Extended Finite Element Method on Polygonal and Quadtree Meshes

A. Tabarraei [1], N. Sukumar [2,*]

Department of Civil and Environmental Engineering, One Shields Avenue, University of California, Davis, CA 95616.

Abstract

In this paper, we present mesh-independent modeling of discontinuous fields on polygonal and quadtree finite element meshes. This approach falls within the class of extended and generalized finite element methods, where the partition of unity framework is used to introduce additional (enrichment) functions within the classical displacement-based finite element approximation. For crack modeling, a discontinuous function and the two-dimensional asymptotic crack-tip fields are used as enrichment functions. Linearly complete partition of unity approximations are adopted on polygonal (convex and nonconvex elements) and quadtree meshes. Excellent agreement with reference solution results is obtained for mixed-mode stress intensity factors on benchmark crack problems, and crack growth simulations without remeshing are conducted on polygonal and quadtree meshes to reveal the potential of the proposed techniques in computational failure mechanics.

Key words: partition of unity, discontinuous enrichment, natural neighbors, Voronoi tessellation, nonconvex polygons, fracture, crack growth

1 Introduction

Polygonal finite elements [1] are a generalization of triangular and quadrilateral finite element methods to meshes with n-sided elements ($n \geq 3$). The ability to construct conforming approximations on convex and nonconvex shapes provides greater flexibility in mesh generation [2] and render such elements to

* Corresponding author. Email: nsukumar@ucdavis.edu
[1] Graduate Student
[2] Associate Professor

be a viable and potentially attractive choice in computer modeling and simulation. Recently, partition of unity finite element techniques have emerged as a powerful tool for crack growth simulations without the need for any remeshing. The extended and generalized finite element methods [3–5] are particular instances of the partition of unity finite element method [6, 7]. In the extended finite element method (X-FEM) [3, 4], the emphasis has been on modeling crack discontinuities using minimal enrichment. In this paper, we introduce crack growth modeling using the X-FEM on polygonal and quadtree meshes.

Rational finite element basis functions on convex polygons were introduced by Wachspress [8], and in recent years there has been growing interest in the construction of shape functions (barycentric coordinates) on irregular polygons and polyhedra. Many new contributions on barycentric polygonal interpolation have been realized in geometry modeling and graphics, and in finite element methods [1, 9–16]. In Reference [1], natural neighbor based (Laplace) interpolants [17] are used to construct shape functions on irregular polygons (n-gons), which is adapted to quadtree meshes (resolves the issue of *hanging nodes*) in References [18, 19]. Barycentric coordinates are non-negative, form a partition of unity, have linear precision, and are linear on element edges. Mean value coordinates retain the key properties of barycentric coordinates on arbitrary (convex and nonconvex) planar polygons [15], which makes them a suitable candidate for use in finite element modeling on meshes with nonconvex elements.

The partition of unity (PU) method [6, 7] generalized the classical finite element method by providing the ability to incorporate *a priori* information on the nature of the solution of the boundary-value problem within the finite element approximation space. Any set of basis functions $\{\phi_i(\mathbf{x})\}_{i=1}^n$ that sum to unity form a partition of unity. Belytschko and Black [3] were the first to use the PU framework to model crack discontinuities, and this was subsequently extended by Moës *et al.* [4] who introduced simple (Heaviside function) enrichments for the interior of the crack. The literature on generalized and extended finite element techniques is extensive, and we point the interested reader to Karihaloo and Xiao [20] for a review and to the recent article of Bordas *et al.* [21] for an extensive listing. The driving force in the X-FEM is to obtain good accuracy on relatively coarse meshes and to eliminate the need for remeshing in crack growth simulations—remeshing is a necessity for crack propagation studies using classical finite element methods. In the X-FEM, crack modeling and crack growth are carried out on a fixed finite element mesh, which lends itself to parametric studies for failure analyses of single and multiple cracks. In this paper, we appeal to References [1, 15, 18] to first construct partition of unity approximation spaces on polygonal (convex and nonconvex elements) meshes. These advances are used to develop extended finite element methods for crack growth modeling on polygonal meshes, and as a special case, the capabilities are extended to quadtree meshes. The ad-

vantages and flexibility of using nonconvex finite elements and quadtree-based meshes in mesh-independent crack growth simulations is demonstrated.

The rest of the paper is organized as follows. In Section 2, the construction of conforming interpolants on polygonal and quadtree meshes is described. The X-FEM formulation for crack modeling on polygonal meshes is touched upon in Section 3, and the computer implementation is presented in Section 4. In Section 5, the performance of the proposed technique is assessed on benchmark crack problems and for crack growth simulations in linear elastic media. Finally, we close with some concluding remarks in Section 6.

2 Polygonal and quadtree finite elements

2.1 Polygonal meshes

The construction of conforming Laplace interpolants on arbitrary convex polygons is summarized; for details, the interested reader can refer to References [1,16]. The Laplace interpolant is a natural neighbor-based interpolation scheme [17] that is based on the underlying Voronoi diagram and Delaunay triangulation. For a set $\mathscr{N}$ consisting of n nodes with locations $\{\mathbf{x}_i\}_{i=1}^n$, the *natural neighbors* [22] of a point p within the convex hull of $\mathscr{N}$ are defined through the Delaunay circumcircles: if p lies within the circumcircle of a Delaunay triangle t, the nodes that define t are neighbors of p. For the polygonal reference elements shown in Fig. 1 (regular n-gons), the Delaunay tessellation of either polygonal domain is non-unique. This non-uniqueness arises in two dimensions when four or more nodes are cocircular. For a valid tessellation of either polygon, the Delaunay triangles t of the polygon are circumscribed by the same circle, and therefore all the vertex nodes of the polygon are natural neighbors of any interior point p.

A polygonal reference element and the Voronoi cell of a point p inside the reference element are shown in Fig. 2. The Laplace shape function at point p is given by [17]

$$\phi_a(\boldsymbol{\xi}) = \frac{\alpha_a(\boldsymbol{\xi})}{\sum\limits_{b=1}^{n} \alpha_b(\boldsymbol{\xi})}, \quad \alpha_a(\boldsymbol{\xi}) = \frac{s_a(\boldsymbol{\xi})}{h_a(\boldsymbol{\xi})}, \quad \boldsymbol{\xi} \in \Omega_0, \tag{1}$$

where n is the number sof vertex nodes, $\alpha_a(\boldsymbol{\xi})$ is the Laplace weight function, $s_a(\boldsymbol{\xi})$ is the length of the common edge between the Voronoi cell of point p and Voronoi cell of node a, and $h_a(\boldsymbol{\xi})$ is the distance between point p and node a. By construction, Laplace shape functions are non-negative, form a partition of unity, are linearly complete, and are linear on the element edges [23]. The

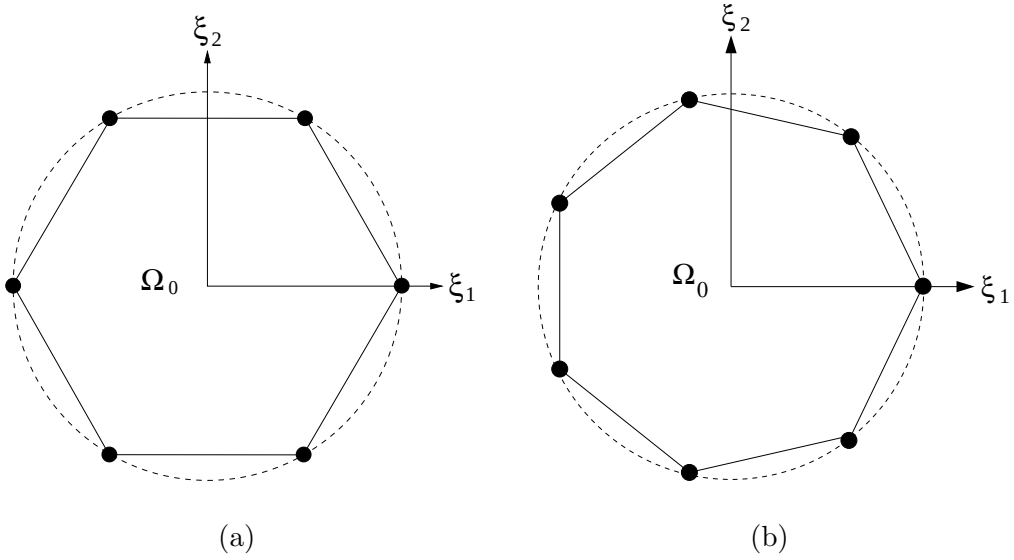


Fig. 1. Reference elements. (a) Hexagon; and (b) Heptagon.

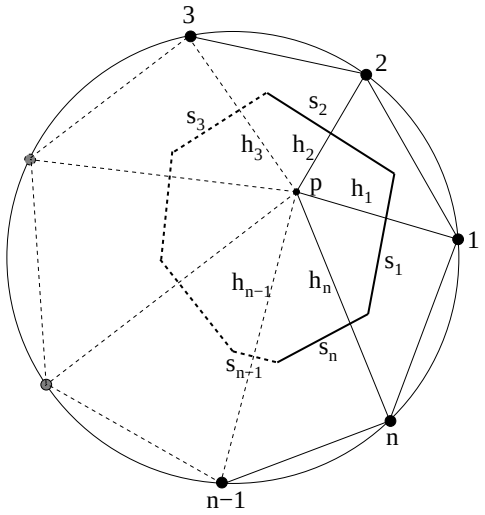


Fig. 2. Construction of Laplace shape functions in a reference element.

Laplace interpolant on the physical element is constructed via an isoparametric mapping from the corresponding reference element (the regular polygon with the same number of nodes as the physical element) to the physical element. In Fig. 3, the isoparametric mapping from a hexagonal reference element to a six-noded polygonal element is illustrated. Since the mapping is isoparametric, Laplace shape functions remain linear on the edges of the physical element. Due to linearity of the Laplace interpolant on the element edges, the resulting basis functions are conforming on polygonal meshes. Laplace shape functions reduce to barycentric coordinates on a triangle and to bilinear finite element shape functions on a biunit square [23]. Hence, polygonal finite elements are an extension of three- and four-noded finite elements to irregular convex n-gons. To integrate a function on a polygonal element, we proceed similar to

finite elements. Numerical integration is performed on the reference element by subdividing it into subtriangles and then quadrature rules for a triangle are applied [1].

