along the edge with a dashed line; (b) split edges to create an allowable pattern; and (c) split edge along the dashed edge.

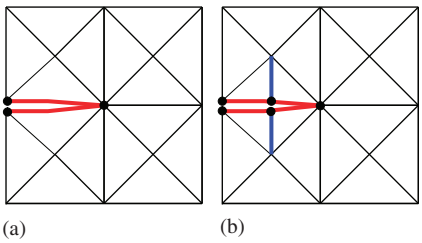


Figure 4. Edge split along a cohesive surface element.

(Figure 3(c)). Therefore, one can select any edge in a **4k** mesh for the edge-split operator. A similar strategy could be employed for completely unstructured triangular meshes, using, for instance, the longest-edge bisection procedure [42]. One advantage of using the **4k** mesh, combined with the edge-swap operator, consists in offering eight different directions for crack propagation at every vertex, thus improving crack path prediction. Mesh orientation dependency, introduced by the regular geometry of **4k** structured meshes, is avoided by the use of nodal perturbation [39].

In addition, edge splits are allowed when a cohesive element exists in between the two triangles of the patterns of Figure 2(a). When one of the edges of the cohesive elements is split, the other edge is also split by the edge-split operator in order to keep mesh topology consistent. As a consequence, the cohesive element is divided into two new elements (Figure 4). However, the edge-split operator on cohesive elements is not employed in computational simulation because the finite element mesh is refined before the insertion of cohesive elements.

During the edge-split operations, edge depths are computed. The initial depth of all the edges of an unmodified mesh is zero. The value assigned to the newly created edges is equal to the maximum depth of all the edges of the triangles adjacent to the split edge, plus one (`MaxDepth(edges) + 1`). For example, when the horizontal dashed edge on the left is split,

DOI: 10.1002/nme

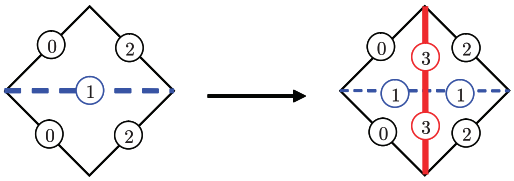


Figure 5. Example of the edge-split operation and computed edge depth.

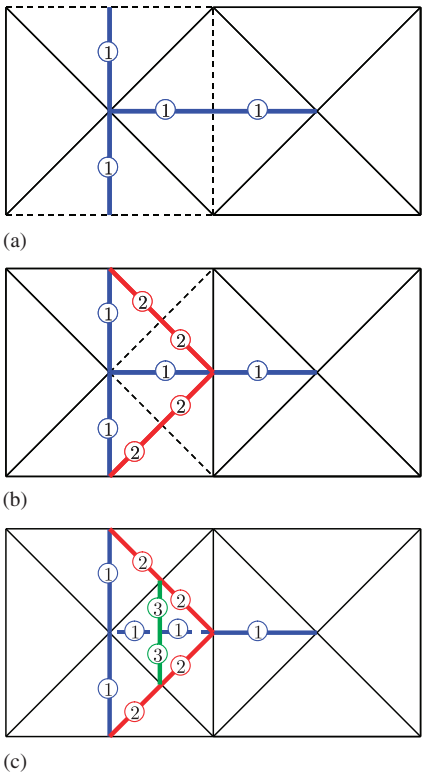


Figure 6. Sequence of edge-split operations along the dashed edges.

the two vertical edges are created, as shown in Figure 5. The depth value assigned to the vertical edges is ③, which is equal to the maximum depth of the edges (②) of the adjacent triangles plus one. The depth of the two new horizontal edges (①) is the same as the original edge. In addition, Figure 6 shows a sequence of edge splits executed on a sample mesh. At each step, the dashed edges are split, and depth values are assigned to the newly created ones, shown in Figures 6(a)–(c). The depth values are illustrated by the numbered labels on them. Notice that labels of edges with zero-depth value (initial value) are omitted. The depth value is associated with a relative local ordering for edge splits, which is utilized for AMC schemes, as discussed in the next section.

2.2. *Refinement criterion and interpolation of new nodes*

In the computational simulation of cohesive fracture problems, AMR is performed by employing an *a priori* assumption that a relatively fine mesh is utilized around the crack tip region in order to capture high stress and/or strain gradients. In order to have a uniformly refined mesh around the crack tip region, the edge-split operator is used non-recursively. The mesh refinement procedure is to split all the allowable edges located within a circular region around the crack tip. This is done in several steps until a maximum level of refinement is reached. Figure 7 illustrates the refinement

DOI: 10.1002/nme

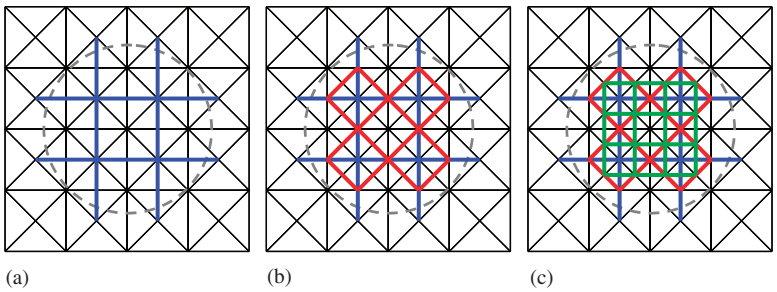


Figure 7. Three refinement steps executed inside a circular region: (a) level 1; (b) level 2; and (c) level 3.

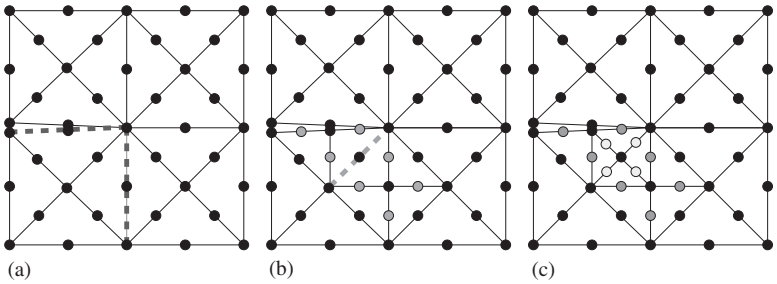


Figure 8. Example of the edge-split operations.

procedure on a sample mesh for three levels of refinement. However, the edge-split operator along a cohesive element is not used in the computational simulation, as mentioned previously.

The AMR strategy with the edge-split operations leads to new nodes whose quantities should be interpolated from adjacent nodes in the non-linear finite element analysis. For example, in the **4k** mesh with quadratic triangular elements of six nodes each (Figure 8(a)), two dashed edges are split: one edge is on the boundary and the other is in the interior. The boundary edge-split operation generates three dark-gray nodes, whereas the interior edge-split operation leads to four dark-gray nodes, as shown in Figure 8(b). One can further split the dashed interior edge in Figure 8(b), and this operation provides four additional light-gray nodes, as shown in Figure 8(c). Notice that these new nodes can be locally relocated (or perturbed) at the time of the insertion of new nodes (if one utilizes the nodal perturbation). In addition, nodal quantities (e.g. displacement, velocity, and acceleration) of new nodes should be interpolated from neighboring nodes for the explicit time integration in elasto-dynamic problems.

For the interpolation of new nodes, the choices of sampling nodes and interpolation functions are essential. First, sampling nodes are selected in a recursive procedure. For example, the three new dark-gray nodes, which resulted from the boundary edge-split operation, are interpolated from six nodes, which correspond to the members of a triangular element that shares the boundary edge (Figure 9(a)). Next, the four new dark-gray nodes, which resulted from the interior edge-split operation, are interpolated from nine nodes, which are the members of the adjacent elements that share the interior edge (Figure 9(a)). Similarly, the four light-gray nodes are interpolated from nine nodes, which are the members of the adjacent elements that share the interior edge, as shown in Figure 9(b). Because the light-gray nodes are interpolated from five black nodes and four dark-gray nodes, the dark-gray nodes are interpolated before the light-gray nodes are interpolated. In other words, new nodes which are inserted first should be interpolated first. Therefore, the sequence of the interpolations is the same as the sequence of the edge-split operations.

For interpolation functions, the quadratic Lagrange basis functions (i.e. shape functions) are utilized. The nine node quadratic interpolation functions (i.e. Q9 shape functions) are used for the interior edge split, whereas the six node quadratic interpolation functions (i.e. T6 shape functions) are utilized for the boundary edge split. Note that the six node quadratic interpolation functions

DOI: 10.1002/nme

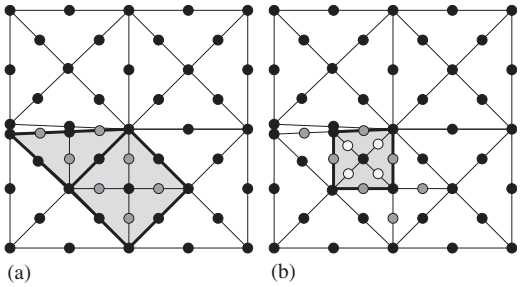


Figure 9. Interpolation scheme for new nodes.

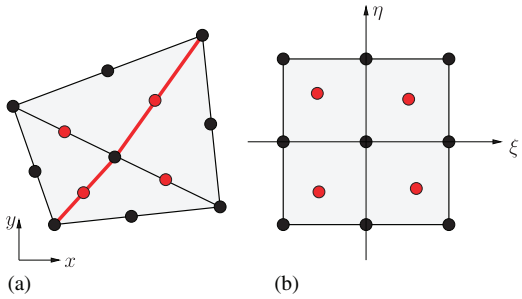


Figure 10. (a) Physical coordinate system and (b) mapped coordinate system.

could be alternatively used for the interior edge split. However, the nine node interpolation generally leads to better approximation of local node quantities than the six node interpolation because it uses more local information for the interpolation, as discussed in Section 2.3. The approximated nodal quantities of a new node (e.g. displacement, $\widetilde{u_i}$) are obtained by

$$\widetilde{u_i} = \sum_{j=1}^{n} N_j(\xi_i, \eta_i) u_j \tag{1}$$

where n is the number of interpolation functions, N_j are the Lagrange basis functions, and ξ_i and η_i are nodal locations in the mapped coordinate system (e.g. Figure 10(b)). Alternatively, the new nodal quantities could be evaluated by using a superconvergence patch recovery technique [43, 44].

The nodal locations of new nodes are mapped from the physical coordinate $(x - y)$ system to the mapped coordinate $(\xi - \eta)$ system. For example, the transformation of a quadrilateral from the mapped coordinate system (Figure 10(b)) to the physical coordinate system (Figure 10(a)) is given as $T : (\xi, \eta) \longrightarrow (x, y)$, where $x = \sum_{i=1}^{n} N_i(\xi, \eta) x_i$ and $y = \sum_{i=1}^{n} N_i(\xi, \eta) y_i$. Then, the inverse mapping $(T^{-1} : (x, y) \longrightarrow (\xi, \eta))$ is performed by utilizing Newton's algorithm to evaluate the nodal locations of new nodes in the mapped coordinate system. Notice that the size of non-linear system for the inverse mapping in two-dimensions is two, and thus the computational cost of this inverse mapping is not significant.

2.3. *Energy balance*

Energy balance is one of the essential aspects in AMR. The proposed AMR scheme leads to the strain energy conservation when the finite element shape functions are utilized for the interpolation of new nodal quantities. In order to demonstrate the strain energy conservation, let us consider a rectangular domain. The domain is discretized into two different ways: one with two T6 elements (Figure 11(a)), and the other with four T6 elements (Figure 11(b)). Notice that two triangular elements in Figure 11(a) are split into four triangular elements in Figure 11(b), which corresponds to the interior edge split. Thus, Figure 11(a) can be considered as a discretization before the mesh refinement, whereas Figure 11(b) can be considered as a discretization after the mesh refinement.

DOI: 10.1002/nme

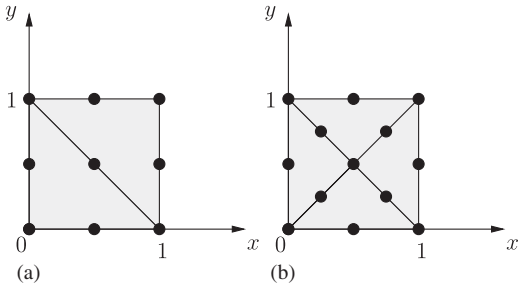


Figure 11. Rectangular domain discretized into (a) two linear strain triangular (T6) elements and (b) four linear strain triangular (T6) elements.

If the energy in Figure 11(a) is the same as the energy in Figure 11(b), the energy is conserved during the mesh refinement.

First, a quadratic displacement field is assumed as $u_x(x, y) = x^2 + 2xy$ and $u_y(x, y) = y^2 + 2xy$ within the rectangular domain. For the plane stress condition, the exact strain energy is $7E(5-3\nu)/6(1-\nu^2)$ where E is the elastic modulus and ν is the Poisson ratio. The exact strain energy is obtained from the finite element analysis with two T6 elements and four T6 elements, as expected. This is because the finite element formulation captures the exact strain energy for the linear strain field (or quadratic displacement field) when a linear strain triangular element (T6) is utilized.

Next, let us consider a cubic displacement field, which is given as $u_x(x, y) = x^3$ and $u_y(x, y) = y^3$. For the plane stress condition, the exact strain energy is given as $E(9 + 5\nu)/45(1 - \nu^2)$ within the rectangular domain. If E and ν are 1 and 0, the exact strain energy is $\frac{1}{5}$. In the finite element calculation, the strain energy of two T6 elements (Figure 11(a)) is 0.194444. In the calculation of the strain energy of the four T6 elements (Figure 11(b)), the nodal displacements are obtained from the interpolation using the Q9 shape functions because the exact displacement field is usually unknown during finite element simulation. Thus, the displacement field used in Figure 11(b) corresponds to the span of interpolation functions. The obtained strain energy is 0.194444, which is the same as the strain energy of the two T6 elements (i.e. strain energy conservation). The same results are obtained when the T6 shape functions are used for the interpolation.

Finally, let us consider a cubic displacement field within the rectangular domain where $u_x(x, y) = x^2y$ and $u_y(x, y) = xy^2$. Note that this displacement field includes terms that correspond to the span of the Q9 shape functions (i.e. x^2y and xy^2), while the previous displacement field did not include any term related to the Q9 shape functions. For the plane stress condition, the exact strain energy is $E(27+13\nu)/45(1-\nu^2)$. The exact strain energy is 0.6 when $E = 1$ and $\nu = 0$. In the finite element calculation, the strain energy with two T6 elements (Figure 11(a)) is 0.614583. The strain energy with four T6 elements (Figure 11(b)) is 0.607422 when the Q9 shape functions are used for the interpolation of the displacement field. When the T6 shape functions are used for the interpolation of the displacement field, the strain energy of the four T6 elements (Figure 11(b)) is 0.614583, which corresponds to the strain energy of the two T6 elements (Figure 11(a)). Thus, one obtains better approximation with the Q9 shape functions than the T6 shape functions in this case because more nodal quantities are used for the new node interpolation, which can lead to better approximation for the displacement field.

Moreover, the AMR scheme leads to the kinetic energy conservation when the nodal velocities are interpolated using the same shape functions and the consistent mass matrix is utilized. For the numerical integration of the consistent mass matrix (e.g. polynomial order of 4 for T6 elements), the quadrature rule proposed by Dunavant [45] is utilized. Similar to the strain energy, for example, let us assume a cubic velocity field, $\dot{u}_x(x, y) = x^3$ and $\dot{u}_y(x, y) = 0$, within the rectangular domain. If the density is one, the exact kinetic energy is 0.071429. The kinetic energy is estimated for the two different discretizations, shown in Figures 11(a) and (b). Note that the finite element formulation is able to represent the exact kinetic energy for the quadratic velocity field. The kinetic energy with two

DOI: 10.1002/nme

T6 elements (Figure 11(a)) is 0.079167. The same kinetic energy is obtained from the discretization shown in Figure 11(b) when the Q9 (or T6) shape functions are used for the new nodal velocity interpolation. Next, let us assume a cubic velocity field, $\dot{u}_x(x, y) = x^2 y$ and $\dot{u}_y(x, y) = 0$, within the rectangular domain, which includes a term that related to the span of the Q9 shape functions (i.e. $x^2 y$). The exact kinetic energy is $\frac{1}{30}$ when the density is one. For the finite element calculation, the kinetic energy with two T6 elements (Figure 11(a)) is 0.031944. The kinetic energy with four T6 elements (Figure 11(b)) is 0.033203 when the Q9 shape functions are used for the interpolation, and is 0.031944 when the T6 shape functions are used. The use of the Q9 shape functions provides better approximation for the velocity fields than the use of the T6 shape functions in this example because the span of the Q9 shape functions provides larger space than the span of the T6 shape functions. Note that the results are consistent with the results for the strain energy examples.

In summary, the strain energy is conserved during the edge-split of linear strain triangular elements when the nodal displacements are interpolated with the same shape functions (i.e. T6 shape functions). Moreover, the kinetic energy is conserved during the edge-split when the nodal velocities are interpolated with the same shape functions and the consistent mass matrix is utilized. In the current approach, the T6 shape functions are utilized for the interpolation of new nodal quantities associated with the boundary edge split, whereas the Q9 shape functions are utilized for the interpolation of new nodal quantities related to the interior edge split so that one generally obtains better approximation.

In addition, the equilibrium on the refined mesh can be satisfied in a weak sense. For a given displacement vector, the corresponding internal force vector satisfies the equilibrium in a weak sense along the horizontal and vertical directions for two-dimensional problems. However, nodal equilibrium is not individually satisfied. Note that the summation of the internal force along each direction is equal to zero before and after mesh refinement. The remaining force associated with the equilibrium is the force related to Newton's second law because the finite element meshes are refined before the insertion of cohesive elements. For a given acceleration field, the sum of the equivalent nodal forces associated with Newton's second law before the edge split is the same as the sum of the forces of the split elements (i.e. after the edge split) in each direction when the consistent mass matrix is used.

3. ADAPTIVE MESH COARSENING (AMC)

The coarsening procedure reverses a sequence of edge-splits by utilizing a vertex-removal (or edge-collapse) operator. This operator works by removing a vertex and merging adjacent elements in order to restore a previous level of refinement. In this section, mesh coarsening schemes and coarsening criterion are presented.

3.1. Mesh coarsening schemes

Mesh coarsening consists of a sequence of vertex-removal operators. Mesh structure is maintained by constraining internal vertex removals to vertices which have four adjacent elements [41, 46], and boundary vertex removals to vertices which have two adjacent elements. This corresponds to the four-face and two-face triangle clusters presented, respectively, in Figures 12(a) and (b). Because the local sub-mesh in Figure 12(a) can be originated from two distinct cases (Figure 2), the refinement procedure must record historical data so that ambiguities can be consistently handled. The pattern in Figure 12(b) occurs on mesh boundaries, and does not require handling of ambiguous cases.

In order to reverse a sequence of mesh refinements in a consistent way, the vertex-removal operator needs to determine from which pattern the current local sub-mesh originated. To accomplish this, we record the history of edge-split operations executed. However, instead of keeping a direct acyclic graph of edge splits explicitly, a single integer value is assigned to each edge in order to identify its corresponding *depth* in the graph. Edge depth values are stored during the edge-split operations, as discussed in the previous section. This is sufficient for the purpose of the adaptive procedures used in this work and leads to memory reduction.

DOI: 10.1002/nme

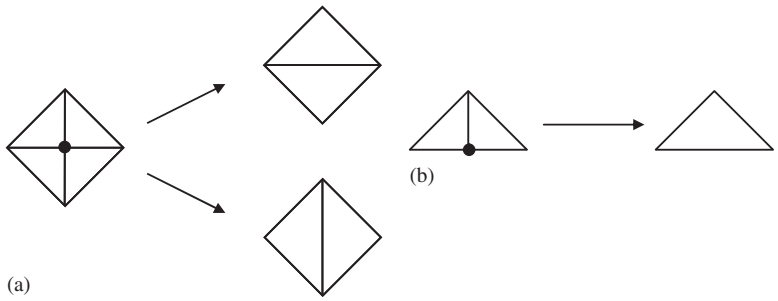


Figure 12. Vertex-removal operator for mesh coarsening: (a) interior vertex-removal and (b) boundary vertex-removal.

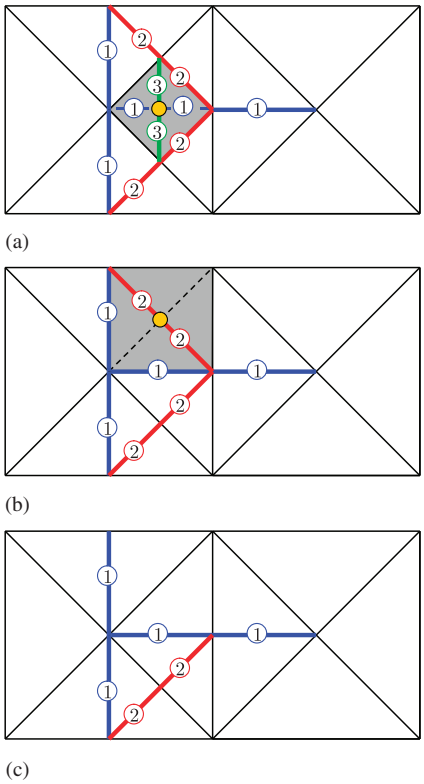


Figure 13. Sequence of vertex-removal operations in shaded **4k** patches.

The execution of the vertex-removal operator is illustrated in Figure 13. For each removed vertex, the incident edges with the greatest depth are deleted, and the corresponding adjacent elements are merged. For example, the vertex-removal operator is applied in a shaded **4k** patch. The maximum edge depths in the **4k** patches in Figures 13(a) and (b) are ③ and ②, respectively. Then, the two edges with higher edge depth are deleted, and the other two edges are merged.

In order to preserve attributes and geometry of crack paths and mesh boundary, we impose some additional constraints to the coarsening procedure. The vertex-removal operator is applied to neither vertices of cohesive elements nor the mesh boundary. Edge swaps can occur as cracks propagate [39], and undoing edge swaps is not necessary because vertices of cohesive elements are not removed.

DOI: 10.1002/nme

3.2. Coarsening criterion and local update

The AMC strategy is based on *a posteriori* coarsening criterion, i.e. the root mean square of strain error, which is convenient to use because it is non-dimensional. The strain error for volume of the ith element (Ω_i) is expressed as

$$e_i = \left[\frac{\int_{\Omega_i} (\epsilon_{\text{exact}} - \overline{\epsilon})^{\text{T}} (\epsilon_{\text{exact}} - \overline{\epsilon}) \, \text{d}\Omega}{|\Omega_i|} \right]^{1/2} \tag{2}$$

where ϵ_{exact} is the exact strain, and $\overline{\epsilon}$ is the strain obtained from the finite element solution. In this study, because an original **4k** patch (Figure 14(a)) generally provides better approximation in displacement field than a coarsened **4k** patch (Figure 14(b)), the strain of the original **4k** patch ($\overline{\epsilon}_{\text{f}}$) is used for ϵ_{exact} while the strain of the coarsened **4k** patch ($\overline{\epsilon}_{\text{c}}$) is utilized for $\overline{\epsilon}$. Then, the strain error indicator for the **4k** patch is given as:

$$e_{4k} = \left[\sum_{i=1}^{4} \frac{\int_{\Omega_i} (\overline{\epsilon}_{\text{f}} - \overline{\epsilon}_{\text{c}})^{\text{T}} (\overline{\epsilon}_{\text{f}} - \overline{\epsilon}_{\text{c}}) \, \text{d}\Omega}{|\Omega_i|} \right]^{1/2} \tag{3}$$

Note that the integration is numerically performed in the four elements (Ω_i) of the original **4k** patch. When the approximated error is smaller than a specified value (e.g. 2%), an original **4k** patch is coarsened. In addition, mesh coarsening is not performed around a crack tip region in order to avoid consecutive mesh coarsening and refinement. Notice that Belytschko and Tabbara [9] introduced a strain-projection criterion by replacing ϵ_{exact} with the strain obtained from the L_2 projection of the finite element strain. In addition, the root mean square of strain error, the L_2 norm of strain, and the energy norm in terms of strain have a similar structure, and thus these can be similarly approximated [47].

The AMC strategy leads to the removal of shape functions while the AMR results in the addition of shape functions. The reduction of a solution space (i.e. mesh coarsening) does not guarantee conservation of strain energy between the original **4k** patch (Figure 14(a)) and the coarsened **4k** patch (Figure 14(b)). In order to minimize the difference of the strain energies, the nodal displacement of a center node ($\mathbf{u}_9$) is locally modified by matching the strain energy of the two different discretizations. Notice that this modification does not influence the adjacent elements of the **4k** patch. The updated displacement ($\widetilde{\mathbf{u}}_9$) is utilized to evaluate the velocity and acceleration in a coarsened **4k** patch, which is explained in the time integration section.

The nodal displacement of the center node can be updated by two approaches. First, one can optimize the displacement by minimizing the difference between the original displacement ($\mathbf{u}_9$) and the updated displacement ($\widetilde{\mathbf{u}}_9$),

$$\min \|\mathbf{u}_9 - \widetilde{\mathbf{u}}_9\|^2 \tag{4}$$

under the condition such that

$$g(\widetilde{\mathbf{u}}_9) = E_{\text{int}} - \widetilde{E}_{\text{int}}(\widetilde{\mathbf{u}}_9) = 0 \tag{5}$$

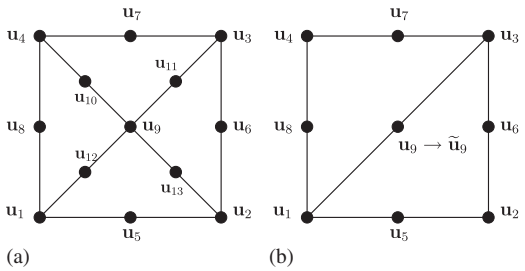


Figure 14. (a) Original **4k** patch and (b) coarsened **4k** patch.

DOI: 10.1002/nme

where E_{int} is the strain energy in the original **4k** patch, and $\widetilde{E}_{\text{int}}$ is the strain energy in the coarsened **4k** patch. The strain energy in the coarsened **4k** patch ($\widetilde{E}_{\text{int}}$) is a function of the updated displacement ($\widetilde{\mathbf{u}}_9$). The constraint function can be expressed in closed form, and the optimization problem is solved by introducing a Lagrange multiplier. This optimization problem can be solved efficiently because the number of unknowns is two and because the objective function and the constraint function are quadratic, which leads to a constant Hessian matrix of the objective function.

The second approach consists of updating the nodal displacement ($\mathbf{u}_9$) by equating the strain energy in the original **4k** patch to the strain energy in the coarsened **4k** patch along the horizontal and vertical directions for two-dimensional problems. The strain energy of the original **4k** patch is decomposed into the strain energy associated with the horizontal direction ($E_{\text{int}(x)}$) and the energy related to the vertical direction ($E_{\text{int}(y)}$), i.e. $E_{\text{int}} = E_{\text{int}(x)} + E_{\text{int}(y)}$. Similarly, the strain energy of the coarsened **4k** patch is also divided into the two components, i.e. $\widetilde{E}_{\text{int}}(\widetilde{\mathbf{u}}_9) = \widetilde{E}_{\text{int}(x)}(\widetilde{\mathbf{u}}_9) + \widetilde{E}_{\text{int}(y)}(\widetilde{\mathbf{u}}_9)$. Each directional strain energy is set to be the same, which provides two equations for two unknowns. Then, these non-linear equations are solved by using Newton's algorithm.

4. COMPUTATIONAL IMPLEMENTATION

The governing equations of dynamic fracture problems are solved by Galerkin finite element methods. A strong form is converted to a weak form, and the explicit method is utilized for the time integration. The finite element formulation, time integration, and the constitutive relation of cohesive fracture are explained in this section.

4.1. Finite element formulation

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 ($\mathbf{T}^{\text{coh}}$),

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

where Ω_0 and Γ_0 are domain and boundary in the reference configuration, respectively. In addition, Γ_{coh} denotes boundary associated with cohesive fracture surface at a given time in the reference configuration. The second Piola–Kirchhoff stress (**S**) and the Green deformation strain (**E**) are utilized. The superposed dots in $\ddot{\mathbf{u}}$ denote second time derivatives, **u** is the displacement vector, $\boldsymbol{\Delta}$ is the 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 is used.