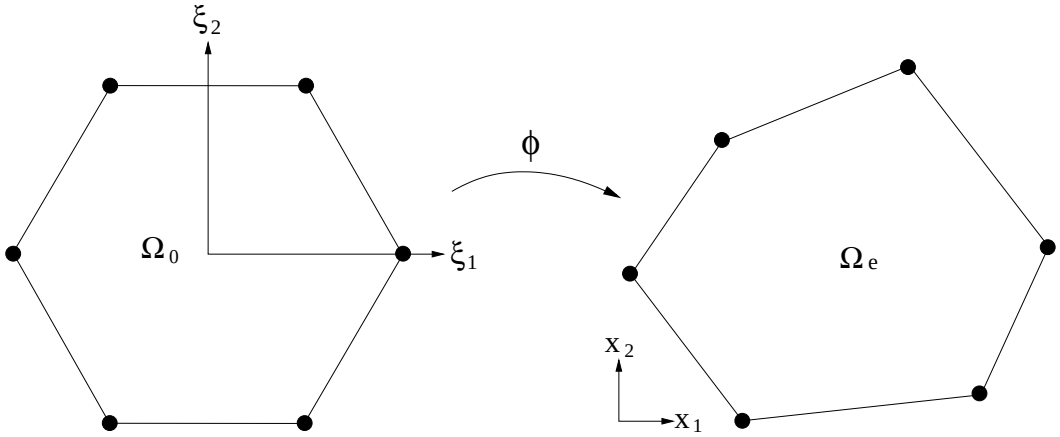


Fig. 3. Isoparametric mapping from a hexagonal reference element to a six-noded physical element.

Meyer *et al.* [9] introduced a simple formula for Wachspress's basis functions, whereas Floater [10] utilized the mean value theorem for harmonic functions to propose a new barycentric coordinate on irregular polygons. Floater's linearly precise mean value coordinate at point p is [10]:

$$\phi_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum\limits_{j=1}^{n} w_j(\mathbf{x})}, \quad w_i(\mathbf{x}) := \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{r_i}, \quad r_i := |\mathbf{x}_i - \mathbf{x}|, \tag{2}$$

where $r_i \equiv r_i(\mathbf{x})$ is the Euclidean distance between p and p_i, and the angles α_{i-1} and α_i are shown in Fig. 4b. In Fig. 4b, p_{i-1}, p_i, and p_{i+1} are contiguous vertex nodes whose coordinates are $\mathbf{x}_{i-1}$, $\mathbf{x}_i$, and $\mathbf{x}_{i+1}$, respectively. For nonconvex polygons, mean value coordinates are positive in the kernel of the polygon (Fig. 4a). Furthermore, they have a smooth extension outside the polygon, and are linear on any element edge (Fig. 5). These properties are not met by other barycentric constructions [13,16]. Hormann and Floater [15] have presented the important properties of mean value coordinates along with an implementation for planar polygons. On letting $\mathbf{r}_i = \mathbf{x}_i - \mathbf{x}$, we have [15]

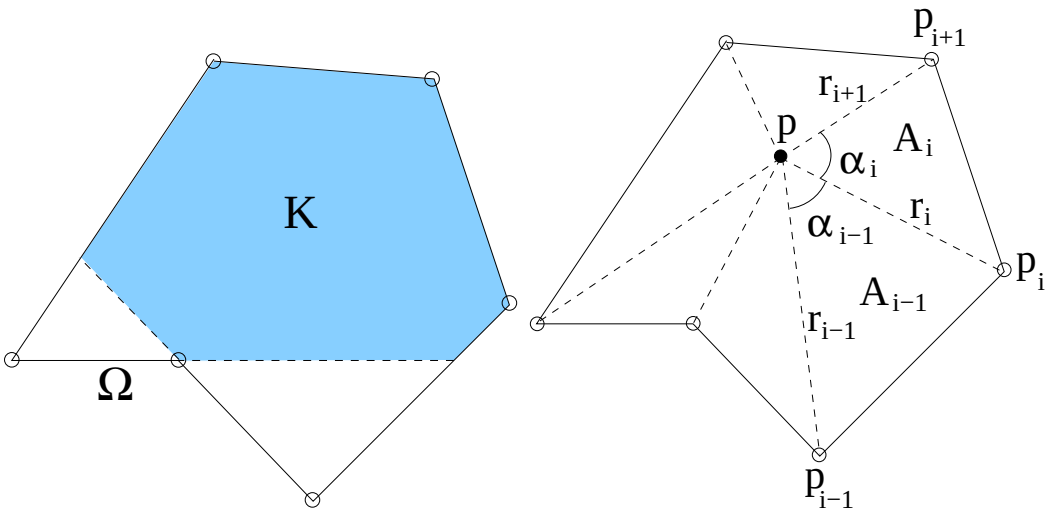
$$\tan\left(\frac{\alpha_i}{2}\right) = \frac{1-\cos\alpha_i}{\sin\alpha_i} = \frac{r_i r_{i+1} - \mathbf{r}_i \cdot \mathbf{r}_{i+1}}{2A_i} = \frac{r_i r_{i+1} - \mathbf{r}_i \cdot \mathbf{r}_{i+1}}{|\mathbf{r}_i \times \mathbf{r}_{i+1}|}, \tag{3}$$

where A_i is the area of triangle $[p_i, p_{i+1}, p]$ (see Fig. 4b). If $A_i = 0$, then two cases are possible: (1) if $p \in [p_i, p_{i+1}]$ (p lies on a boundary edge), then the interpolant is linear on the edge with only $\phi_i(\mathbf{x})$ and $\phi_{i+1}(\mathbf{x})$ being non-zero, and (2) if p lies in the interior of a nonconvex polygon with $\alpha_i = 0$, then the weight is zero but a procedure to find the derivatives of the above weight at p is not apparent. To ease the derivative computations, we have implemented a

modified form of Eq. (3):

$$\tan\left(\frac{\alpha_i}{2}\right) = \frac{\sin\alpha_i}{1+\cos\alpha_i} = \frac{2A_i}{r_i r_{i+1} + \mathbf{r}_i \cdot \mathbf{r}_{i+1}} = \frac{|\mathbf{r}_i \times \mathbf{r}_{i+1}|}{r_i r_{i+1} + \mathbf{r}_i \cdot \mathbf{r}_{i+1}}, \tag{4}$$

which is now valid for all points p that are in the interior of an arbitrary planar (convex and nonconvex) polygon. The denominator vanishes when $\alpha_i = \pi$, i.e., when p lies on the boundary of the polygon, but this does not arise in the present application since all integration points are in the polygon interior.



(a) (b)

Fig. 4. Mean value coordinates [10]. (a) Kernel K of the polygon; and (b) Construction.

To numerically integrate the weak form integrals, the derivatives of the basis functions are required. Unlike Laplace shape functions that are defined on a reference element, mean value coordinates are directly evaluated on the physical element. The physical element is subdivided into triangles for the purpose of numerical integration. On applying the chain rule to Eq. (2) and using Eq. (4), the derivatives of mean value coordinates are obtained. Letting $W(\mathbf{x}) := \sum_{j=1}^{n} w_j(\mathbf{x})$ and $\tilde{w}_i(\mathbf{x}) := \tan(\alpha_i/2)$, we can write

$$\boldsymbol{\nabla}\phi_i(\mathbf{x}) = \frac{\boldsymbol{\nabla} w_i(\mathbf{x}) - \phi_i(\mathbf{x})\boldsymbol{\nabla} W(\mathbf{x})}{W(\mathbf{x})}, \tag{5a}$$

where

$$\boldsymbol{\nabla} w_i(\mathbf{x}) = \frac{\boldsymbol{\nabla}\tilde{w}_{i-1}(\mathbf{x}) + \boldsymbol{\nabla}\tilde{w}_i(\mathbf{x}) - w_i(\mathbf{x})\boldsymbol{\nabla} r_i}{r_i}, \quad \boldsymbol{\nabla} r_i = \frac{\mathbf{x}-\mathbf{x}_i}{r_i}, \tag{5b}$$

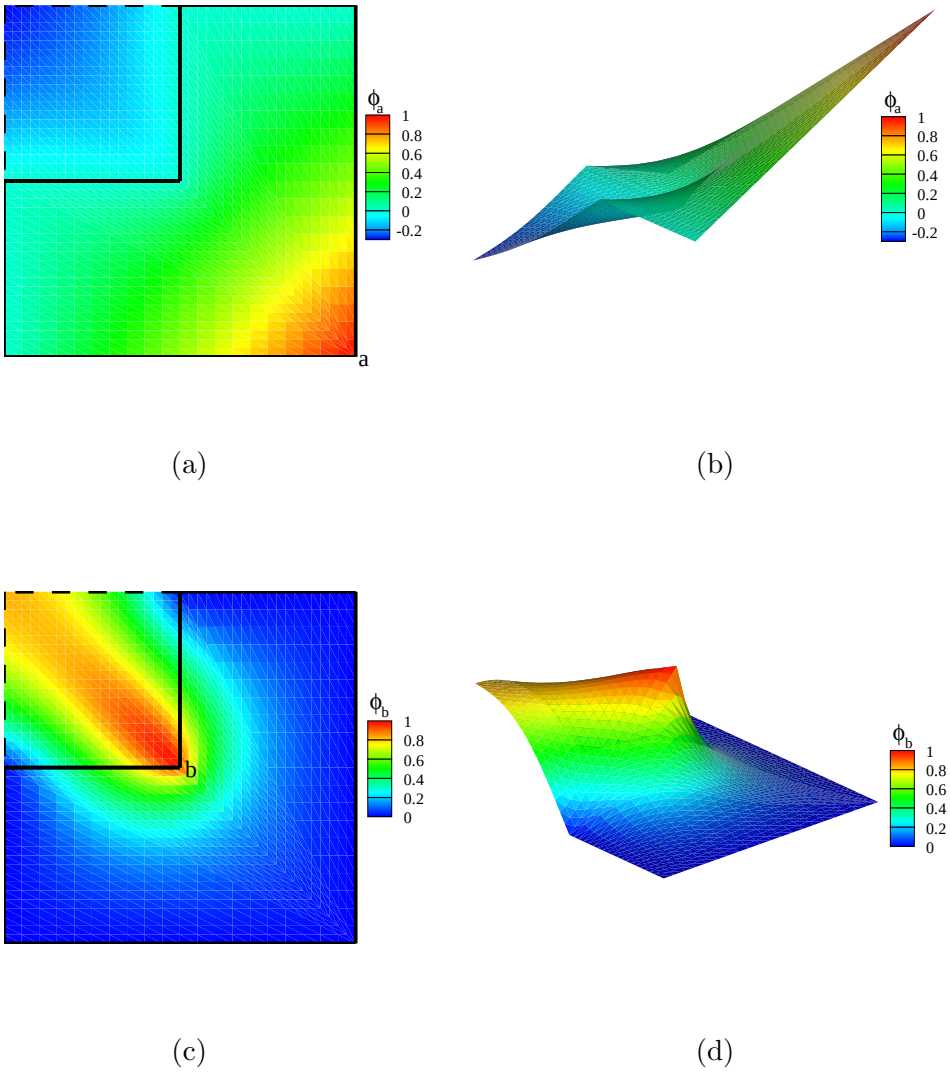
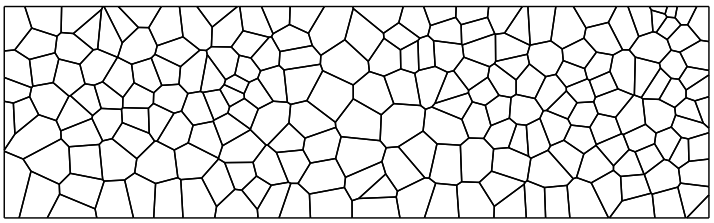


Fig. 5. Smooth extension of mean value coordinates in the plane. Nodes are vertices of a nonconvex polygon (thick line), and the square on the top-left in (a) and (c) lies outside the polygon. (a) and (b) Contour and 3D plot of $\phi_a(\mathbf{x})$; and (c) and (d) Contour and 3D plot of $\phi_b(\mathbf{x})$.

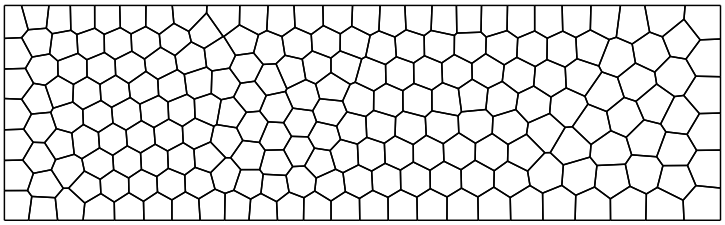
and

$$\boldsymbol{\nabla}\tilde{w}_i(\mathbf{x}) = \frac{\boldsymbol{\nabla}\Big(|\mathbf{r}_i \times \mathbf{r}_{i+1}|\Big) - \tilde{w}_i\Big(r_i\boldsymbol{\nabla}r_{i+1} + r_{i+1}\boldsymbol{\nabla}r_i + 2\mathbf{x} - \mathbf{x}_i - \mathbf{x}_{i+1}\Big)}{r_i r_{i+1} + \mathbf{r}_i \cdot \mathbf{r}_{i+1}}. \quad (5c)$$

Polygonal meshes are generated by using the Voronoi tessellation of the domain [19]. To this end, first a set of random seed points are inserted in the domain, and then the Voronoi diagram of this set of generators is constructed. This results in a nonuniform polygonal mesh (see Fig. 6a). Quasiuniform



(a)



(b)

Fig. 6. Polygonal meshes. (a) Initial mesh; and (b) Final mesh.

polygonal meshes are constructed using the centroidal Voronoi tessellation (CVT) algorithm. In a centroidal Voronoi diagram, the generator point is the same as the centroid of the Voronoi cell. Lloyd's algorithm is suitable to compute the centroidal Voronoi diagram [24]:

(1) Generate a set of random generator $\mathbf{x}_i$.
(2) Construct the Voronoi diagram of this set of generators.
(3) Calculate the centroid of each Voronoi cell $\mathbf{C}_i$.
(4) If $\mathbf{C}_i$ converges to $\mathbf{x}_i$, stop, otherwise put $\mathbf{x}_i = \mathbf{C}_i$, goto step 2.

An initial nonuniform polygonal mesh is constructed from random generators (Fig. 6a), and a final quasiuniform polygonal mesh is obtained by applying Lloyd's algorithm to the initial mesh (Fig. 6b). A public-domain package is used to generate the CVT mesh [25]. Nonuniform discretizations with local refinement are obtained if a nonuniform density function is used in the CVT algorithm; a few such examples are presented in Reference [19].

2.2 Quadtree meshes

Quadtree is a hierarchical data structure that is based on the recursive decomposition of a square into four equal subelements [26]. A quadtree mesh along with its representative tree is shown in Fig. 7. As illustrated in Fig. 7, hanging

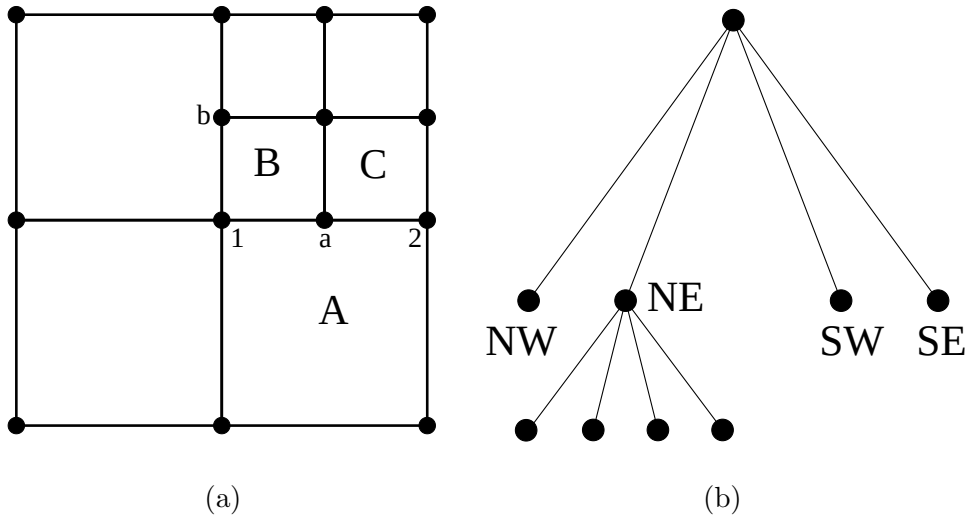


Fig. 7. A quadtree mesh and its representative tree. Hanging nodes a and b are generated due to level mismatch between adjacent elements.

nodes arise in a quadtree mesh if adjacent elements are not in the same level of refinement. Owing to the generation of hanging nodes, classical finite element shape functions are nonconforming on quadtree meshes. Typically, Lagrange multipliers, Nitsche's method, or use of multipoint constraints are used to address the issue of nonconformity. As an alternative, directly constructing conforming approximations on a quadtree is appealing—the quadtree data structure is untouched and a standard Galerkin formulation suffices with no changes in the properties of the stiffness matrix. The application of B-spline finite elements [27], hierarchical nodal refinement [28, 29], and use of natural neighbor basis functions [18,30] are a few approaches that share this viewpoint.

In this paper, we adopt the technique introduced in Reference [18], where quadtree elements are considered as special cases of polygonal elements in which at least three nodes are collinear. The implementation parallels that described in Section 2.1 for polygonal elements. First, Laplace shape functions are constructed on the polygonal reference element and then through an isoparametric mapping, the shape functions on the quadtree element are obtained. The mapping from a pentagonal reference element to element A is shown in Fig. 8. As indicated in Section 2.1, the Laplace interpolant is linear on the edges of the physical element, and therefore, piecewise linear interpolation is realized along edge 1–a–2 of element A (see Fig. 7). Since the shape functions of elements B and C in Fig. 7 are linear along edges 1–a and a–2, the Laplace interpolant is conforming on edge 1–2. The numerical integration scheme on quadtree elements follows that done for polygonal elements. As an alternative to the above approach, conforming approximations can be constructed on quadtree meshes using mean value coordinates [14, 15], which is a potentially preferred choice in three dimensions.

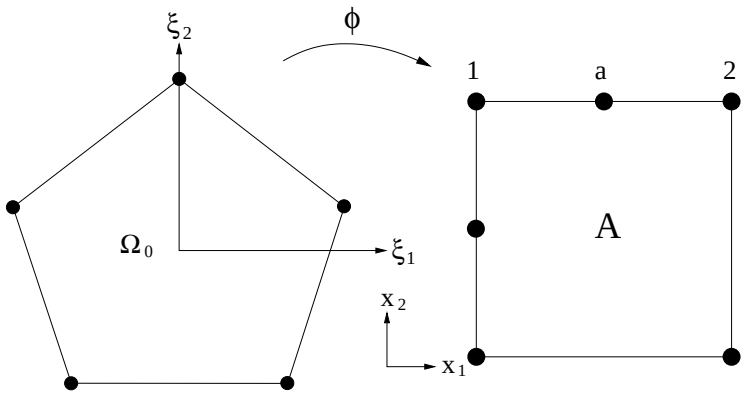


Fig. 8. Mapping from a regular pentagon to a quadtree element with one hanging node.

3 Extended finite elements for two-dimensional crack modeling

In contrast to the classical finite element method where crack surfaces are explicitly meshed and must conform to mesh boundaries, in the extended finite element method, inter-element crack discontinuities are modeled entirely independent of the underlying mesh. In the X-FEM, cracks are represented via the displacement approximation—enriching the classical displacement-based approximation by a function that is discontinuous across the crack permits crack modeling to be realized. The enriched displacement field in the X-FEM is [4]:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i\in I} \phi_i(\mathbf{x})\mathbf{u}_i + \sum_{j\in J} \phi_j(\mathbf{x})H(\mathbf{x})\mathbf{a}_j + \sum_{k\in K} \phi_k(\mathbf{x}) \sum_{\alpha=1}^{4} \psi_\alpha(\mathbf{x})\mathbf{b}_{k\alpha}, \quad (6)$$

where $\phi_i(\mathbf{x})$ is the polygonal basis function of node i, $\mathbf{u}_i$ are the classical degrees of freedom associated with node i, $\mathbf{a}_j$ are the enriched degrees of freedom associated with node j and the Heaviside function $H(\mathbf{x})$ (discontinuous across the crack interior), and $\mathbf{b}_{k\alpha}$ are the enriched degrees of freedom associated with node k and the near-tip enrichment functions $\psi_\alpha(\mathbf{x})$, which are defined as [3]

$$\{\psi_\alpha(\mathbf{x}),\, \alpha = 1\text{–}4\} = \left\{ \sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}\sin\theta, \sqrt{r}\cos\frac{\theta}{2}\sin\theta \right\}, \quad (7)$$

where r and θ are local crack-tip polar coordinates of point $\mathbf{x}$. Furthermore, in Eq. (6), I is the set of all nodes in the mesh, the set K consists of nodes that contain the crack-tip within their basis function support closure, and J is the set of nodes whose basis function supports are cut by the crack interior

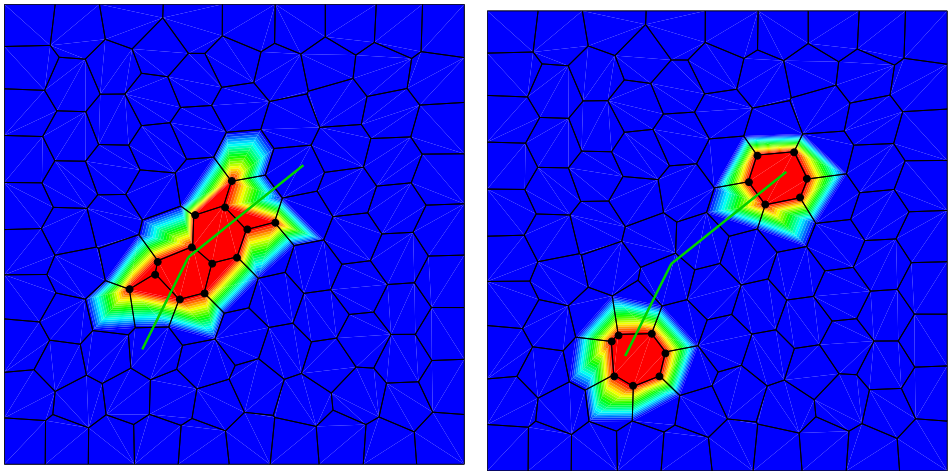
and do not belong to set K:

$$K = \{k : k \in I : \bar{\omega}_k \cap \Lambda_c \neq \emptyset\} \tag{8a}$$

$$J = \{j : j \in I : \omega_j \cap \Gamma_c \neq \emptyset, j \notin K\}, \tag{8b}$$

$$\omega_i = \{\mathbf{x} : \phi_i(\mathbf{x}) > 0\}, \tag{8c}$$

where ω_i is the support (open set) of $\phi_i(\mathbf{x})$, Γ_c denotes the crack surface and Λ_c is the crack-tip. The nodes that are enriched by the Heaviside and near-tip functions on a quasiuniform polygonal mesh containing an embedded crack are shown in Fig. 9.