4.2. Time integration

The equation of motion without a damping term is expressed as

$$\mathbf{M}\ddot{\mathbf{u}}_n + \mathbf{K}\mathbf{u}_n = \mathbf{R}_n^{\text{ext}} \tag{7}$$

where **M** and **K** are the mass matrix and the stiffness matrix, respectively. In addition, $\ddot{\mathbf{u}}_n$, $\mathbf{u}_n$, and $\mathbf{R}_n^{\text{ext}}$ are the nodal acceleration, nodal displacement, and external nodal force at time step n, respectively. For cohesive fracture problems, **K** consists of a continuum part (i.e. bulk element contribution) and a cohesive fracture part (i.e. cohesive surface element contribution). The stiffness matrix is decomposed as $\mathbf{K} = \mathbf{K}^b + \mathbf{K}^c$ where $\mathbf{K}^b$ is the stiffness matrix of bulk elements and $\mathbf{K}^c$ is the stiffness matrix of cohesive surface elements.

This equation of motion is solved by the Newmark algorithm [48]. The Newmark algorithm evaluates the displacement ($\mathbf{u}_{n+1}$) and velocity ($\dot{\mathbf{u}}_{n+1}$) at time $n+1$ on the basis of the displacement

DOI: 10.1002/nme

($\mathbf{u}_n$), velocity ($\dot{\mathbf{u}}_n$), and acceleration ($\ddot{\mathbf{u}}_n$) at time n with the acceleration ($\ddot{\mathbf{u}}_{n+1}$) at time $(n+1)$, which are expressed as

$$\begin{aligned}\mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta t\dot{\mathbf{u}}_n + (\frac{1}{2}-\beta)\Delta t^2\ddot{\mathbf{u}}_n + \beta\Delta t^2\ddot{\mathbf{u}}_{n+1}\\ \dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n + (1-\gamma)\Delta t\ddot{\mathbf{u}}_n + \gamma\Delta t\ddot{\mathbf{u}}_{n+1}\end{aligned} \tag{8}$$

where parameters β and γ are associated with the stability and accuracy of the algorithm. The acceleration at time $(n+1)$ is obtained from the equation of motion

$$(\mathbf{M}+\beta\Delta t^2\mathbf{K})\ddot{\mathbf{u}}_{n+1} = \mathbf{R}^{\text{ext}}_{n+1} - \mathbf{K}(\mathbf{u}_n + \Delta t\dot{\mathbf{u}}_n + (\frac{1}{2}-\beta)\Delta t^2\ddot{\mathbf{u}}_n). \tag{9}$$

The proposed AMR&C strategy leads to the change of discretization, and thus the energy conservation is essential for the stability and accuracy of the algorithm. The kinetic energy (E_{kin}) and the strain energy (E_{int}) evolution between t_n and t_{n+1} is expressed as

$$\begin{aligned}&\frac{1}{2}\dot{\mathbf{u}}^{\mathrm{T}}_{n+1}\mathbf{M}_{n+1}\dot{\mathbf{u}}_{n+1} - \frac{1}{2}\dot{\bar{\mathbf{u}}}^{\mathrm{T}}_n\bar{\mathbf{M}}_n\dot{\bar{\mathbf{u}}}_n + \frac{1}{2}\mathbf{u}^{\mathrm{T}}_{n+1}\mathbf{K}^b_{n+1}\mathbf{u}_{n+1} - \frac{1}{2}\bar{\mathbf{u}}^{\mathrm{T}}_n\bar{\mathbf{K}}^b_n\bar{\mathbf{u}}_n\\ &\quad= \frac{1}{2}[\dot{\mathbf{u}}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{u}} + \mathbf{u}^{\mathrm{T}}\mathbf{K}^b\mathbf{u}]^{n+1}_n + \underbrace{\frac{1}{2}\dot{\mathbf{u}}^{\mathrm{T}}_n\mathbf{M}_n\dot{\mathbf{u}}_n - \frac{1}{2}\dot{\bar{\mathbf{u}}}^{\mathrm{T}}_n\bar{\mathbf{M}}_n\dot{\bar{\mathbf{u}}}_n}_{\Delta E_{\text{kin}}} + \underbrace{\frac{1}{2}\mathbf{u}^{\mathrm{T}}_n\mathbf{K}^b_n\mathbf{u}_n - \frac{1}{2}\bar{\mathbf{u}}^{\mathrm{T}}_n\bar{\mathbf{K}}^b_n\bar{\mathbf{u}}_n}_{\Delta E_{\text{int}}}\end{aligned} \tag{10}$$

where the superposed bar denotes the discretization before the mesh modification. For example, $\bar{\mathbf{K}}^b_n$ and $\bar{\mathbf{u}}_n$ are the stiffness matrix and displacement vector before the mesh modification at time t_n, whereas $\mathbf{K}^b_n$ and $\mathbf{u}_n$ are the stiffness matrix and displacement vector after the mesh modification at time t_n.

The first term on the right-hand side of Equation (10) is the kinetic energy and the strain energy evolution between t_n and t_{n+1} based on the fixed discretization. In the Newmark algorithm, the energy evolution between t_n and t_{n+1} for a fixed discretization is expressed [49] as

$$\frac{1}{2}[\dot{\mathbf{u}}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{u}} + \mathbf{u}^{\mathrm{T}}\mathbf{K}^b\mathbf{u}]^{n+1}_n = \frac{1}{2}(\dot{\mathbf{u}}_{n+1}+\dot{\mathbf{u}}_n)^{\mathrm{T}}\mathbf{M}(\dot{\mathbf{u}}_{n+1}-\dot{\mathbf{u}}_n) + \frac{1}{2}(\mathbf{u}_{n+1}+\mathbf{u}_n)^{\mathrm{T}}\mathbf{K}^b(\mathbf{u}_{n+1}-\mathbf{u}_n), \tag{11}$$

where $\mathbf{K}^b_n$ and $\mathbf{K}^b_{n+1}$ are assumed to be the same (i.e. linear elastic bulk material). The substitution of Equation (8) into the energy evolution expression (11) leads to

$$\begin{aligned}\frac{1}{2}[\dot{\mathbf{u}}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{u}} + \mathbf{u}^{\mathrm{T}}\mathbf{K}^b\mathbf{u}]^{n+1}_n &= \frac{1}{2}(\dot{\mathbf{u}}_{n+1}+\dot{\mathbf{u}}_n)^{\mathrm{T}}\mathbf{M}\left\{\frac{\Delta t}{2}(\ddot{\mathbf{u}}_{n+1}+\ddot{\mathbf{u}}_n) + \left(\gamma-\frac{1}{2}\right)\Delta t(\ddot{\mathbf{u}}_{n+1}-\ddot{\mathbf{u}}_n)\right\}\\ &\quad+\frac{1}{2}(\mathbf{u}_{n+1}+\mathbf{u}_n)^{\mathrm{T}}\mathbf{K}^b\left\{\frac{\Delta t}{2}(\dot{\mathbf{u}}_{n+1}+\dot{\mathbf{u}}_n) + \left(\beta-\frac{1}{2}\gamma\right)\Delta t^2(\ddot{\mathbf{u}}_{n+1}-\ddot{\mathbf{u}}_n)\right\}.\end{aligned} \tag{12}$$

Finally, one obtains the following energy variation expression:

$$\begin{aligned}\frac{1}{2}[\dot{\mathbf{u}}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{u}} + \mathbf{u}^{\mathrm{T}}\mathbf{K}^b\mathbf{u}]^{n+1}_n &= \underbrace{\frac{1}{2}(\mathbf{u}_{n+1}-\mathbf{u}_n)^{\mathrm{T}}(\mathbf{R}^{\text{ext}}_{n+1}+\mathbf{R}^{\text{ext}}_n)}_{E_{\text{ext}}} - \underbrace{\frac{1}{2}(\mathbf{u}_{n+1}-\mathbf{u}_n)^{\mathrm{T}}(\mathbf{K}^c_{n+1}\mathbf{u}_{n+1}+\mathbf{K}^c_n\mathbf{u}_n)}_{E_{\text{fra}}}\\ &\quad+\frac{\Delta t}{2}\left(\gamma-\frac{1}{2}\right)(\dot{\mathbf{u}}_{n+1}+\dot{\mathbf{u}}_n)^{\mathrm{T}}\mathbf{M}(\ddot{\mathbf{u}}_{n+1}-\ddot{\mathbf{u}}_n)\\ &\quad-\frac{\Delta t^2}{2}\left(\beta-\frac{1}{2}\gamma\right)(\ddot{\mathbf{u}}_{n+1}-\ddot{\mathbf{u}}_n)^{\mathrm{T}}\mathbf{M}(\ddot{\mathbf{u}}_{n+1}+\ddot{\mathbf{u}}_n)\end{aligned} \tag{13}$$

by substituting Equations (7) and (8) into Equation (12). The first term on the right-hand side of Equation (13) is the external work (E_{ext}) between the time steps, and the second term is associated with the fracture energy (E_{fra}). The remaining two terms are zero for the case of $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$, which lead to the energy conservation in the time increment from n to $(n+1)$. For the central

DOI: 10.1002/nme

difference method (i.e. $\gamma = \frac{1}{2}$ and $\beta = 0$), the last term contributes to the total energy. Because of the Δt^2 in this term, one can reduce this contribution by selecting a smaller time step, which is discussed in Section 5.

The second term (ΔE_{kin}) on the right-hand side of Equation (10) is the difference of the kinetic energy between two discretizations, whereas the last term (ΔE_{int}) corresponds to the difference of the strain energy between two discretizations. Therefore, if the strain energy and the kinetic energy are conserved during mesh modification, both ΔE_{kin} and ΔE_{int} are equal to zero, and the energy evolution with mesh modification becomes the same as the evolution without mesh modification.

Algorithm 1 Explicit time integration for the extrinsic cohesive zone model with AMR&C.

Initialization: displacements ($\mathbf{u}_0$), velocity ($\dot{\mathbf{u}}_0$), acceleration ($\ddot{\mathbf{u}}_0$)
for $n = 0$ to $n_{\max}$ **do**
  Update displacement: $\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t \dot{\mathbf{u}}_n + \Delta t^2/2 \ddot{\mathbf{u}}_n$
  Perform adaptive mesh coarsening (AMC) and update local displacement
  Check if cohesive elements need to be inserted
  Update acceleration: $\ddot{\mathbf{u}}_{n+1} = \mathbf{M}^{-1}(\mathbf{R}^{\text{ext}}_{n+1} + \mathbf{R}^{\text{coh}}_{n+1} - \mathbf{R}^{int}_{n+1})$
  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
  Perform adaptive mesh refinement (AMR) and interpolate nodal quantities
end for

In this paper, the time integration scheme is implemented by using the central difference method (i.e. explicit method). The outline of the procedure is provided 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, AMC is performed on the basis of the coarsening criterion outside the crack tip regions, and local displacements are updated for coarsened **4k** patches to minimize ΔE_{int} in Equation (10). Alternatively, AMC may be performed at the end of time step (i.e. after AMR), and one updates both local displacements and velocities while minimizing ΔE_{int} and ΔE_{kin}, respectively. However, the former approach may be preferable because both acceleration and velocities can be directly calculated on the basis of a coarsened discretization in the former approach.

After AMC, one checks the insertion of cohesive elements based on an external criterion such as the maximum hoop stress [50], the minimum strain energy density [51], or the loss of material stability [28]. In the current study, cohesive elements are adaptively inserted between volumetric elements when the averaged normal traction is greater than the cohesive strength ($\sigma_{\max}$). 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 cohesive force vector ($\mathbf{R}^{\text{coh}}_{n+1}$) is obtained on the basis of the PPR potential-based constitutive model, which is explained in Section 4.4. The consistent mass matrix is diagonalized using a special lumping technique, which provides positive lumped masses [52]. In addition, whenever mesh modification events occur, the consistent mass matrix is obtained in a new configuration and the nodal masses are updated, which leads to mass conservation. The nodal acceleration, velocity, and boundary conditions are updated for the current time step. Finally, the mesh around cohesive surface elements, which are inserted in the current time step, is refined in conjunction with the edge split operators and the nodal perturbation. The new nodes resulting from the edge split operators are relocated on the basis of the nodal perturbation factor (e.g. 0.2) [39]. After the relocation of the new nodes, new nodal quantities are interpolated in recursive order. In addition, the time increment is approximately one order lower than the critical time step, which is also employed by previous non-linear dynamic fracture investigations with the cohesive surface element approach [22, 39].

4.3. Remarks on explicit time integration with GFEM/XFEM

If one employs discontinuous shape functions to represent arbitrary discontinuity within an element, the time increment should be significantly reduced [34]. This is because a discontinuity splits an

DOI: 10.1002/nme

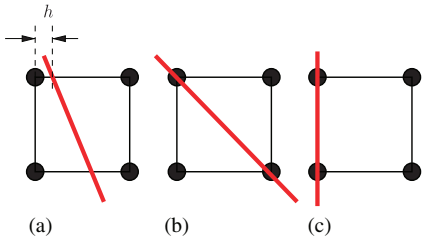


Figure 15. Illustration of discontinuities (GFEM/XFEM) within an element (part(a) has been extracted from Remmers *et al.* [34]).

element, and each split part can be considered as an individual element. Figure 15 illustrates three cases of discontinuities. For the first case (Figure 15(a)), as the discontinuity approaches the node (h is infinitesimal, but h is not equal to zero), the critical time increment for a stable solution procedure will become almost infinitesimal because h governs the element size [34]. Next, if h is equal to zero, which corresponds to Figure 15(b), a rectangular elements is split into two triangular elements, and the element size remains the same. Therefore, one does not need to reduce time increment due to the discontinuity within an element. The third case (Figure 15(c)) also does not require reduction of the time increment because the element size remains the same. Similarly, if one uses cohesive surface elements, discontinuity does not occur within an element, then one does not need to reduce the time increment. In order to avoid an infinitesimal time step associated with an arbitrary discontinuity within the element, Remmers *et al.* [34] did not allow the discontinuity to occur too close to the node so that the split element size is larger than a specified value. They set the minimum split element size as 10% of an original element, and thus the time step with discontinuous shape functions was approximately two orders lower than the critical time step in their computation. Notice that the time increment with the cohesive surface element approach is one order lower than the critical time step.