(a) (b)

Fig. 9. Nodes that are enriched in a polygonal mesh for an embedded crack. (a) Heaviside; and (b) Near-tip nodal enrichment.

3.1 Weak form and discrete equations

Consider a domain $\Omega \subset \mathbf{R}^2$ with boundary $\Gamma = \Gamma_u \cup \Gamma_t$, where Γ_u and Γ_t are the essential and natural boundaries, respectively. The strong form is the equilibrium equations of linear elasticity with no body forces, small strain kinematics, linear elastic constitutive law, and $\mathbf{u} = \bar{\mathbf{u}}$ on Γ_u and $\mathbf{t} = \bar{\mathbf{t}}$ on Γ_t are the essential and natural boundary conditions, respectively. In addition, the interior of the domain contains one or more traction-free cracks. The weak form (principle of virtual work) is: find $\mathbf{u} \in S \times S$ such that

$$\int_\Omega \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\, d\Omega = \int_{\Gamma_t} \bar{\mathbf{t}} \cdot \delta\mathbf{u}\, d\Gamma \quad \forall \delta\mathbf{u} \in V \times V, \tag{9}$$

where $\mathbf{u}$ is the displacement field, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\boldsymbol{\varepsilon}$ is the small strain tensor, and δ is the first variation operator. The discrete trial

space $S^h \subset S$ and the discrete test space $V^h \subset V$ contain functions that are discontinuous across the crack. Referring to Eq. (6), the discrete trial space S^h is

$$S^h = \mathtt{span}\Big\{\{\phi_i\}_{i\in I} \cup \{\phi_j H\}_{j\in J} \cup \{\phi_k\psi_1, \phi_k\psi_2, \phi_k\psi_3, \phi_k\psi_4\}_{k\in K}\Big\}. \qquad (10)$$

Polygonal and enriched bases on a mesh that is cut by a crack are presented in Figures 10 and 11. For the convex element A in Fig. 10, Laplace shape functions are used, whereas for the nonconvex element A in Fig. 11, mean value coordinates are adopted. From Figures 10c and 11c, we observe that the enriched basis function is discontinuous across the crack within element A.

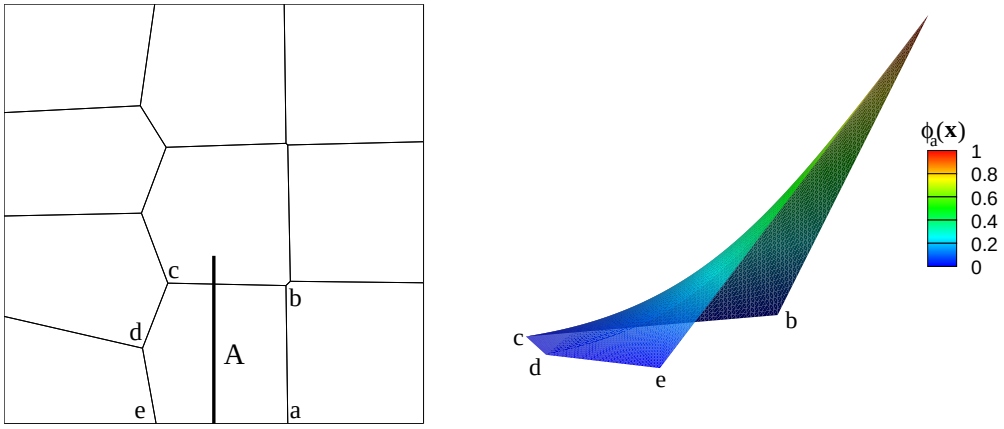
To obtain the weak form given in Eq. (9), the divergence theorem is invoked, which hinges on the regularity of displacement field within the domain (absence of singularities or discontinuities). To preserve this assumption, the integration domain must conform to the crack surfaces. For the purpose of numerical integration in elements that are cut by the crack discontinuity, the element is subdivided into subtriangles, and integration is done on the subtriangles [4]. The subtriangles of intersected polygonal elements in a polygonal mesh are shown in Fig. 12. The displacement (trial) functions and admissible displacement variations of the form given in Eq. (6) are inserted into Eq. (9). On invoking the arbitrariness of nodal variations, the discrete equations are obtained as

$$\mathbf{K}\mathbf{d} = \mathbf{f}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{ua} & \mathbf{K}_{ub} \\ \mathbf{K}_{au} & \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{bu} & \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{bmatrix}, \quad \mathbf{d} = \{\mathbf{u}\ \mathbf{a}\ \mathbf{b}\}^T, \quad \mathbf{f} = \{\mathbf{f}_u\ \mathbf{f}_a\ \mathbf{f}_b\}^T, \quad (11)$$

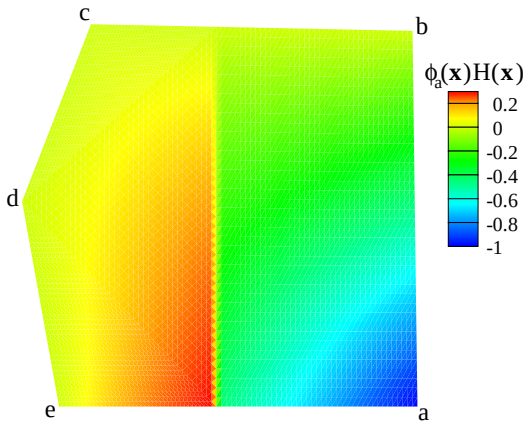
where $\mathbf{d}$ is the vector of nodal unknowns, and $\mathbf{K}$ and $\mathbf{f}$ are the global stiffness matrix and external force vector, respectively. In Eq. (11), $\mathbf{K}_{uu}$ is the stiffness matrix contribution due to the polygonal basis functions and other submatrices arise due to the presence of enriched basis functions. A detailed description of the entries in the X-FEM discrete equations is given in Reference [31].

3.2 Evaluation of enrichment functions

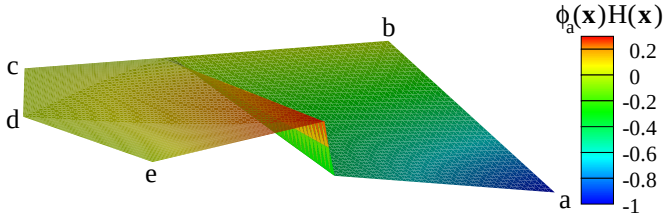
To assemble the stiffness matrix, the enrichment functions need to be evaluated at each Gauss point. On using the crack-tip coordinates, the polar coordinates (r, θ) of any point $\mathbf{x}$ are readily computed. On using these polar coordinates, the near-tip enrichment functions given in Eq. (7) are evaluated. The Heaviside



(a) (b)

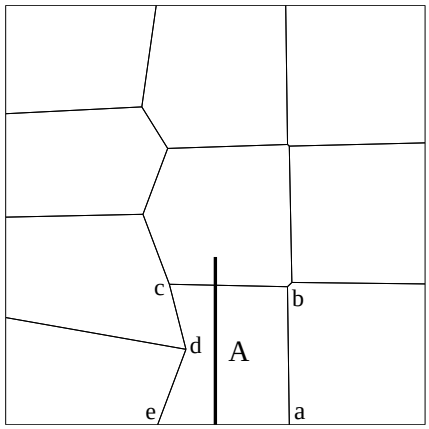


(c)

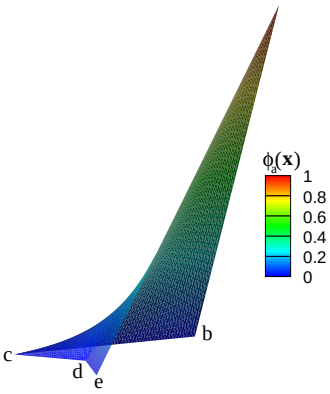


(d)

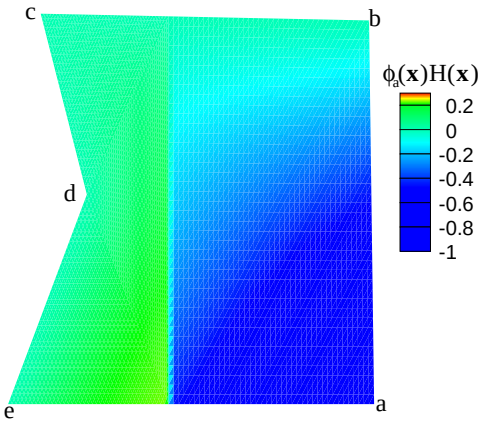
Fig. 10. Basis functions on a convex element cut by a crack. (a) Mesh and crack geometry; (b) $\phi_a(\mathbf{x})$; (c) Contour plot of enriched basis, $\phi_a(\mathbf{x})H(\mathbf{x})$; and (d) 3D plot of $\phi_a(\mathbf{x})H(\mathbf{x})$. The function $H(\mathbf{x})$ is +1 to the left of the crack and -1 to its right.



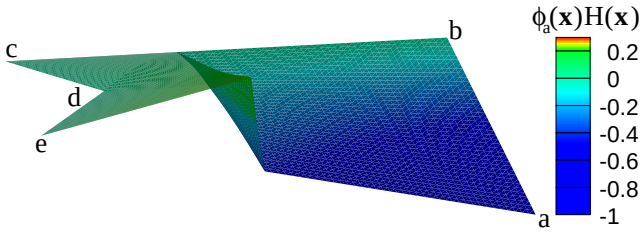
(a)



(b)



(c)



(d)

Fig. 11. Basis functions on a nonconvex element cut by a crack. (a) Mesh and crack geometry; (b) $\phi_a(\mathbf{x})$; (c) Contour plot of enriched basis, $\phi_a(\mathbf{x})H(\mathbf{x})$; and (d) 3D plot of $\phi_a(\mathbf{x})H(\mathbf{x})$. The function $H(\mathbf{x})$ is +1 to the left of the crack and -1 to its right.

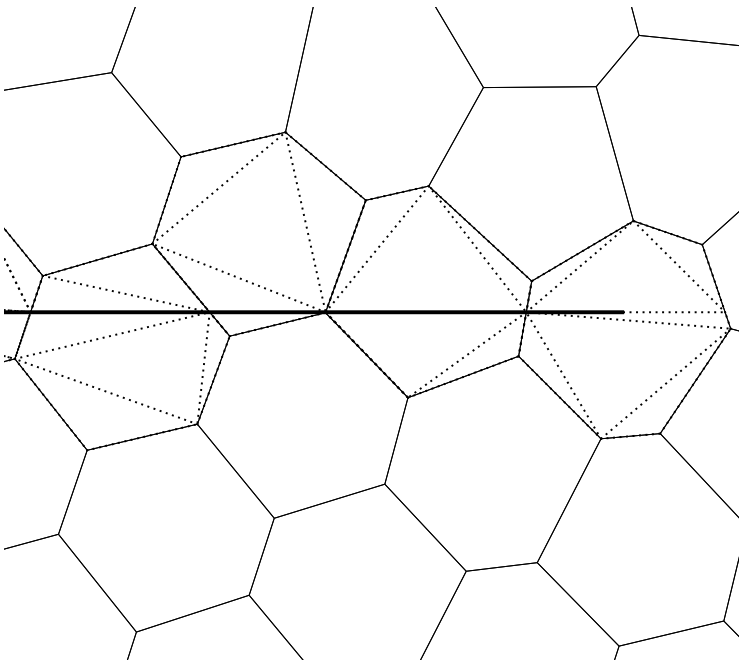


Fig. 12. The subtriangles of the intersected elements of a polygonal mesh.

function, which appears in Eq. (6), is defined as [4]

$$H(\mathbf{x}) = \begin{cases} 1 & \text{if } (\mathbf{x} - \mathbf{x}^*) \cdot \mathbf{n} \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{12}$$

where $\mathbf{x}^*$ is the closest point on the crack segment to point $\mathbf{x}$ and $\mathbf{n}$ is the normal to the crack segment at point x^*. If the vector $\mathbf{x} - \mathbf{x}^*$ belongs to the cone of normals at x^*, $H(\mathbf{x}) = 1$, and otherwise it is -1. Geometric predicates are used to evaluate the sign of $H(\mathbf{x})$.

4 Computer implementation

As in many recent studies [4, 21, 32–35], we have used C++ to implement the X-FEM. In contrast to finite elements, the key distinctions that need to be accounted for in the X-FEM are as follows: ability to have variable number of degree of freedom per node, necessary geometric tools to evaluate enrichment functions, and algorithms to partition elements for the purpose of numerical integration of the weak form integrals. The C++ Standard Template Library (STL) is extensively used in the code. The essential features of the class `DofManager` from previous C++ implementations [4, 32] have been adapted to fit our needs. A short description of the important C++ classes follows.

4.1 Class FeInterpolation

Class `FeInterpolation` is an abstract base class from which four classes are publicly derived: `Triangle3Interpolation` (linear interpolation on triangular elements), `Quad4Interpolation` (bilinear interpolation on quadrilateral elements), `LaplaceInterpolation` (Laplace interpolation on convex polygons), and `MVCInterpolation` (mean value interpolation on nonconvex polygons). Functions to compute the shape function and its derivatives are provided. Also, function *DirectMap* performs the mapping from the reference element to the physical element to find the global coordinates of a point in the reference element, and in function *InverseMap*, a Newton method is carried out to map the global coordinate to the local coordinate within the reference element [31]. This function is called to calculate the stiffness matrix entries of intersected and nonconvex elements as explained in Section 4.5.

4.2 Class Elements

Class `Elements` is an abstract base class from which three classes are publicly derived: `Tri3Element` for three-noded triangular elements, `Quad4Element` for four-noded quadrilateral elements and `PolygonalElement` for elements with more than four nodes and also for nonconvex elements. For each element, the element connectivity data, type of element interpolation function, number of degrees of freedom of the element, element material type, if element is intersected or not, and the list of element Gauss points are stored. A nonconvex element is considered as a polygonal element in which mean value interpolation is used. Function *ComputeStiffnessMatrix* is the key function of this class, and is used to assemble the element stiffness matrix. To perform this task, the function *IsIntersected* is called to verify if the element is a classical one or if it is intersected. If the element is not intersected but has enriched degrees of freedom, a standard stiffness matrix assembly procedure is used with higher number of Gauss points. For intersected elements, the element stiffness matrix is computed by taking into account contributions of classical and enriched basis functions. To this end, the value of the enrichment functions and its gradients are calculated at each Gauss point within the element. The integration procedure on different elements is described in Section 4.5.

4.3 Storage of degrees of freedom

Since in the X-FEM, nodes can have different number of degrees of freedom, the data structure for the storage of degrees of freedom (coefficients) is more involved than that in the standard finite element method. The data structure

for the degrees of freedom in an X-FEM code should be flexible enough to allow for saving variable number of degrees of freedom per node and to provide tools to add or delete degrees of freedom that are tied to a node. In our C++ code, three classes are used to manage the storage and retrieval of degrees of freedom: class `DofKey`, class `DofValue` and class `DofManager`. In what follows, instances of classes `DofKey` and `DofValue` are referred to as *key* and *value*, respectively. A *key* and a *value* are associated with a degree of freedom, and keys are used to differentiate between degrees of freedom. The couple (*key*, *value*) is saved in an STL map container and the *key* provides access to the *value* of a coefficient. In what follows, the *key* of coefficients (class `DofKey`) and *value* of coefficients (class `DofValue`) are presented.

4.3.1 Class DofKey

Every coefficient *key* is defined by four components: (1) The x_1- or x_2-direction that is associated with the coefficient; (2) The association of the coefficient with a basis function $\tilde{\phi}_i$ (either classical finite element or enriched basis function). If $\tilde{\phi}_i$ is an enriched basis function, then the type of enrichment can be determined; (3) The node number that the *key* is linked to; and (4) For coefficients tied to enriched basis functions, the particular internal discontinuity that the coefficient is tied to. For example, if two cracks pass through the same element, the coefficients associating with each crack should be distinguishable. Operators are defined to compare and retrieve *key*s.

4.3.2 Class DofValue

This class saves the standard data associated with a degree of freedom. The public functions of `DofValue` include members to save the value of the coefficient, the global equation number of the degree of freedom, and the status of the degree of freedom (free or fixed). Functions are defined to set and retrieve the member data.

4.3.3 Class DofManager

The couples (*key*, *value*) are stored in an STL map container using the interface class `DOFManager`. All the degrees of freedom are contained in the class `DOFManager`, and hence it is defined as a *singleton* [36]—only one instance of this class is created, which can be accessed in the entire code. This class includes methods for adding new degree of freedoms to the map container, changing the status of a degree of freedom, setting the value of coefficients, retrieving degree of freedoms, and finding degrees of freedom that are tied to a node.

4.4 Crack-mesh interaction

The intersection between a crack and the finite element mesh is used to determine the elements that need to be partitioned into subtriangles. The crack is represented by a union of line segments. To find the elements that are intersected by a crack, one checks for the intersection of each segment with element edges. For the numerical integration, the polygon (convex or nonconvex) is partitioned into subtriangles. The area-criterion given in Reference [4] is used to determine if a node is enriched by the Heaviside function. If a crack-tip lies inside an element, the nodes in the element connectivity are enriched by the near-tip functions given in Eq. (7). To check if an element contains a crack-tip, one of the in-polygon test algorithms given in Reference [37] is suitable. For each crack, two types of enrichment items exist: *crack interior* and *crack-tip*. The enrichment items of each node are saved in a STL set container. If the crack-tip and crack interior of the same crack exist in the enrichment item set of a node, the crack interior item is erased. The evaluation of the Heaviside and near-tip functions at a Gauss point is discussed in Section 3.2.

4.5 Numerical integration

The numerical integration scheme is tied to the element-type. The element-type dictates the interpolation that is used: for triangles and convex quadrilaterals, finite element interpolation is used, Laplace shape functions are used for convex polygons with more than four sides, and mean value coordinates apply for nonconvex polygons. A short description on the integration classes follows:

(1) Integration on non-intersected quadrilateral or triangular elements is performed by using Gauss quadrature rules. Class `StandardIntegration` serves the purpose of providing Gauss point weight and position on reference quadrilateral and triangular elements. An element is considered to be enriched if at least one of its nodes is enriched. For enriched triangular and quadrilateral elements, 6 and 3×3 Gauss quadrature rules are used, respectively.
(2) To integrate on non-intersected convex polygonal elements, the reference polygonal element is triangulated. In class `PolygonalIntegration`, the polygonal reference element is triangulated, and the location and weight of each Gauss points inside the reference subtriangle are saved. The location of the Gauss points on the physical element is obtained by using the isoparametric mapping from the reference element to the physical element. In each subtriangle, a 25-point Gauss quadrature rule is used.
(3) Integration on non-intersected nonconvex polygonal elements is performed

by triangulating the physical element and performing the integral on each physical subtriangle. Class `SplitElement` accomplishes this task by saving the global coordinate and weight of Gauss points on each physical subtriangle. Mean value coordinates are calculated on the physical element, and therefore no mapping is required. A public-domain C++ code [38] is used to triangulate a polygon (convex and nonconvex).

(4) Integration on the intersected elements is done by triangulating the physical element into subtriangles. The element is triangulated such that the crack does not intersect any triangle. Class `SplitElement` is used to save the weight and position of Gauss points on the physical element. For convex elements, the shape function and its derivatives are calculated on the corresponding reference element. Therefore, an inverse map from the physical element to the reference element is needed to find the local position of the Gauss point (code listed in Reference [31, Appendix] is used). The integration schemes for a convex polygon, a nonconvex polygon and an intersected convex polygon are shown in Fig. 13.

As in finite elements, the global stiffness matrix is computed via element-level assembly procedures. The direct sparse solver, SuperLU [39], has been interfaced to our code to solve the linear system of equations.