Alternatively, for linear elastic fracture problems, Menouillard *et al.* [53] introduced a mass lumping strategy, which leads to a block diagonal matrix, to avoid infinitesimal time increments due to a Heaviside enrichment. Elguedj *et al.* [54] developed another mass lumping strategy for enrichment functions associated with the Westergaard solution and, based on empirical calculations, demonstrated that the critical time step of GFEM/XFEM may be half of the critical time step of the standard FEM.

4.4. Constitutive relation of cohesive fracture

For the constitutive relationships of cohesive elements, the unified potential-based model [38], called the PPR model, is utilized. 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

$$\Psi(\Delta_n, \Delta_t) = \min(\phi_n, \phi_t) + \left[\Gamma_n \left(1 - \frac{\Delta_n}{\delta_n}\right)^{\alpha} + \langle \phi_n - \phi_t \rangle\right]\left[\Gamma_t \left(1 - \frac{|\Delta_t|}{\delta_t}\right)^{\beta} + \langle \phi_t - \phi_n \rangle\right] \quad (14)$$

where Δ_n, Δ_t are normal and tangential separations, respectively, and δ_n, δ_t are characteristic length scales. The energy constants are expressed as

$$\Gamma_n = (-\phi_n)^{\langle \phi_n - \phi_t \rangle / \phi_n - \phi_t}, \quad \Gamma_t = (-\phi_t)^{\langle \phi_t - \phi_n \rangle / \phi_t - \phi_n} \quad (\phi_n \neq \phi_t) \quad (15)$$

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

$$\Gamma_n = -\phi_n, \quad \Gamma_t = 1 \quad (\phi_n = \phi_t) \quad (16)$$

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 ($\alpha, \beta \approx 2$).

DOI: 10.1002/nme

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

$$\begin{aligned} T_n(\Delta_n,\Delta_t) &= -\alpha\frac{\Gamma_n}{\delta_n}\left(1-\frac{\Delta_n}{\delta_n}\right)^{\alpha-1}\left[\Gamma_t\left(1-\frac{|\Delta_t|}{\delta_t}\right)^{\beta}+\langle\phi_t-\phi_n\rangle\right] \\ T_t(\Delta_n,\Delta_t) &= -\beta\frac{\Gamma_t}{\delta_t}\left(1-\frac{|\Delta_t|}{\delta_t}\right)^{\beta-1}\left[\Gamma_n\left(1-\frac{\Delta_n}{\delta_n}\right)^{\alpha}+\langle\phi_n-\phi_t\rangle\right]\frac{\Delta_t}{|\Delta_t|} \end{aligned} \tag{17}$$

The above normal and tangential cohesive interactions are defined in a cohesive interaction (or softening) region. When separation is outside a softening region, the cohesive traction is set to be zero. The normal softening region is a rectangular domain associating 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 associating with the conjugate normal final crack opening width ($\bar{\delta_n}$) and the tangential final crack opening width (δ_t). The final crack opening widths (δ_n, δ_t) are expressed as

$$\delta_n = \alpha\phi_n/\sigma_{\max}, \quad \delta_t = \beta\phi_t/\tau_{\max} \tag{18}$$

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

$$\bar{\delta_n} = \delta_n - \delta_n\left(\frac{\langle\phi_n-\phi_t\rangle}{\phi_n}\right)^{1/\alpha}, \quad \bar{\delta_t} = \delta_t - \delta_t\left(\frac{\langle\phi_t-\phi_n\rangle}{\phi_t}\right)^{1/\beta} \tag{19}$$

which satisfy the conditions of $T_t(\bar{\delta_n},\Delta_t)=0$ and $T_n(\Delta_n,\bar{\delta_t})=0$, respectively.

The PPR potential and its gradients are plotted in Figure 16. The mode I fracture energy and cohesive strength are selected as 352 N/m and 324 MPa, respectively. 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 describe softening in each fracture mode.

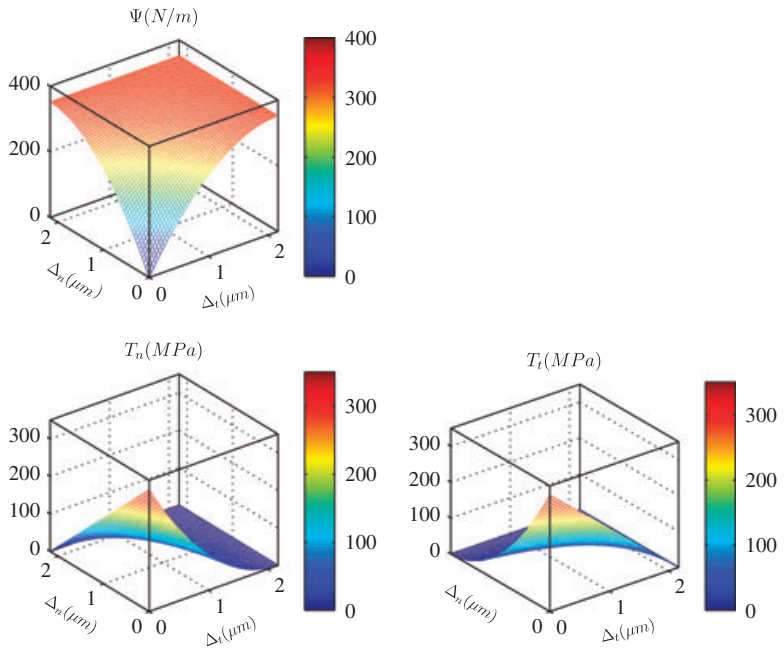


Figure 16. The PPR potential and its gradients.

DOI: 10.1002/nme

The coupled unloading/reloading relationship [55] is employed, which is based on the coupled loading history index, i.e. $\eta(\Delta_n, \Delta_t) = \sqrt{{\Delta_n}^2 + {\Delta_t}^2}$, and the maximum loading history index, i.e. $\eta_{\max} = \sqrt{{\Delta_{n_{\max}}}^2 + {\Delta_{t_{\max}}}^2}$, where $\Delta_{n_{\max}}$ and $\Delta_{t_{\max}}$ are the maximum normal and tangential separations in a loading (or separation) history, respectively. When the loading history index is smaller than the maximum loading history index, the current separation state represents unloading/reloading condition. The cohesive tractions of the coupled unloading/reloading relationship are given as

$$T_n^{\upsilon}(\Delta_n, \Delta_t) = T_n(\Delta_n^{\upsilon}, \Delta_t^{\upsilon}) \left(\frac{\eta}{\eta_{\max}}\right)^{\alpha_{\upsilon}}, \quad T_t^{\upsilon}(\Delta_n, \Delta_t) = T_t(\Delta_n^{\upsilon}, \Delta_t^{\upsilon}) \left(\frac{\eta}{\eta_{\max}}\right)^{\beta_{\upsilon}} \tag{20}$$

where

$$\Delta_n^{\upsilon} = \Delta_n \frac{\eta_{\max}}{\eta}, \quad \Delta_t^{\upsilon} = \Delta_t \frac{\eta_{\max}}{\eta} \tag{21}$$

The unloading/reloading shape parameters (α_{υ}, β_{υ}) change the shape of the traction-separation relation. In this study, the parameters are assumed to be one, which provide linear relation from the origin to unloading/reloading boundaries.

5. NUMERICAL EXAMPLES

The AMR&C schemes are employed for computational simulation of dynamic crack propagation. In this section, a mode I predefined crack path problem is first studied to demonstrate the convergence of the cohesive zone model, and the consistency of the AMR&C schemes. Then, mixed-mode crack propagation and crack branching problems are investigated to validate the computational framework and illustrate the robustness of the AMR&C schemes.

5.1. Predefined crack path: Mode I fracture

The geometry of a single end notched specimen is described in Figure 17, and the initial strain (ε_0) of 0.036 is applied to the specimen. A potential crack path is predefined along the horizontal direction, which provides mode I fracture. The elastic modulus is 3.24 GPa, the Poisson ratio is 0.35, and the density is 1190 kg/m^3. For the cohesive fracture parameters, the mode I fracture energy (G_I), the cohesive strength ($\sigma_{\max}$), and the shape parameter (α) are 352 N/m, 324 MPa, and 2, respectively. In the absence of further experimental data, the mode II fracture parameters are assumed to be the same as the mode I fracture parameters.

The extrinsic cohesive zone model is employed with a predefined crack path, and finite element meshes are generated by a **4k** structured mesh, shown in Figure 18(a). Notice that Zhang and Paulino [20] investigated the predefined crack path problem with the intrinsic cohesive zone model

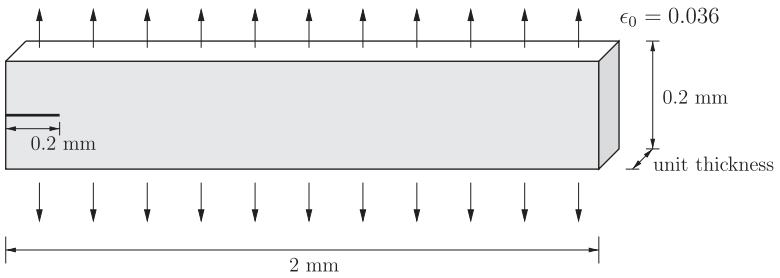


Figure 17. Schematics of geometry and boundary condition.

DOI: 10.1002/nme

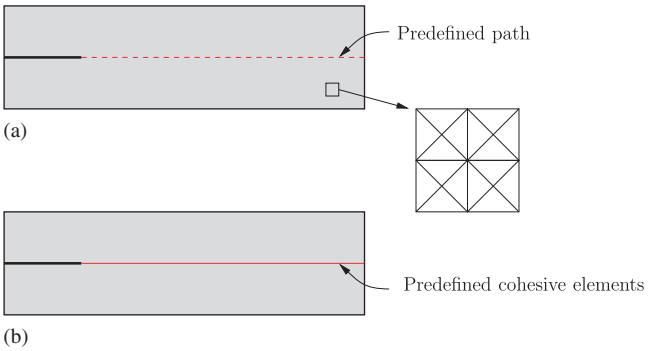


Figure 18. Mode I fracture problem: (a) extrinsic cohesive zone model with predefined path and (b) intrinsic cohesive zone model with predefined cohesive surface elements.

(Figure 18(b)). Because of the discontinuous nature of the extrinsic cohesive zone model, convergence analysis is first performed for uniformly refined meshes. Then, the computational results using AMR&C are compared with the results using a uniformly refined mesh.

5.1.1. Convergence analysis. Convergence of computational results with respect to the time increments and element sizes is investigated. For convergence analysis of time increments (Δt), two finite element meshes are generated within the domain. One mesh has a 100×10 element grid with an element size of 20 μm, and the other has a 200×20 element grid with an element size of 10 μm. Four time increments ($\Delta t = 1 \times 10^{-3}$, 0.4×10^{-3}, 0.2×10^{-3} and 0.1×10^{-3} μs) are tested for each mesh. The normalized crack tip location versus time is plotted for the element size of 20 μm (Figure 19(a)) and the element size of 10 μm (Figure 19(b)). The decrease in the time increment results in the decrease in the slope of the crack tip location versus time curve, i.e. velocity. The computational result with $\Delta t = 0.2 \times 10^{-3}$ μs is almost the same as the result with $\Delta t = 0.1 \times 10^{-3}$ μs for both finite element meshes. This fact demonstrates that the smaller time step leads to converged results in this problem. Additionally, the element size of 20 μm provides more sensitive results than the element size of 10 μm with respect to the change in the time increments.

In addition, the contribution of the last term in the energy variation expression (13), i.e. $\Delta t^2/2(\beta - \frac{1}{2}\gamma)(\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)^{\mathrm{T}}\mathbf{M}(\ddot{\mathbf{u}}_{n+1} + \ddot{\mathbf{u}}_n)$, is estimated with respect to different time increments. Five time increments ($\Delta t = 1 \times 10^{-3}$, 0.8×10^{-3}, 0.4×10^{-3}, 0.2×10^{-3} and 0.1×10^{-3} μs) are tested with the 200×20 **4k** mesh grid. Figure 20 illustrates that the contribution is not significant when one selects small time step in the time integration.

For convergence analysis of element sizes, five **4k** structured meshes are generated with the mesh grid of 80×8, 100×10, 140×14, 200×20 and 400×40 whose element sizes (h_{el}) are 25, 20, 14, 10, and 5 μm, respectively. Time increment (Δt) is 0.2×10^{-3} μs for all five meshes, which is small enough to obtain converged results. The crack tip location versus time is plotted in Figure 21 for each element size. The decrease in element sizes leads to the increase in the slope, i.e. velocity. When element size is smaller than 14 μm, the change of crack tip velocity is less significant. Notice that this study focuses on the convergence of crack path and crack velocity with respect to mesh refinement.

5.1.2. Adaptive mesh refinement. In order to demonstrate the accuracy of AMR in the extrinsic cohesive zone model, the computational results using AMR are compared with the results using a uniformly refined mesh. A finite element mesh with uniform element size of 5 μm is generated by the 400×40 mesh grid, which provides 64 000 elements and 128 881 nodes. The zoom of the uniform mesh around the crack tip is shown in Figure 22(a). Finite element mesh for AMR simulation is initially generated by the 100×10 mesh grid with the refined elements around a crack tip region. The coarse mesh grid has an element size of 20 μm, while element size at the crack tip region is 5 μm, which corresponds to the element size of a uniform mesh. The zoom of the mesh

DOI: 10.1002/nme

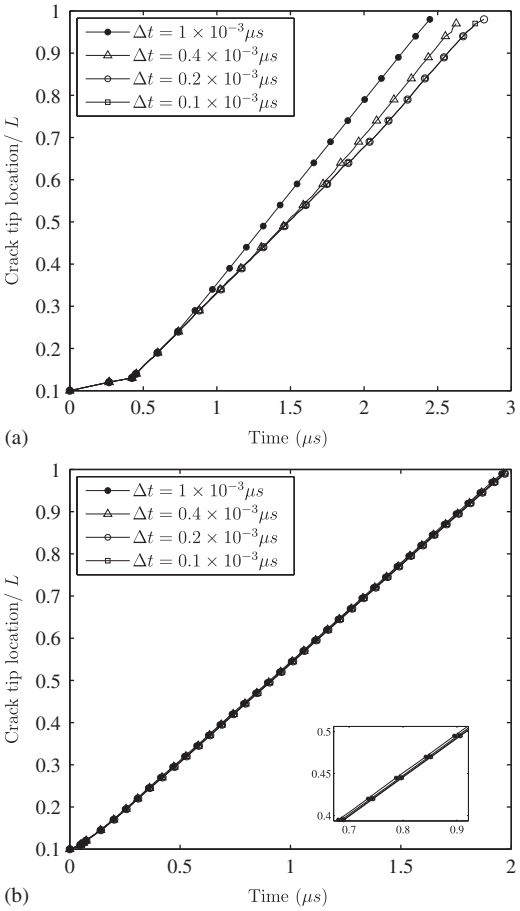


Figure 19. Convergence of crack tip location with respect to the time increment: (a) element size of 20 μm and (b) element size of 10 μm.