5 Numerical examples

The convergence of conforming polygonal finite elements is established in Reference [1] and use of quadtree finite elements to solve linear and nonlinear boundary-value problems is presented in References [18, 19]. In this section, four numerical examples are presented to illustrate the performance of the X-FEM on polygonal and quadtree meshes. Plane stress conditions are assumed with $E = 200$ GPa and $\nu = 0.3$. The domain form of the interaction integral is used to extract the stress intensity factors [40, 41] with a domain radius $r_q = 2h$ (h is the size of the crack-tip element); details on the extraction of SIFs in the X-FEM are provided in Reference [4].

5.1 Double-edge cracked plate in tension (DET)

The first example is a double-edge cracked plate in tension. The geometry and boundary conditions are shown in Fig. 14. The following parameters are chosen to solve the problem: $a/w = 0.25$, $h/w = 2$, and $w = 1$. The reference mode I stress intensity factor for this problem is [42]

$$K_I = \alpha\sigma\sqrt{\pi a}, \tag{13}$$

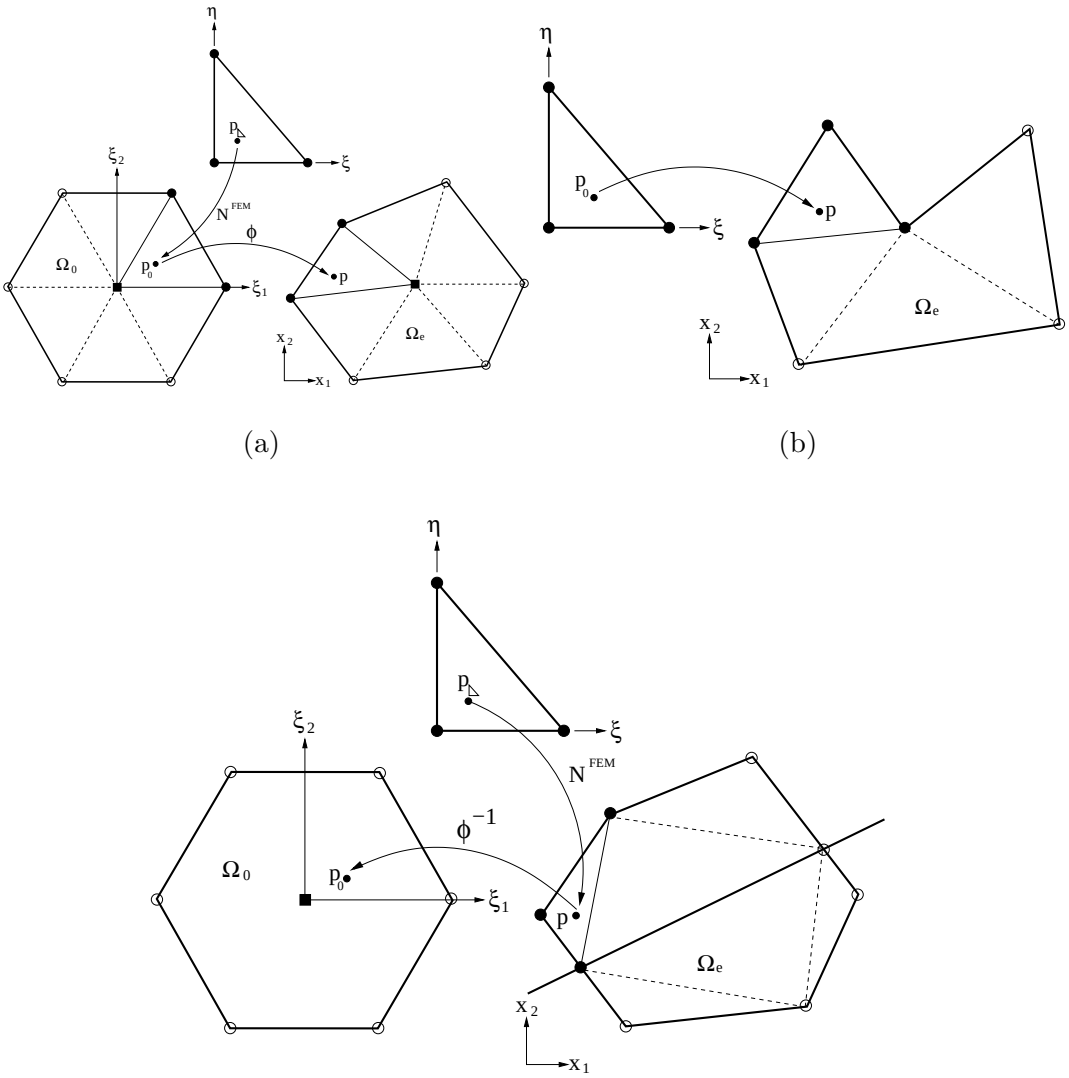
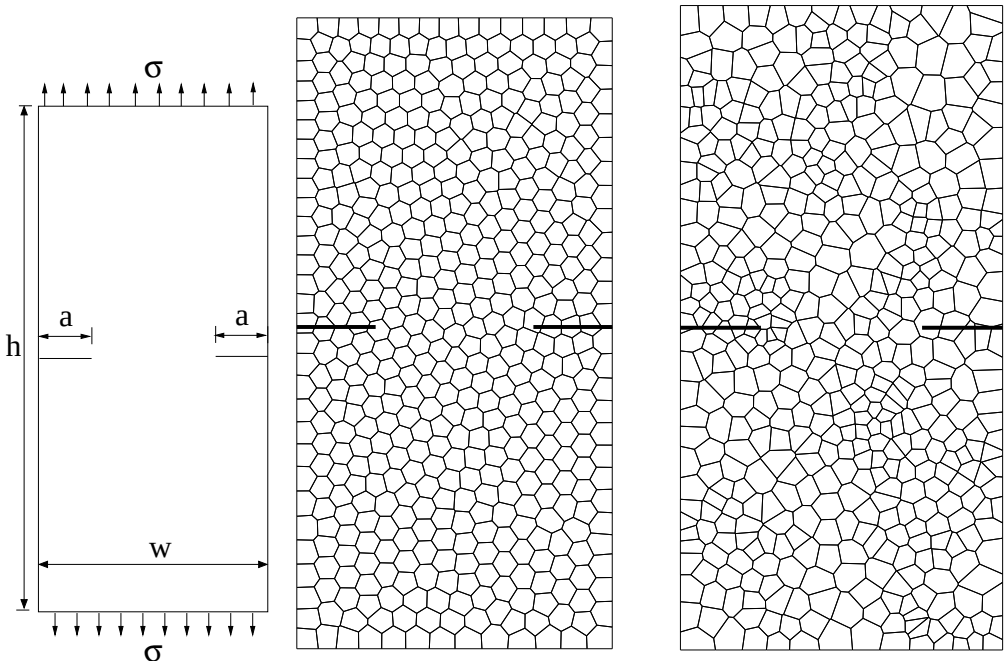


Fig. 13. Integration on polygonal elements. (a) Integration on a convex polygonal element is performed by triangulating the reference element; (b) Integration on a nonconvex element is done by triangulating the physical element; and (c) Integration on an intersected element is performed by triangulating the physical element.

where α is a finite dimension correction factor, which is given by

$$\alpha = 1.12 + 0.406\left(\frac{a}{w}\right) - 4.788\left(\frac{a}{w}\right)^2 + 15.44\left(\frac{a}{w}\right)^3. \tag{14}$$

On using Eqs. (13) and (14), the exact normalized stress intensity factor for the chosen set of parameters is $K_I/\sqrt{\pi a} = 1.1635$. Quasiuniform (CVT algorithm) polygonal and nonuniform polygonal meshes along with uniform quadrilateral meshes are used to solve the problem and the results are presented in Table 1. The polygonal meshes with 500 elements are shown in Figures 14b and 14c.



(a) (b) (c)

Fig. 14. Double-edge cracked plate in tension. (a) Geometry; (b) Quasiuniform mesh using CVT algorithm (500 elements); and (c) Nonuniform mesh (500 elements).

Table 1
Normalized SIF for double-edge cracked plate under tension.

Mesh	Number of Nodes	$\frac{K_I}{\sqrt{\pi a}}$	%Error
Uniform Polygonal mesh	402	1.1622	0.1
	1002	1.1625	0.1
	2001	1.1631	0.1
Nonuniform Polygonal mesh	402	1.1863	1.95
	1002	1.1788	1.31
	2001	1.1650	0.1
Quadrilateral mesh	435	1.1570	0.6
	1035	1.1611	0.2
	2016	1.1647	0.1

5.2 Oblique crack in an infinite plate

As a second example, an infinite plate with an oblique central crack under biaxial loading is considered. The plate is subjected to tensile loading of σ_1

and σ_2 in the x_1- and x_2-directions, respectively. The schematic of the problem is shown in Fig. 15a. The exact stress intensity factors are a function of the angle of the crack β with the x_2-axis and are given by [43]

$$K_I = (\sigma_2 \sin^2 \beta + \sigma_1 \cos^2 \beta)\sqrt{\pi a}, \tag{15a}$$

$$K_{II} = (\sigma_2 - \sigma_1) \sin \beta \cos \beta \sqrt{\pi a}. \tag{15b}$$

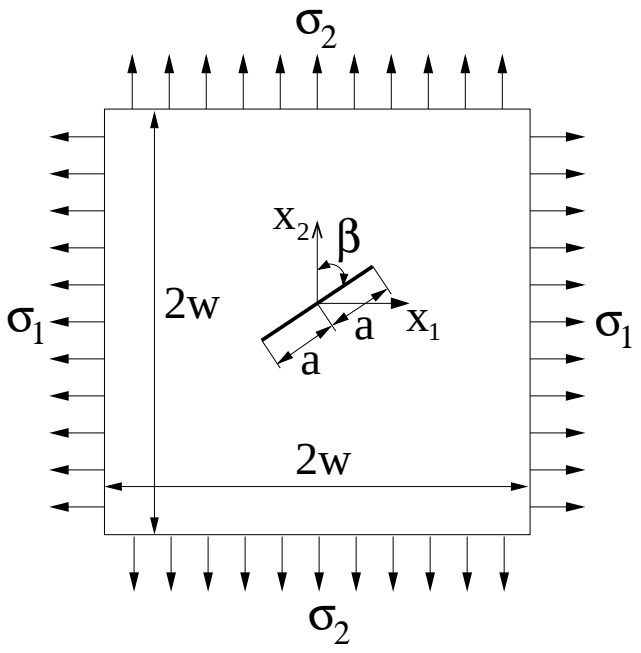
The infinite plate with a crack is modeled by a square plate with width of $2w$, crack of length $2a$, $w/a = 10$, $\sigma_1 = 1$, and $\sigma_2 = 2$. Since the crack length is much smaller than the specimen dimensions, the numerical results are compared to the reference solution of a crack in an infinite plate. The problem is solved on three meshes: a quadtree mesh with 292 elements, a nonconvex mesh with 292 elements, and a uniform quadrilateral mesh consisting of an array of 40×40 elements. The quadtree and the nonconvex mesh are shown in Figures 15b and 15c, respectively. The nonconvex mesh is generated from the quadtree mesh by perturbing the position of the hanging nodes. Laplace basis functions are adopted for quadtree elements, whereas mean value coordinates are used for nonconvex elements. The mode I and mode II stress intensity factors as a function of the angle β are presented in Table 2 and Fig. 16. Excellent agreement with the reference solution results is obtained, and for comparable accuracy, quadtree meshes with local refinement require far fewer degrees of freedom than a uniform rectangular mesh. Suitable enrichment functions to model intraelement cracks within the X-FEM have been developed [44]; the modeling of discontinuities on quadtree meshes is an alternative approach to represent and capture the growth of very small cracks without incurring significant additional costs.