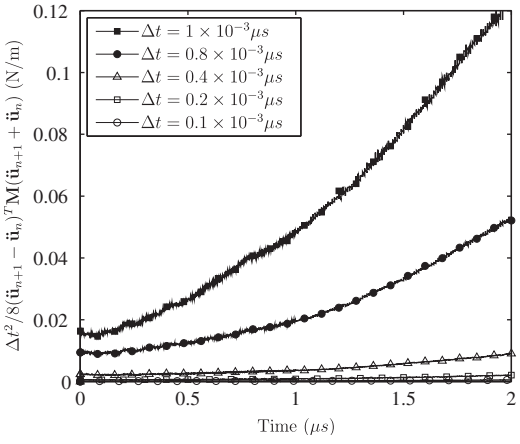


Figure 20. Contribution of the term $\Delta t^2/8(\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)^{\mathrm{T}}\mathbf{M}(\ddot{\mathbf{u}}_{n+1} + \ddot{\mathbf{u}}_n)$ with respect to time increments.

around the crack tip is illustrated in Figure 22(b). The number of elements is 4448, and the number of nodes is 9147 at the initial discretization. Since the AMR approach significantly reduces the number of nodes and the number of elements, the computational cost with the AMR technique is considerably cheaper than standard computational simulations with a uniformly refined mesh.

DOI: 10.1002/nme

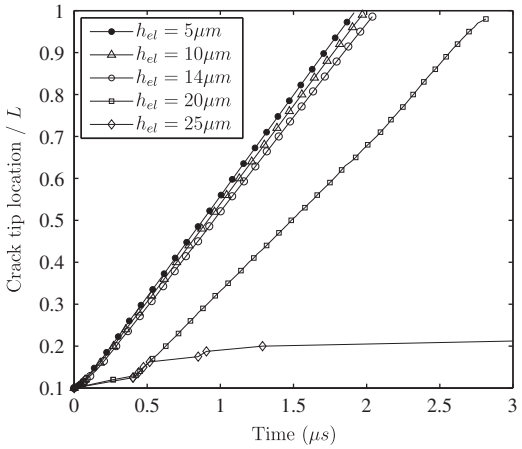


Figure 21. Convergence of crack tip location with respect to the characteristic element size.

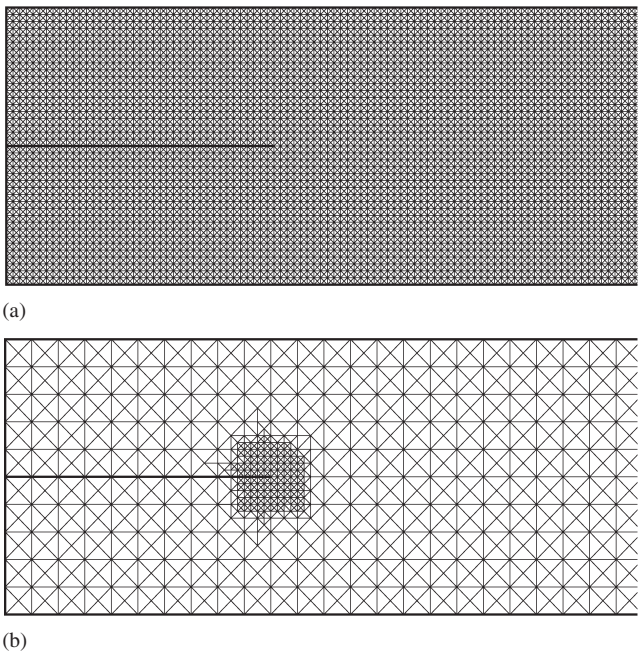


Figure 22. (a) Finite element mesh with uniform element size of 5 μm and (b) finite element mesh with local refinement around crack tip displaying characteristic element size of 5 μm.

While the crack propagates, the crack tip position is updated, and finite element mesh around a crack tip is adaptively refined, as shown in Figure 23(b). Deformed shapes for both finite element meshes at the time (t) 0.2 μs are illustrated in Figure 23. The crack tip position and shape obtained from the AMR technique are almost identical to those resulted from a uniformly refined mesh. Figures 24(a) and (b) illustrate the strain energy at 0.2 μs for the uniform 400×40 mesh grid and the 100×10 mesh grid with AMR, respectively. Both meshes demonstrate the high strain energy around the crack tip, and globally similar strain energy variation. However, the 400×40 mesh grid provides a more detailed description of the strain energy, like the expanding wave from the crack tip, than the 100×10 mesh grid with AMR, as expected. Notice that the most essential part in dynamic crack propagation is to capture non-linear crack tip behavior.

The energy balance is a necessary condition to ensure the numerical stability [56]. The total energy (E_{tot}) consists of the external work (E_{ext}), strain energy (E_{int}), kinetic energy (E_{kin}), and

DOI: 10.1002/nme

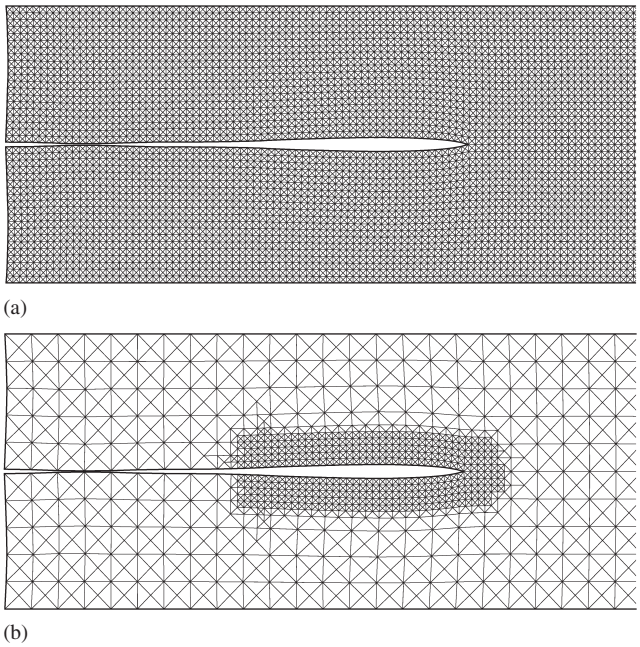


Figure 23. Deformed configuration with finite element mesh at $t = 0.2\,\mu$s (a) uniform 400×40 mesh grid, and (b) 100×10 mesh grid with adaptive local mesh refinement.

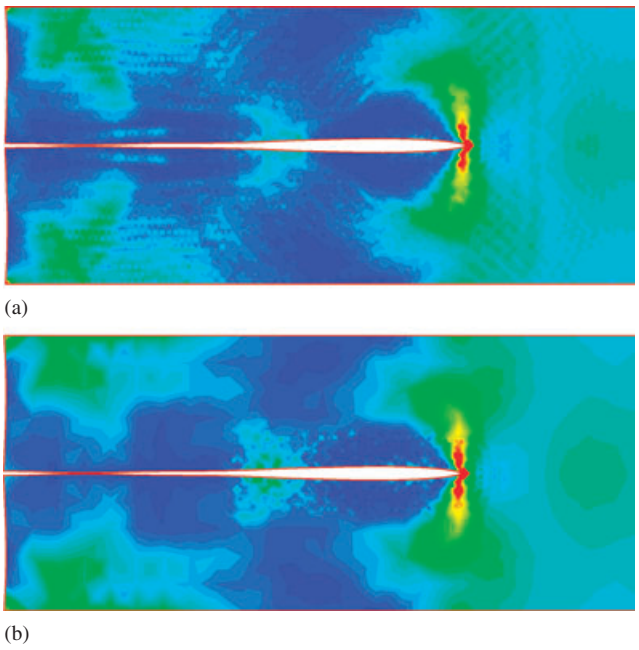


Figure 24. Strain energy at time $t = 0.2\,\mu$s: (a) 400×40 mesh grid and (b) 100×10 mesh grid with AMR.

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 while the kinetic energy and the fracture energy increase. Figure 25 illustrates the energy conservation with respect to time for both the uniform mesh refinement and AMR. The energy evolutions of strain energy, kinetic energy and fracture energy for the uniform mesh refinement are almost the same as the evolutions for AMR; they are nearly on top of each

DOI: 10.1002/nme

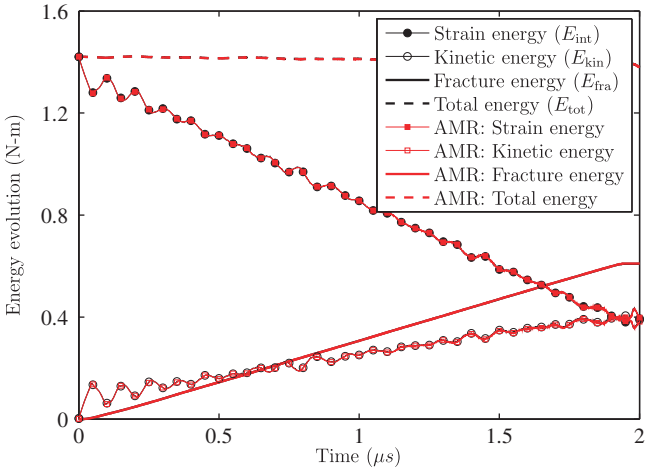


Figure 25. Energy evolution with respect to time for the uniform mesh refinement and AMR.

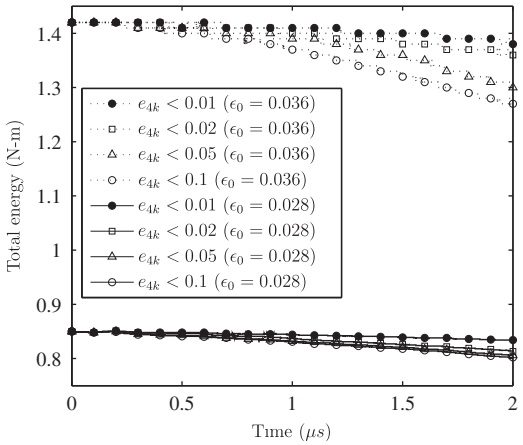


Figure 26. Total energy variation with respect to time for different coarsening error levels of e_{4k}.

other. Therefore, the AMR technique in the extrinsic cohesive zone model provides equivalently accurate computational results compared with the results with the uniformly refined mesh in this example.

5.1.3. Adaptive mesh coarsening. The AMC scheme reduces a solution space, and thus leads to numerical error in finite element analysis. The error due to the adaptive coarsening is investigated on the basis of the energy evolution. While a crack propagates, the crack tip region is adaptively refined, and the outside of the crack tip region is coarsened on the basis of the coarsening criterion, as discussed previously. Finite element size for the coarse mesh grid is 20 μm, while element size for the fine mesh grid is 5 μm. Two sets of finite element analysis are preformed with the initial strain of 0.036 and 0.028. Four error levels (0.01, 0.02, 0.05, and 0.1) are tested for the coarsening criterion. Figure 26 illustrates the total energy evolution with respect to time for different coarsening error levels of e_{4k}. The lower value of the coarsening criterion, which corresponds to tighter criterion for the coarsening, results in less deviation from the constant total energy. Table I demonstrates the numbers of elements and nodes, and the relative error of the total energy ($\Delta E_{\text{tot}}/E_{\text{tot}}$) at the final discretization with respect to different coarsening error levels. A loose mesh coarsening criterion yields fewer nodes and elements (i.e. coarser mesh), which results in a higher relative error at the final discretization.

DOI: 10.1002/nme

Table I. Numbers of nodes and elements, and relative error of the total energy with respect to the coarsening error levels (e_{4k}).

	$\varepsilon_0 = 0.036$			$\varepsilon_0 = 0.028$		
	# of nodes	# of elem	$\Delta E_{\text{tot}}/E_{\text{tot}}$	# of nodes	# of elem	$\Delta E_{\text{tot}}/E_{\text{tot}}$
e_{4k}<0.01	37 530	18 274	0.0282	33 974	16 496	0.0188
e_{4k}<0.02	28 250	13 634	0.0423	19 098	9058	0.0424
e_{4k}<0.05	19 622	9320	0.0845	17 546	8282	0.0518
e_{4k}<0.1	18 904	8556	0.1056	17 542	8280	0.0565

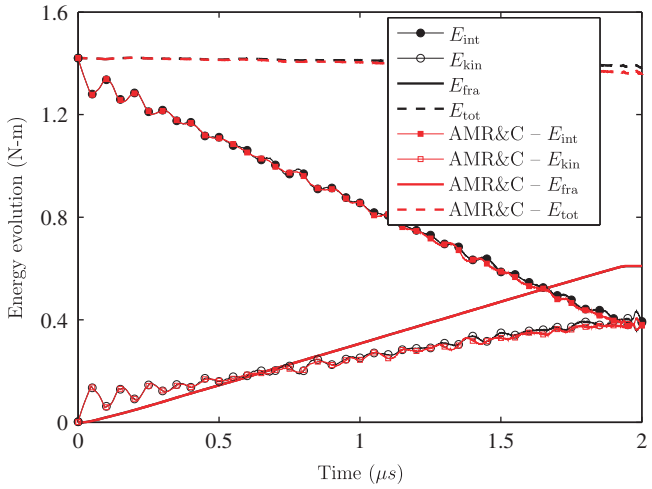


Figure 27. Energy evolution with respect to time considering uniform mesh refinement and AMR&C.

In addition, the energy evolution with AMR&C is compared with the energy evolution with the uniform mesh refinement. The initial strain is 0.036, and **4k** patches are coarsened when the root mean square of strain error (e_{4k}) is less than 0.02. Figure 27 illustrates the good agreement of the energy evolution between AMR&C and the uniform mesh refinement. Although the total energy slightly decreases for the case of AMR&C, the fracture energy (E_{fra}) evolutions are almost identical to each other.

5.2. *Mixed-mode crack propagation*

A doubly notched specimen under impact loading is investigated here. The geometry of the specimen is shown in Figure 28(a), and the impact loading is applied by a projectile. Kalthoff and Winkler [57] tested the specimen to investigate failure mode transition with respect to the loading rates. They observed that relatively lower loading rates resulted in brittle failure with a crack propagation angle of about 70°, whereas higher loading rates generated a shear band ahead of the initial notch with a negative angle of about −10°. This study mainly focuses on brittle fracture behavior.

In the computational simulation, the symmetry condition is utilized, and thus fixed displacement along the vertical direction is imposed at the bottom of the specimen, shown in Figure 28(b). The projectile is assumed to have the same elastic impedance as the doubly notched specimen, and thus one-half of the projectile speed is applied as the impact velocity [58]. The impact velocity (v_0) of 16.54 m/s is applied at the lower left region. Maraging steel 18Ni(300) is used as a representative material property of the doubly notched specimen [20, 28]. The elastic modulus is 190 GPa, the Poisson ratio is 0.3, and the density is 8000 kg/m^3, which lead to the Rayleigh wave speed of 2800 m/s. For mode I fracture parameters, fracture energy and cohesive strength are 22.2 kJ/m^2

DOI: 10.1002/nme

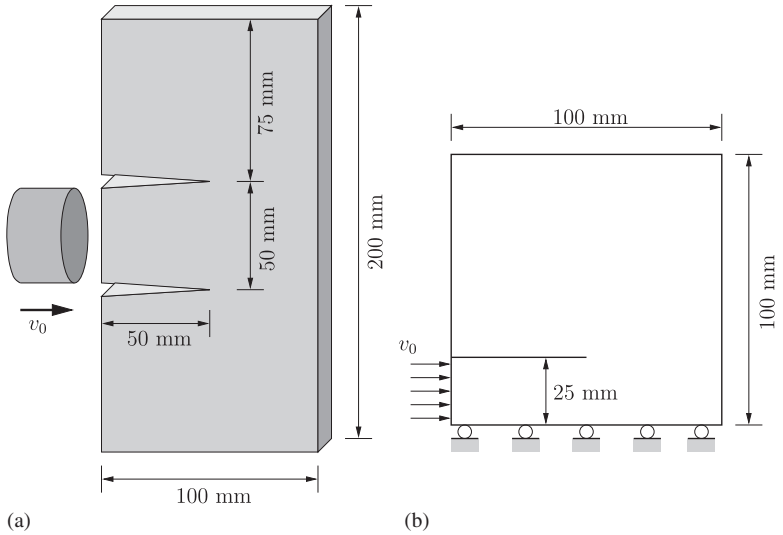


Figure 28. (a) Geometry and boundary condition of a doubly notched specimen and (b) its symmetry domain for finite element analysis.

and 1.733 GPa, respectively. The shape parameter is selected as two, which leads to an almost linear softening behavior. The mode II fracture parameters are assumed to be the same as the mode I fracture parameters. In addition, the nodal perturbation and the edge-swap operators are adaptively utilized in order to improve the crack path in this mixed-mode crack propagation problem [39].

In the analysis of AMR&C, the finite element mesh is initially discretized as shown in Figure 29(a). The mesh has 20×20 coarse mesh grid with an element size of 5 mm, and is refined around a crack tip with an element size of 1.25 mm. The numbers of elements and nodes are 4191 and 5351, respectively, at the initial discretization. A cohesive surface element is first inserted at time 25.4 μs, and the complete separation of the cohesive element occurs at time 26.5 μs. The finite element mesh discretization at time 55 and 77.5 μs is illustrated in Figures 29(b) and (c), respectively. The crack tip region is adaptively refined, and the far field from the crack tip is coarsened. During this dynamic crack propagation, nodal perturbation (NP $=$ 0.2) is adaptively employed around the crack tip region, and 28 edge-swap operations are performed when and where they are needed. More **4k** patches on the left-hand side of the main crack are coarsened than on the right-hand side, because the right-hand side of the crack provides a more complicated displacement field. The strain energy density at time 77.5 μs is shown in Figure 29(d), and it clearly illustrates that the right-hand side has higher strain energy than the left-hand side. The initiation angle is approximately 62°, which is obtained by linear regression. In addition, the computational results with the symmetric boundary condition are consistent with the results without employing the symmetric boundary condition [55], because the interaction between two crack tip regions is not significant. Moreover, these computational results agree well with the previous computational results [20, 28, 59]. Belytschko *et al.* [28] simulated the brittle fracture behavior based on loss of hyperbolicity criterion in conjunction with discontinuous enrichment functions (i.e. X-FEM). Zhang and Paulino [20] utilized an intrinsic cohesive zone model for dynamic crack propagation, and further investigated the effect of material gradation. Song and Belytschko [59] introduced a cracking node method for this problem.

The finite element analysis of AMR&C is compared with the analysis of the uniform mesh refinement and of AMR. Finite element meshes for the uniform mesh refinement and AMR at time $t = 77.5\,\mu$s are shown in Figures 30(a) and (b), respectively. The uniform **4k** mesh has 80×80 mesh grid, and element size is 1.25 mm. The numbers of elements and nodes are initially 51 601 and 25 600, respectively, which are almost 10 times higher than the numbers for AMR&C. The analysis of AMR has the same initial discretization as the analysis of AMR&C. The computational

DOI: 10.1002/nme

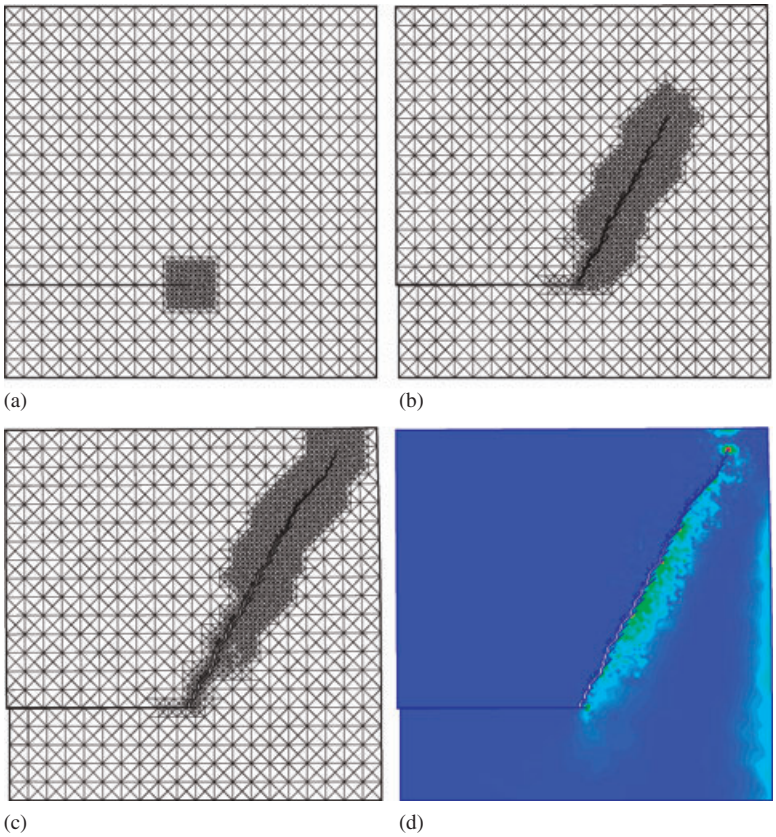


Figure 29. Finite element discretization for AMR&C: (a) at the initial time step; (b) at time $t = 55\,\mu$s; (c) at time $t = 77.5\,\mu$s; and (d) strain energy density at time $t = 77.5\,\mu$s.

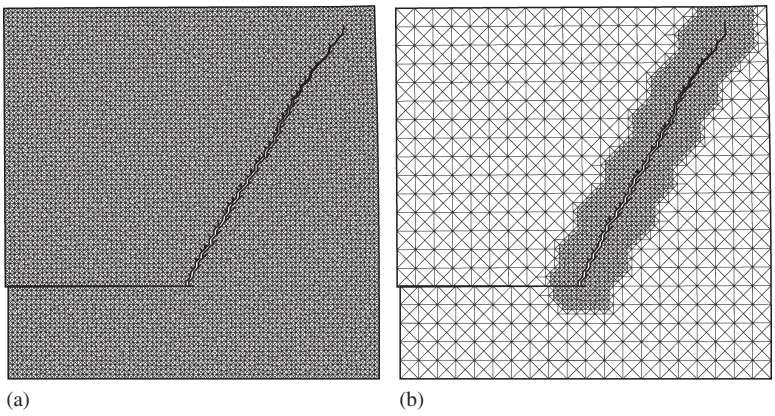


Figure 30. Finite element discretization at time $t = 77.5\,\mu$s considering (a) uniform mesh refinement and (b) AMR.

time is listed in Table II. Computation using AMR&C is approximately 10 times faster than computation using the uniform mesh, whereas computation using AMR is almost seven times faster than computation using the uniform mesh. The crack path for each analysis is described in Figure 31(a), and the overall crack paths correspond well with each other. The crack velocity is estimated by the linear fitting of three adjacent crack tip points with respect to time. The crack velocities for the uniform mesh refinement, AMR, and AMR&C illustrate similar trends, as shown in

DOI: 10.1002/nme

Table II. Computational cost comparison for the mixed-mode crack propagation.

	Uniform mesh	AMR	AMR&C
Computational time (min)	256	36	25

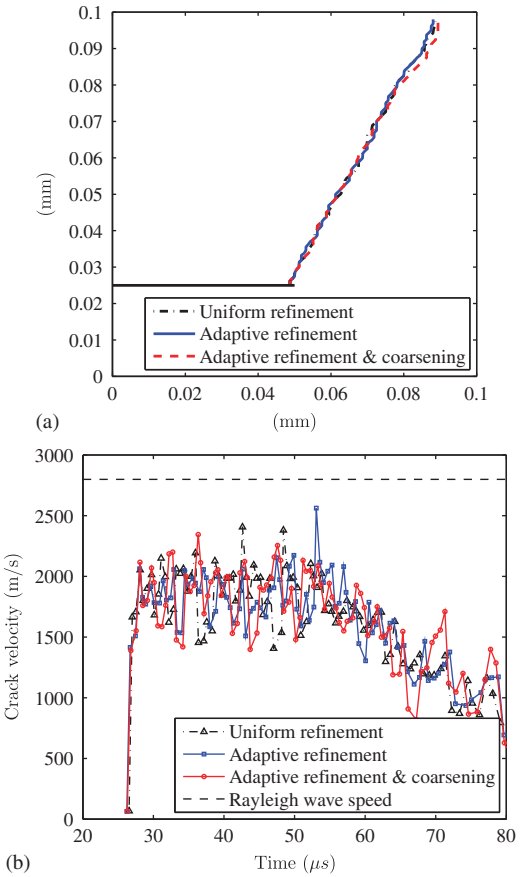


Figure 31. (a) Crack path and (b) crack velocity for the uniform mesh refinement, AMR, and AMR&C.

Figure 31(b). The crack velocity steeply increases at the crack initiation, and gradually decreases around time $t = 50\,\mu$s. The average crack velocity is about 2000 m/s, which is 71% of the Rayleigh wave speed.

Two different element sizes are compared in the analysis of AMR&C. One finite element mesh has 20×20 coarse mesh grid with an element size of 5 mm, and is locally refined around the crack tip with an element size of 1.25 mm. The other has 40×40 coarse mesh grid with an element size of 2.5 mm, and is locally refined around the crack tip with an element size of 0.625 mm. Figure 32 demonstrates the agreement of the crack path and the velocity for the two sizes of the coarse mesh grid. In the evaluation of the crack velocity, three adjacent points are used for the 20×20 coarse mesh grid whereas six adjacent points are used for the 40×40 coarse mesh grid.

The energy evolutions are estimated for AMR and AMR&C during computational simulation, as shown in Figures 33(a) and (b), respectively. The external energy (E_{ext}) is resulted from the impact velocity, and is converted into the strain energy (E_{int}), the kinetic energy (E_{kin}), and the fracture energy (E_{fra}). The strain energy initially increases because of the impact, and slightly decreases while a crack propagates. Both kinetic and fracture energies increase with respect to time. The summation of E_{int}, E_{kin}, and E_{fra} is almost the same as the external energy for the case of AMR (Figure 33(a)), i.e. energy conservation. On the other hand, AMR&C leads to a little energy

DOI: 10.1002/nme

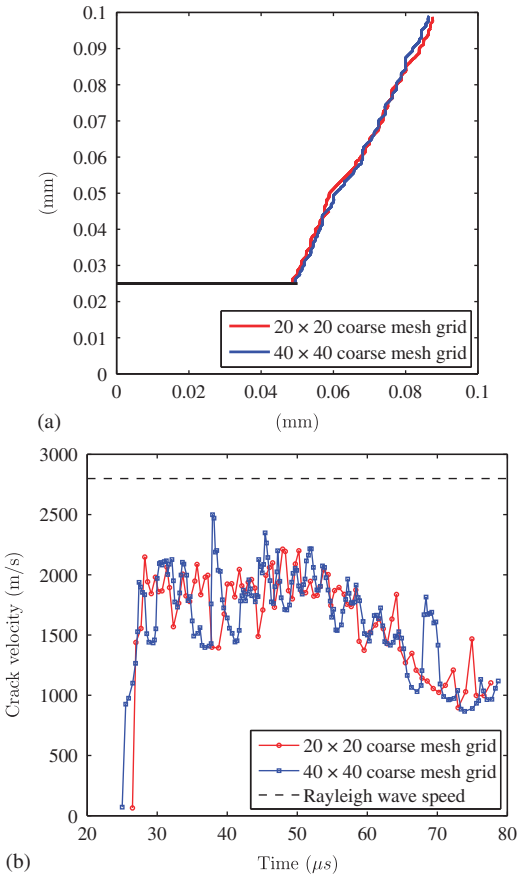


Figure 32. (a) Crack path and (b) crack velocity for different discretizations of the coarse mesh grid.

dissipation, as expected (Figure 33(b)). Notice that such an energy dissipation is also observed in the previous mode I problem (Figure 27).