5.3 Crack Growth Simulations

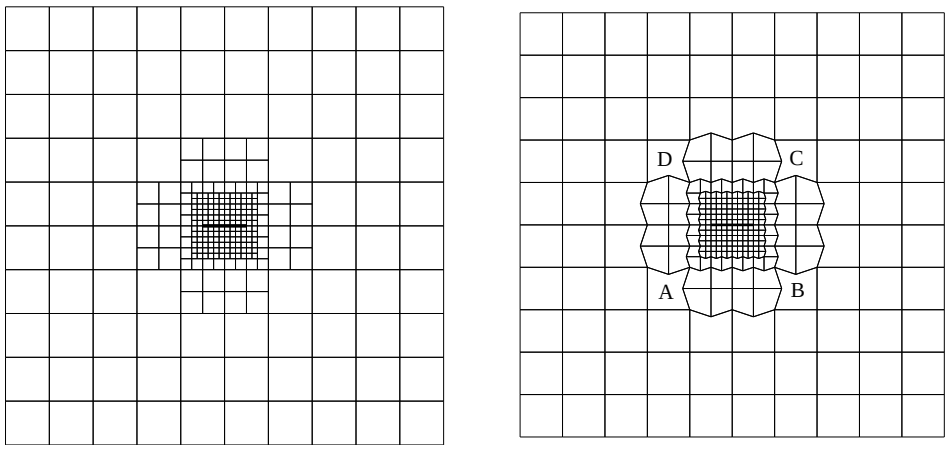
Quasi-static crack growth simulation on polygonal and quadtree meshes are presented. In all simulations, the crack length increment in each step is equal to twice the length of the element that contains the crack tip, and the kink angle is obtained using the maximum circumferential stress criterion as in previous extended finite element studies [4, 31].

5.3.1 Beam under four-point loading

The first crack growth example that we consider is a pre-cracked beam that is subjected to four-point shear loading [45]. The geometry and boundary conditions are shown in Fig. 17a, with the beam length $L = 4$ and load $P = 1$. The problem is solved on three polygonal meshes (CVT algorithm is used) with 500, 1000, 1500 elements and a 60×15 quadrilateral mesh. The crack paths are shown in Fig. 17. Initially, the cracks move away from each



(a)



(b) (c)

Fig. 15. Infinite plate with a central oblique crack. (a) Schematic of the crack geometry and loading; (b) Quadtree mesh; and (c) Quadtree mesh with nonconvex elements (for example, elements A, B, C and D).

other, but finally they are attracted toward the load points. The X-FEM crack paths are in agreement with the results presented in the literature [45, 46].

Table 2
Mode I and mode II stress intensity factors for an infinite plate with an oblique crack.

	Exact values		Quadtree mesh		Nonconvex mesh		Quadrilateral mesh	
β	K_I	K_{II}	K_I	K_{II}	K_I	K_{II}	K_I	K_{II}
90	2.5066	0.0	2.5171	0.0000	2.5171	0.0000	2.5263	0.0000
85	2.4971	0.1088	2.5098	0.1122	2.5090	0.1111	2.5115	0.1148
80	2.4688	0.2143	2.4662	0.2140	2.4658	0.2128	2.4814	0.2197
75	2.4226	0.3133	2.4092	0.3187	2.4070	0.3194	2.4138	0.3149
70	2.3600	0.4028	2.3451	0.4015	2.3431	0.4024	2.3574	0.4060
65	2.2828	0.4800	2.2708	0.4799	2.2700	0.4806	2.2863	0.4836
60	2.1933	0.5427	2.1937	0.5434	2.1932	0.5433	2.2059	0.5495
55	2.0943	0.5888	2.0823	0.5861	2.0823	0.5868	2.0819	0.5914
50	1.9888	0.6171	1.9714	0.6134	1.9706	0.6145	1.9748	0.6181
45	1.8800	0.6266	1.8785	0.6266	1.8782	0.6260	1.8703	0.6282
40	1.7711	0.6171	1.7543	0.6136	1.7536	0.6143	1.7593	0.6179
35	1.6656	0.5889	1.6536	0.5821	1.6538	0.5835	1.6544	0.5884
30	1.5666	0.5427	1.5637	0.5423	1.5640	0.5425	1.5723	0.5472
25	1.4771	0.4800	1.4681	0.4746	1.4684	0.4767	1.4796	0.4832
20	1.4000	0.4028	1.3900	0.3967	1.3893	0.3983	1.3997	0.4026
15	1.3373	0.3133	1.3278	0.3086	1.3269	0.3072	1.3360	0.3104
10	1.2911	0.2143	1.2890	0.2133	1.2892	0.2144	1.2976	0.2145
5	1.2628	0.1088	1.2692	0.1083	1.2688	0.1091	1.2704	0.1074
0	1.2533	0.0000	1.2586	0.0000	1.2585	0.0000	1.2632	0.0000

5.3.2 Inclined central crack in uniaxial tension

Crack growth of an inclined central crack in a rectangular plate under uniaxial tension in the x_2-direction is simulated. The plate dimension is 10×10, the angle of the crack with the x_1-axis is 30° and the ratio of the crack length to the plate dimension is $a/L = 0.01$. Since $a \ll L$, this example is used to illustrate the benefits of using X-FEM to study crack propagation of microcracks on quadtree meshes. The quadtree mesh (1128 elements) is shown in Fig. 18a, with a zoom of the crack region depicted in Fig. 18b. Since the plate is under pure mode I loading, crack advance will take place along the x_1-direction. The crack paths are presented in Figures 18c and 18d, and we observe that the

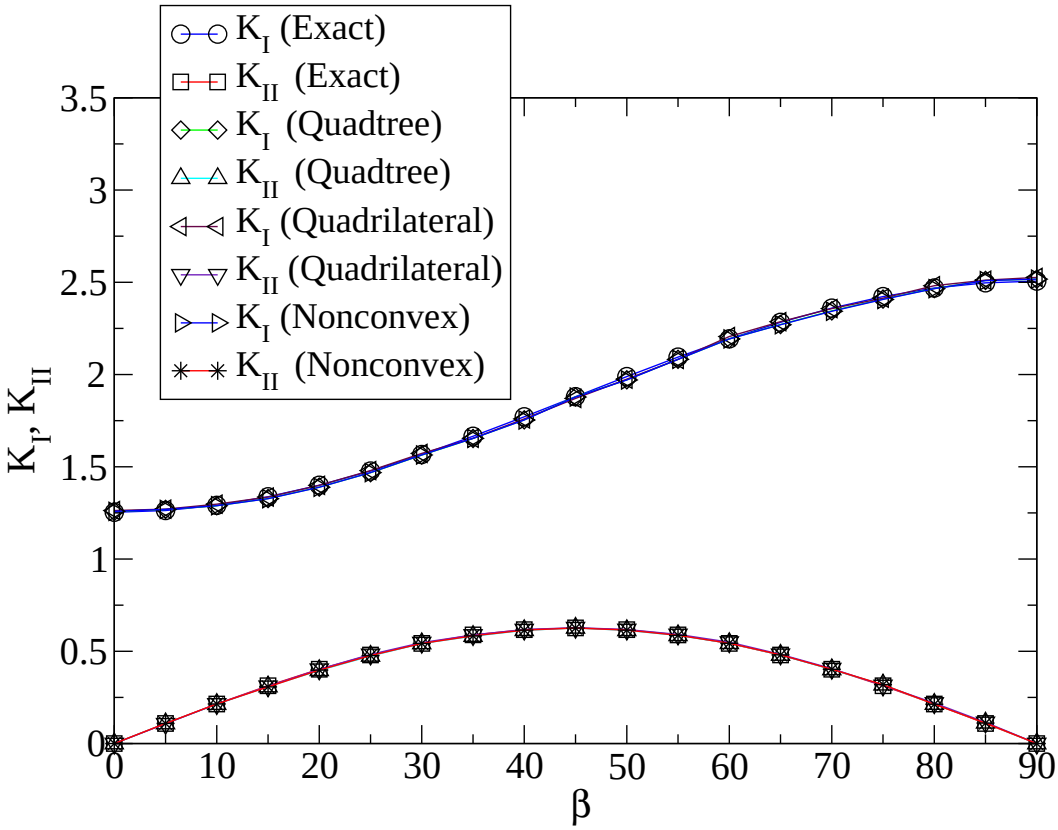
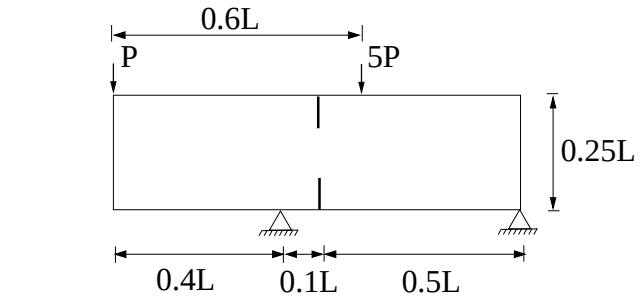


Fig. 16. Mode I and Mode II stress intensity factors versus angle β.

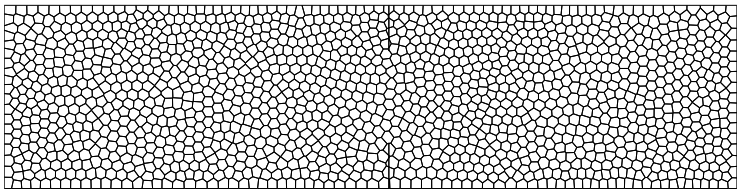
final trajectory is in agreement with theory. In addition to enabling modeling of microcracks on quadtree meshes, there exists the possibility of developing *a posteriori* error estimators on such meshes [19], which can provide improved accuracy at modest increase in computational costs.