5.3. Crack branching instability

The specimen geometry for the crack branching problem is illustrated by Figure 34. Notice that uniform traction (15MPa) is applied at the top and bottom edges of the specimen. For elastic properties, the elastic modulus and the Poisson's ratio are 32 GPa and 0.2, respectively, and the density is 2450 kg/m^3, which provide the Rayleigh wave speed of 2125 m/s. For mode I fracture parameters, the fracture energy (ϕ_n) is 3 N/m, the cohesive strength ($\sigma_{\max}$) is 12 MPa, and the shape parameter (α) is selected as two. The mode II parameters are assumed to be the same as the mode I parameters. Notice that Belytschko *et al.* [28] studied this problem in conjunction with a discontinuous shape function (X-FEM), whereas Song and Belytschko [59] simulated this problem by using the cracking node method.

The branching problem is computed by utilizing three different approaches: uniform mesh refinement, AMR, and AMR&C. For the uniform mesh refinement, the domain is discretized with a 160×64 mesh grid which leads to an element size of 0.625 mm. The number of nodes is 82 529, and the number of elements is 40 960 at the initial discretization. For AMR, the domain is discretized with a coarse mesh grid of 40×16 with an element size of 2.5 mm. The crack tip region is refined with an element size of 0.625 mm, which is the same size as the uniformly refined mesh. For AMC, **4k** patches are coarsened when the root mean square of strain error is less than 0.02 for the outside of crack tip regions.

DOI: 10.1002/nme

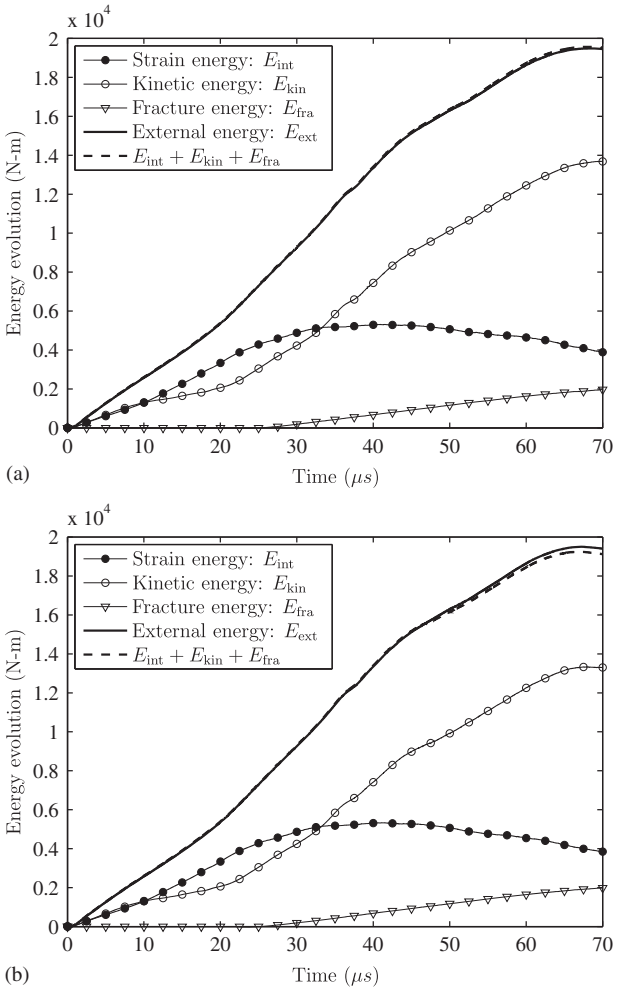


Figure 33. Energy evolution with respect to time for (a) AMR and (b) AMR&C.

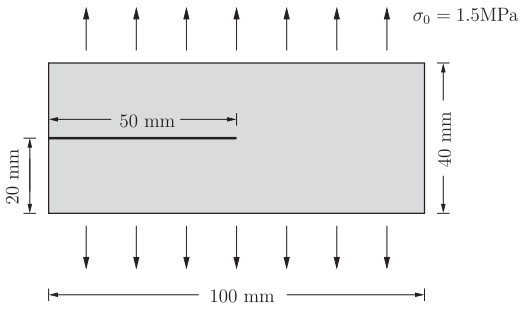


Figure 34. Initial geometry for a branching crack problem.

For the three different approaches, the crack patterns at the time 60 μs are illustrated in Figure 35. The overall crack patterns are similar to each other. The crack initiation time is 11 μs for all cases. After a crack propagates about 15 ∼ 18 mm at the corresponding time of 27 ∼ 30 μs, major branching occurs. In addition, the numbers of nodes in Figures 35(a)–(c) are 83 191, 24 559, and 16 867, and the numbers of volumetric elements are 40 960, 11 845, and 7988, respectively. The computational time for each simulation is provided in Table III. Computation using AMR&C is

DOI: 10.1002/nme

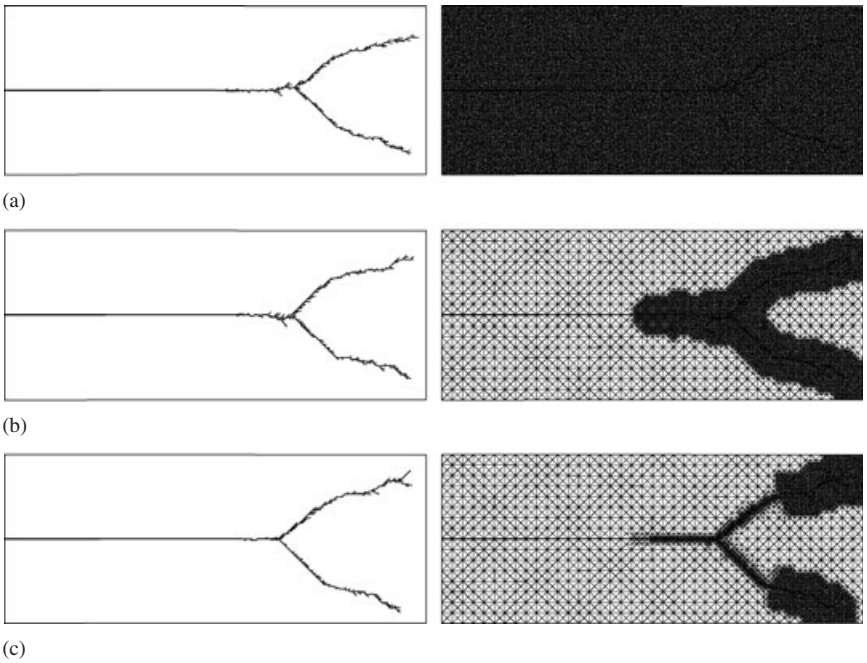


Figure 35. Comparison of crack patterns and finite element meshes for: (a) uniformly refined mesh; (b) AMR; and (c) AMR&C.

Table III. Computational cost comparison for the crack branching problem.

	Uniform mesh	AMR	AMR&C
Computational time (min)	397	62	55

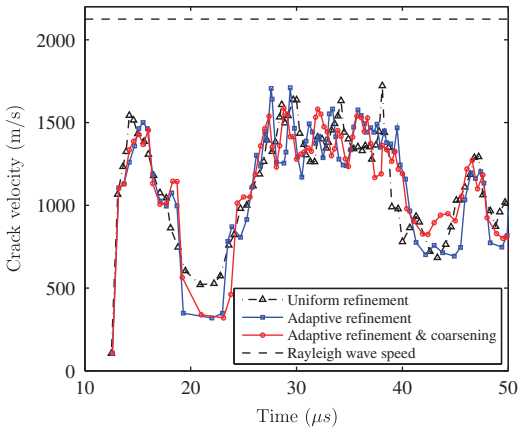


Figure 36. Crack velocity for uniform mesh refinement, AMR, and AMR&C.

approximately seven times faster then computation using the uniform mesh. Therefore, the AMR&C technique is able to significantly reduce the number of degrees of freedom and the computational cost, while providing an almost equivalent computational result to that of the uniformly refined mesh technique.

In addition, Figure 36 plots the crack velocity for the three approaches, and all three results agree with each other. The crack velocity initially increases up to 1500 m/s (60% of the Rayleigh wave speed) until 15 μs. Then, the velocity decreases from 1500 m/s to 500 m/s, and increases to

DOI: 10.1002/nme

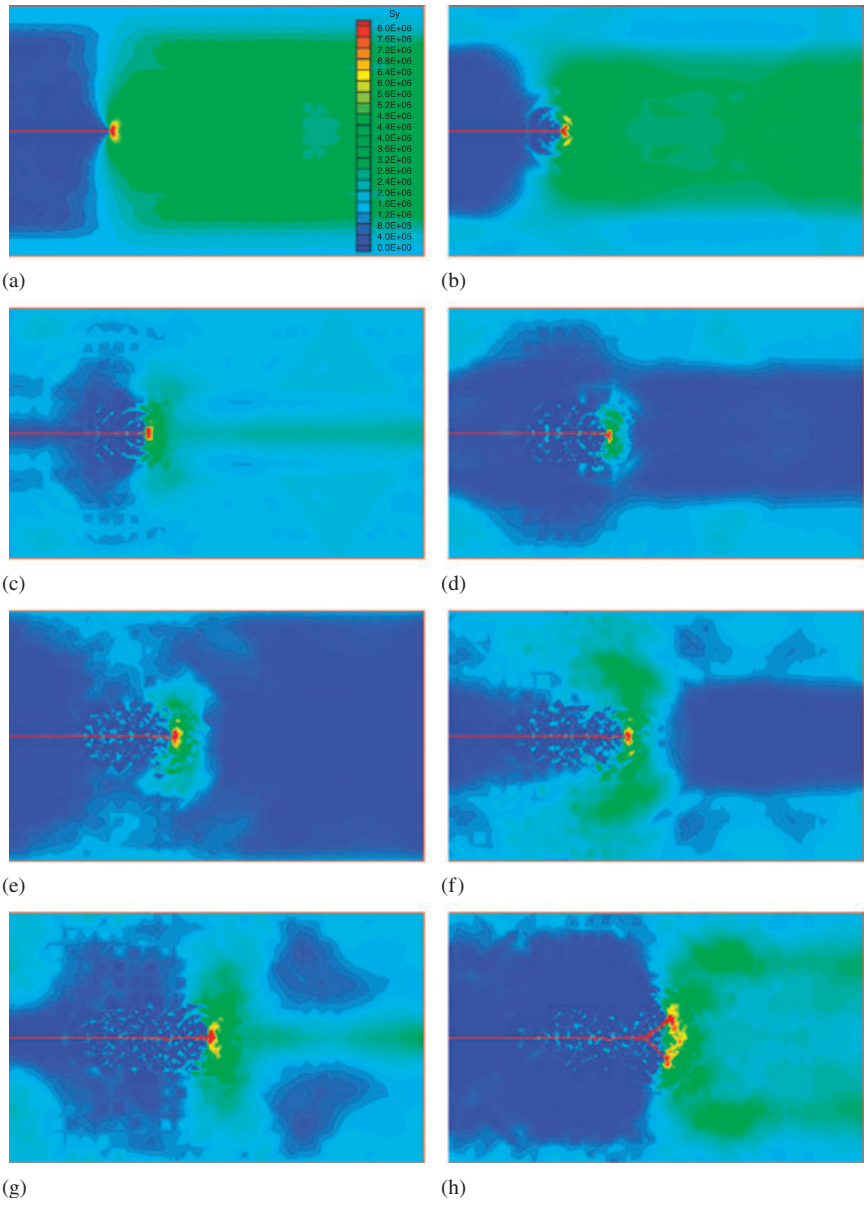


Figure 37. Vertical stress distribution with respect to time: (a) 10 μs; (b) 13 μs; (c) 16 μs; (d) 19 μs; (e) 22 μs; (f) 25 μs; (g) 28 μs; and (h) 31 μs.

1600 m/s (65% of the Rayleigh wave speed) until 27 μs, which corresponds to the time of the major branching initiation. Such a velocity change is correlated with vertical stress wave propagation. Vertical stress with respect to time is illustrated in Figure 37 for the computation with AMR&C. Because of the applied traction (σ_0), the tension wave propagates along the vertical direction, and stress is concentrated at the crack tip before the crack initiation (Figure 37(a)). While a crack propagates, vertical stress is released from the top and bottom edges of the specimen (Figure 37(b)). Then, vertical stress ahead of the crack tip region decreases, as shown in Figures 37(c) and (d), which correspond to the time when the crack velocity decreases. Vertical stress ahead of the crack tip region increases (Figures 37(e)–(g)) with respect to the increase in the crack velocity, and the major branching occurs, as shown in Figure 37(h).

Figure 38 illustrates crack patterns at different times (20, 40, and 60 μs), with corresponding finite element mesh. While a crack propagates, the crack tip region is adaptively refined (Figure 38(a)).

DOI: 10.1002/nme

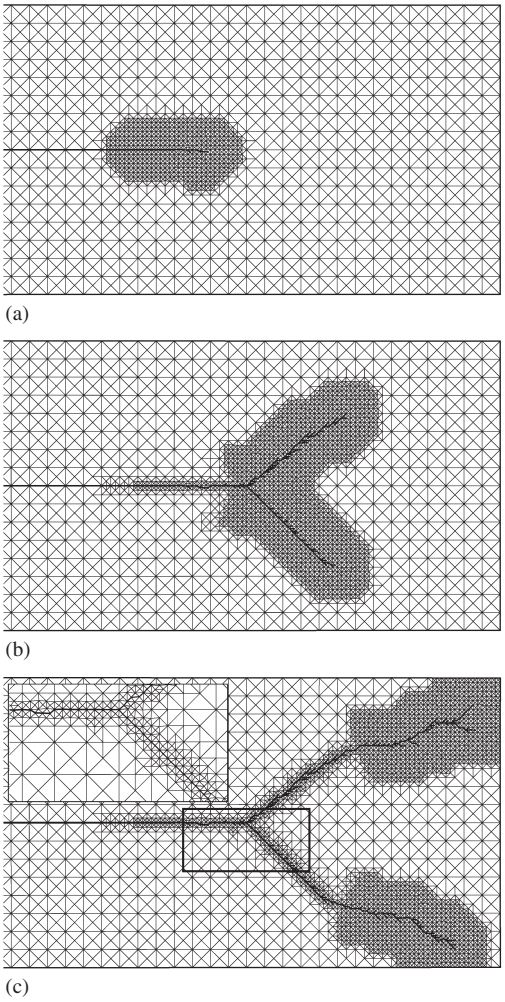


Figure 38. Crack branching at different time steps: (a) 20 μs; (b) 40 μs; and (c) 60 μs.

When branching occurs, two crack tip regions are refined and the outside of the two crack tip regions are coarsened on the basis of the coarsening criterion. The zoom of the finite element mesh at time 60 μs is given in the inset in Figure 38(c). One can observe that local nodes are perturbed with the perturbation factor of 0.2, and that the edge-swap operators are utilized when they are needed.

6. CONCLUDING REMARKS

The AMR&C schemes are systematically developed for solving dynamic cohesive fracture problems. The AMR strategy is based on a sequence of edge-split operators while AMC consists of a sequence of vertex-removal (or edge-collapse) operators in a **4k** mesh. Edge depths are recorded during edge-split operators so that one efficiently removes vertices during mesh coarsening. A sequence of edge splits leads to new nodes, and nodal quantities of the new nodes are interpolated using Lagrange basis shape functions. When the same shape functions (i.e. six Lagrange basis shape functions) are used for the edge split of linear strain triangular elements (i.e. T6 elements), the strain energy is conserved, and the kinetic energy is also conserved provided the consistent mass matrix is utilized. Note that six Lagrange basis shape functions are utilized for boundary edge splits, whereas nine Lagrange basis shape functions are used for interior edge splits so that one generally obtains

DOI: 10.1002/nme

better approximation for the interpolation of nodal quantities. The mesh coarsening is performed on the basis of *a posteriori* coarsening criterion, i.e. the root mean square of strain error. Local nodal displacements are updated to minimize the difference in the strain energy between refined mesh and coarsened mesh, and the updated nodal displacements are utilized to evaluate nodal acceleration and velocity in the coarsened discretization.

Three numerical examples are investigated: a mode I predefined crack path, mixed-mode crack propagation, and crack branching instability. The computational results of AMR&C are consistent with the results of the uniform mesh refinement regarding overall crack patterns, crack velocities, and energy evolutions. The total energy is conserved for AMR, while it slightly decreases for AMC because the mesh coarsening reduces a solution space. However, when the coarsening is performed outside a crack tip region in conjunction with the coarsening criterion, the coarsening does not significantly influence the crack tip behavior. In comparison with standard uniform mesh approaches, the AMR&C strategy significantly reduces computational cost. A multiscale effect is exhibited in the mixed-mode fracture simulation when microbranches form along the main crack. The formation of microbranches leads to a lower main crack velocity, which is closer to experimental observations. This computational framework can be utilized for multiscale computation by introducing several levels of refinement.

APPENDIX A

Support for mesh adaptivity is provided by the TopS data structure API, which consists of a set of functions to create, modify, and destroy meshes. In TopS, meshes are represented by a model object, named `TopModel`. The C language interface of the main functions used for **4k** meshes adaptation in this work is summarized below:

```
TopRefinement4K* topCreateRefinement4K(TopModel* model);
void topDestroyRefinement4K(TopRefinement4K* r);
void topRefinement4K_RefineInRegion(TopRefinement4K* r,
     double x0, double y0, double z0, double radius, int maxdepth);
void topRefinement4K_CoarsenOutRegion(TopRefinement4K* r,
     double x0, double y0, double z0, double radius);
```

The functions `topCreateRefinement4K` and `topDestroyRefinement4K`, respectively, create and destroy an object that is responsible for managing the execution of **4k** adaptive mesh procedures. This object records the history of previous edge splits so that the coarsening procedure is able to undo them. The function `topRefinement4K_RefineInRegion` splits the edges within a circular region of the mesh up to a maximum refinement level. Likewise, the function `topRefinement4K_CoarsenOutRegion` coarsens the mesh outside a circular region.

Whenever mesh modifications occur, the client application needs to transfer or update attributes attached to mesh entities, such as elements and nodes. Many functions of the TopS API directly modify the mesh, and thus the attributes to be updated are known by the application. However, others may need to perform a complex or variable number of indirect modifications. Hence, it may become difficult for the application to discover which attributes have changed so that they can be correctly updated. This is the case of the refinement and coarsening procedures described in this paper.

In order to notify the client application of indirect mesh modification events, we use a callback mechanism, which consists of user-defined functions that can be registered to the FEM model using the TopS API. Therefore, whenever an event for which a callback function is registered happens, the TopS API notifies the application by calling that function.

The prototypes of the callbacks used in this work to update element and nodal attributes are presented below. Every callback receives as parameters a pointer to the model object (`m`) an optional user data (`data`):

```
typedef void (*TopSplitElemCb)(TopModel* m,
  TopElement old_el, TopElement new_el[2], void* data);
typedef void (*TopMergeElemCb)(TopModel* m,
  TopElement old_el[2], TopElement new_el, void* data);
typedef void (*TopInsertNodeCb)(TopModel* m,
  TopNode node, void* data);
```


DOI: 10.1002/nme

```
typedef void (*TopRemoveNodeCb)(TopModel* m,
  TopNode node, void* data);
typedef void (*TopDuplicateNodeCb)(TopModel* m,
  TopNode old_node, TopNode new_node, void* data);
typedef void (*TopSwapElemCb)(TopModel* m,
  TopElement old_el[2], TopElement new_el[2], void* data);
```

The `TopSplitElemCb` is called whenever an element is divided during an edge split operation. It receives as parameters the old element (`old_el`) and the two resulting new elements (`new_el`). The `TopMergeElemCb` is called whenever two elements have been merged in a vertex removal operation. Its parameters are the two old elements (`old_el`) and the resulting element (`new_el`). The `TopInsertNodeCb` and `TopRemoveNodeCb` are called when a node is, respectively, created or removed during an edge split or vertex removal operation. The `TopDuplicateNodeCb` is called when a node is duplicated as a result of the insertion of a new cohesive element into the model. This function is called instead of `TopInsertNodeCb`, so that the application is able to transfer nodal attributes. The parameters of the function are the previous node (`old_node`) and the newly created one (`new_node`). The `TopSwapElemCb` is called whenever an edge swap occurs. In this case, two elements are replaced by two new ones. The function receives as parameters the old elements (`old_el`) and the new ones (`new_el`).

The functions for registering the above callbacks to the model are presented below. For each callback function, an optional pointer to user data may be provided by the application. This pointer will be passed to the callback function when it is called.

```
void topModel_SetSplitElemCb(TopModel* m, TopSplitElemCb cb, void* data);
void topModel_SetMergeElemCb(TopModel* m, TopMergeElemCb cb, void* data);
void topModel_SetInsertNodeCb(TopModel* m, TopInsertNodeCb cb, void* data);
void topModel_SetRemoveNodeCb(TopModel* m, TopRemoveNodeCb cb, void* data);
void topModel_SetDuplicateNodeCb(TopModel* m, TopDuplicateNodeCb cb, void* data);
void topModel_SetSwapElemCb(TopModel* m, TopSwapElemCb cb, void* data);
```

REFERENCES

1. Babuska I, Rheinboldt WC. A-*posteriori* error estimates for the finite element method. *International Journal for Numerical Methods in Engineering* 1978; **12**(10):1597–1615.
2. Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering* 1987; **24**(2):337–357.
3. Ainsworth M, Oden JT. A *posteriori* error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering* 1997; **142**(1–2):1–88.
4. Paulino GH, Menezes IFM, Cavalcante Neto JB, Martha LF. A methodology for adaptive finite element analysis: towards an integrated computational environment. *Computational Mechanics* 1999; **23**(5):361–388.
5. Swenson DV, Ingraffea AR. Modeling mixed-mode dynamic crack propagation using finite elements: theory and applications. *Computational Mechanics* 1988; **3**(6):381–397.
6. Wawrzynek PA, Ingraffea AR. Discrete modeling of crack propagation: theoretical aspects and implementation issues in two and three dimensions. *Technical Report 91-5*, School of Civil Engineering and Environmental Engineering, Cornell University, 1991.
7. Kim JH, Paulino GH. Simulation of crack propagation in functionally graded materials under mixed-mode and non-proportional loading. *International Journal of Mechanics and Materials in Design* 2004; **1**(1):63–94.
8. Wiberg NE, Zeng L, Li X. Error estimation and adaptivity in elastodynamics. *Computer Methods in Applied Mechanics and Engineering* 1992; **101**(1–3):369–395.
9. Belytschko T, Tabbara M. H-adaptive finite element methods for dynamic problems, with emphasis on localization. *International Journal for Numerical Methods in Engineering* 1993; **36**(24):4245–4265.
10. Molinari JF, Ortiz M. Three-dimensional adaptive meshing by subdivision and edge-collapse in finite-deformation dynamic-plasticity problems with application to adiabatic shear banding. *International Journal for Numerical Methods in Engineering* 2002; **53**(5):1101–1126.
11. Rivara MC. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. *International Journal for Numerical Methods in Engineering* 1997; **40**(18):3313–3324.
12. Cavin P, Gravouil A, Lubrecht AA, Combescure A. Efficient FEM calculation with predefined precision through automatic grid refinement. *Finite Elements in Analysis and Design* 2005; **41**(11–12):1043–1055.
13. Cavin P, Gravouil A, Lubrecht AA, Combescure A. Automatic energy conserving space–time refinement for linear dynamic structural problems. *International Journal for Numerical Methods in Engineering* 2005; **64**(3):304–321.
14. Bryan GL. Fluids in the universe: adaptive mesh refinement in cosmology. *Computing in Science and Engineering* 1999; **1**(2):46–53.

DOI: 10.1002/nme

15. Trangenstein JA. Multi-scale iterative techniques and adaptive mesh refinement for flow in porous media. *Advances in Water Resources* 2002; **25**(8–12):1175–1213.
16. Hughes TJR. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**(1–4):387–401.
17. Fish J, Yuan Z. Multiscale enrichment based on partition of unity. *International Journal for Numerical Methods in Engineering* 2005; **62**(10):1341–1359.
18. Hou TY, Wu XH. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics* 1997; **134**(1):169–189.
19. 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.
20. 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.
21. Camacho GT, Ortiz M. Computational modelling of impact damage in brittle materials. *International Journal of Solids and Structures* 1996; **33**(20–22):2899–2938.
22. 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.
23. Huang H, Costanzo F. On the use of space–time finite elements in the solution of elasto-dynamic fracture problems. *International Journal of Fracture* 2004; **127**(2):119–146.
24. 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.
25. Linder C, Armero F. Finite elements with embedded branching. *Finite Elements in Analysis and Design* 2009; **45**(4):280–293.
26. 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.
27. Moes N, Belytschko T. Extended finite element method for cohesive crack growth. *Engineering Fracture Mechanics* 2002; **69**(7):813–833.
28. 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.
29. 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.
30. Oden JT, Duarte CA. Clouds cracks and FEM's. In *Recent Developments in Computational and Applied Mechanics*, Reddy BD (ed.). International Center for Numerical Methods in Engineering: Barcelona, Spain, 1997; 302–321.
31. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 1999; **45**(5):601–620.
32. 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.
33. Fries TP, Zilian A. On time integration in the XFEM. *International Journal for Numerical Methods in Engineering* 2009; **79**(1):69–93.
34. Remmers JJC, de Borst R, Needleman A. The simulation of dynamic crack propagation using the cohesive segments method. *Journal of the Mechanics and Physics of Solids* 2008; **56**(1):70–92.
35. Bishop JE. Simulating the pervasive fracture of materials and structures using randomly close packed Voronoi tessellations. *Computational Mechanics* 2009; **44**(4):455–471.
36. 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.
37. 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.
38. 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.
39. Paulino GH, Park K, Celes W, Espinha R. Adaptive dynamic cohesive fracture simulation using edge-swap and nodal perturbation operators. *International Journal for Numerical Methods in Engineering* 2010; **84**(11):1303–1343.
40. 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.
41. Velho L, Gomes J. Variable resolution 4-k meshes: concepts and applications. *Computer Graphics Forum* 2000; **19**(4):195–212.
42. Rivara MC. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *International Journal for Numerical Methods in Engineering* 1984; **20**(4):745–756.
43. Zienkiewicz OC, Zhu JZ. Superconvergent patch recovery and a posteriori error estimates. Part 1: the recovery technique. *International Journal for Numerical Methods in Engineering* 1992; **33**(7):1331–1364.
44. Wiberg NE, Li XD. Superconvergent patch recovery of finite-element solution and a posteriori L2 norm error estimate. *Communications in Numerical Methods in Engineering* 1994; **10**(4):313–320.
45. Dunavant DA. High degree efficient symmetrical Gaussian quadrature rules for the triangle. *International Journal for Numerical Methods in Engineering* 1985; **21**(6):1129–1148.

DOI: 10.1002/nme

46. Velho L. Mesh simplification using four-face clusters. *Shape Modeling and Applications, SMI 2001 International Conference*, Genova, Italy, 2001; 200–208.
47. Zienkiewicz OC, Taylor RL, Zhu JZ. *The Finite Element Method: its Basis and Fundamentals*. Butterworth-Heinemann: Oxford, U.K., 2005.
48. Newmark NM. A method of computation for structural dynamics. *Journal of Engineering Mechanics Division* (ASCE) 1959; **85**:67–94.
49. Krenk S. Energy conservation in Newmark based time integration algorithms. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(44–47):6110–6124.
50. Ergodan F, Sih GC. On the crack extension in plates under plane loading and transverse shear. *Journal of Basic Engineering* 1963; **85**:519–527.
51. Sih GC. Strain-energy-density factor applied to mixed mode crack problems. *International Journal of Fracture* 1974; **10**(3):305–321.
52. Hughes TJR. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications: New York, 2000.
53. 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.
54. Elguedj T, Gravouil A, Maigre H. An explicit dynamics extended finite element method. Part 1: mass lumping for arbitrary enrichment functions. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(30–32):2297–2317.
55. Park K. Potential-based fracture mechanics using cohesive zone and virtual internal bond modeling. *Ph.D. Thesis*, University of Illinois at Urbana-Champaign, 2009.
56. Belytschko T, Liu WK, Moran B. *Nonlinear Finite Elements for Continua and Structures*. Wiley: New York, 2000.
57. Kalthoff JF, Winkler S. Failure mode transition at high rates of shear loading. *International Conference on Impact Loading and Dynamic Behavior of Materials*, Bremen, Germany, vol. 1, 1987; 185–195.
58. Lee YJ, Freund LB. Fracture initiation due to asymmetric impact loading of an edge cracked plate. *Transactions of the ASME. Journal of Applied Mechanics* 1990; **57**(1):104–111.
59. 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.

DOI: 10.1002/nme