6 Conclusions

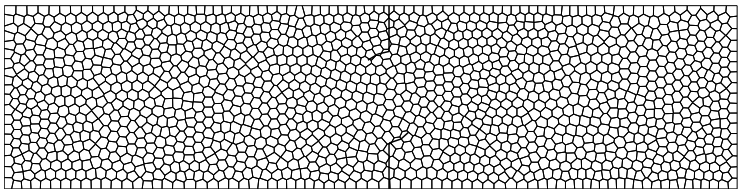
In this paper, two-dimensional crack growth modeling using the extended finite element method on arbitrary polygonal and quadtree meshes was introduced. The Laplace interpolant [17] was used to construct basis functions on convex polygonal and quadtree meshes, and for nonconvex elements, mean value coordinates [10] were adopted. These linearly complete polygonal basis functions were used within the framework of partition of unity framework [6] to enrich the classical displacement approximation. A discontinuous function and the asymptotic crack-tip fields were used as enrichment functions to model cracks, and crack propagation was simulated without the need for remeshing. Excellent agreement with reference solution results was obtained for the stress intensity factors on benchmark crack problems. The benefits and versatility of nonconvex elements and quadtree meshes in finite element modeling



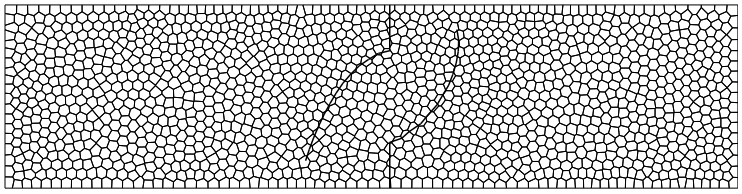
(a)



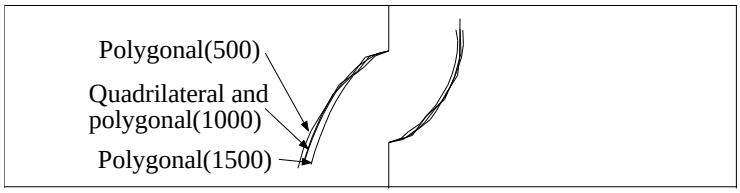
(b)



(c)



(d)



(e)

Fig. 17. Beam under four-point loading.

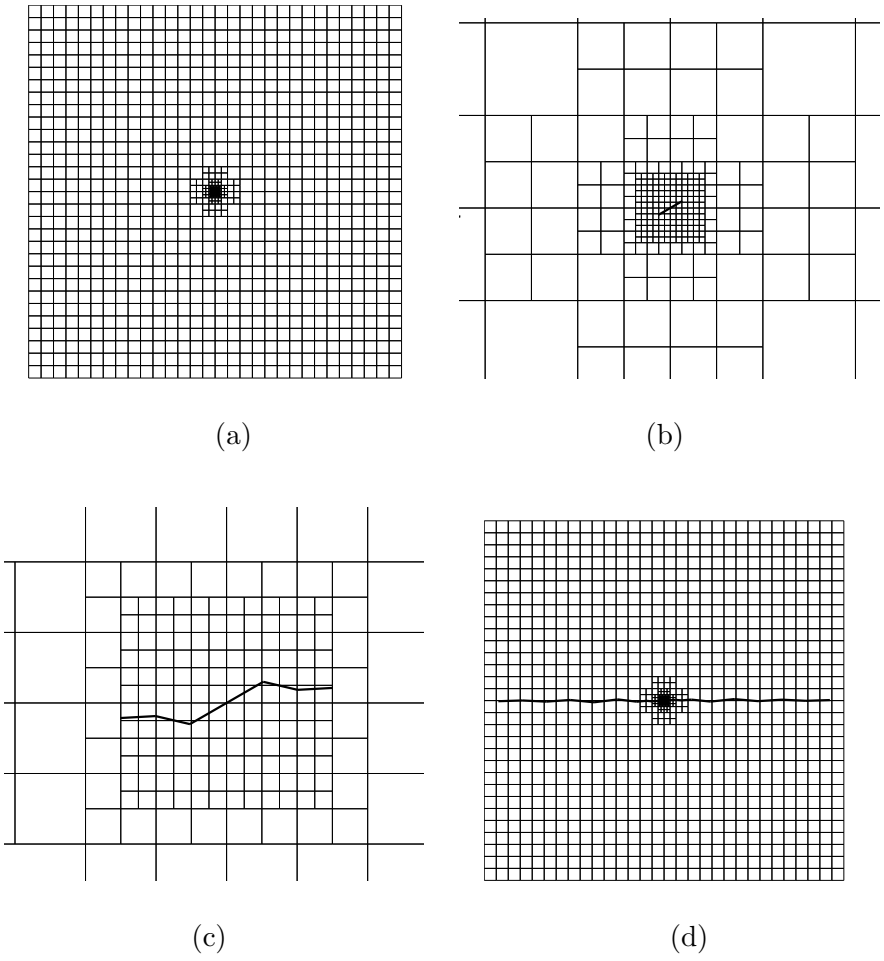


Fig. 18. Inclined central crack in an infinite domain. (a) Initial mesh; (b) Zoom on the refined area containing the initial crack; (c) Crack after second step; and (d) Final crack trajectory.

and simulation was demonstrated. Crack growth simulations on polygonal and quadtree meshes were presented to reveal the potential of the proposed methods in computational failure mechanics.

7 Acknowledgments

The research support of the National Science Foundation through contract CMMI–0626481 is gratefully acknowledged. The authors thank Eric Béchet and Rao Garimella for many helpful discussions.

References

[1] N. Sukumar, A. Tabarraei, Conforming polygonal finite elements, International Journal for Numerical Methods in Engineering 61 (2004) 2045–2066.

[2] S. R. Idelsohn, N. Calvo, E. Onate, Polyhedralization of an arbitrary 3D point set, Computer Methods in Applied Mechanics and Engineering 192 (22–24) (2003) 2649–2667.

[3] T. Belytschko, T. Black, Elastic crack growth in finite elements with minimal remeshing, International Journal for Numerical Methods in Engineering 45 (1999) 601–620.

[4] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, International Journal for Numerical Methods in Engineering 46 (1) (1999) 131–150.

[5] T. Strouboulis, K. Copps, I. Babuška, The generalized finite element method, Computer Methods in Applied Mechanics and Engineering 190 (32–33) (2001) 4081–4193.

[6] J. M. Melenk, I. Babuška, The partition of unity finite element method: Basic theory and applications, Computer Methods in Applied Mechanics and Engineering 139 (1996) 289–314.

[7] I. Babuška, I. Melenk, Partition of unity method, International Journal for Numerical Methods in Engineering 40 (1997) 727–758.

[8] E. L. Wachspress, A Rational Finite Element Basis, Academic Press, N. Y., 1975.

[9] M. Meyer, H. Lee, A. H. Barr, M. Desbrun, Generalized barycentric coordinates on irregular polygons, Journal of Graphics Tools 7 (1) (2002) 13–22.

[10] M. S. Floater, Mean value coordinates, Computer Aided Geometric Design 20 (1) (2003) 19–27.

[11] N. Sukumar, Construction of polygonal interpolants: A maximum entropy approach, International Journal for Numerical Methods in Engineering 61 (2004) 2159–2181.

[12] E. A. Malsch, G. Dasgupta, Shape functions for polygonal domains with interior nodes, International Journal for Numerical Methods in Engineering 61 (12) (2004) 1153–1172.

[13] E. A. Malsch, J. J. Lin, G. Dasgupta, Smooth two dimensional interpolants: a recipe for all polygons, Journal of Graphics Tools 10 (2) (2005) 11–23.

[14] M. S. Floater, G. Kós, M. Reimers, Mean value coordinates in 3D, Computer Aided Geometric Design 22 (7) (2005) 623–631.

[15] K. Hormann, M. S. Floater, Mean value coordinates for arbitrary planar polygons, ACM Transaction on Graphics 25 (4) (2006) 1424–1421.

[16] N. Sukumar, E. A. Malsch, Recent advances in the construction of polygonal finite element interpolants, Archives of Computational Methods in Engineering 13 (1) (2006) 129–163.

[17] N. H. Christ, R. Friedberg, T. D. Lee, Weights of links and plaquettes in a random lattice, Nuclear Physics B 210 (3) (1982) 337–346.

[18] A. Tabarraei, N. Sukumar, Adaptive computations on conforming quadtree meshes, Finite Elements in Analysis and Design 41 (2005) 686–702.

[19] A. Tabarraei, N. Sukumar, Adaptive computations using material forces and residual-based error estimators on quadtree meshes, Computer Methods in Applied Mechanics and Engineering 196 (25–28) (2007) 2657–2680.

[20] B. L. Karihaloo, Q. Z. Xiao, Modelling of stationary and growing cracks in FE framework without remeshing: a state-of-the-art review, Computers and Structures 81 (3) (2003) 119–129.

[21] S. Bordas, P. V. Nguyen, C. Dunant, A. Guidoum, H. Nguyen-Dang, An extended finite element library, International Journal for Numerical Methods in Engineering 71 (6) (2007) 703–732.

[22] R. Sibson, A vector identity for the Dirichlet tesselation, Mathematical Proceedings of the Cambridge Philosophical Society 87 (1980) 151–155.

[23] N. Sukumar, B. Moran, A. Y. Semenov, V. V. Belikov, Natural neighbor Galerkin methods, International Journal for Numerical Methods in Engineering 50 (1) (2001) 1–27.

[24] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: Applications and algorithms, SIAM Review 41 (1999) 637–676.

[25] J. Burkardt, CCVT_BOX: Centroidal Voronoi Tessellation Constrained to a Box, Available at `http://www.csit.fsu.edu/~burkardt/f_src/ccvt_box/ccvt_box.html`, 2004.

[26] H. Samet, Application of Spatial Data Structure, Addison-Wesley, New York, N.Y., 1990.

[27] P. Kagan, A. Fischer, P. Z. Bar-Yoseph, Mechanically based models: Adaptive refinement for B-spline finite element, International Journal for Numerical Methods in Engineering 57 (8) (2003) 1145–1175.

[28] P. Krysl, A. Trivedi, B. Zhu, Object-oriented hierarchical mesh refinement with CHARMS, International Journal for Numerical Methods in Engineering 60 (8) (2004) 1401–1424.

[29] P. Krysl, E. Grinspun, E. Schröder, Natural hierarchical refinement for finite element methods, International Journal for Numerical Methods in Engineering 56 (8) (2003) 1109–1124.

[30] J. J. Laguardia, E. Cueto, M. Doblaré, A natural neighbour Galerkin method with quadtree structure, International Journal for Numerical Methods in Engineering 63 (2005) 789–812.

[31] N. Sukumar, J.-H. Prévost, Modeling quasi-static crack growth with the extended finite element method. Part I: Computer implementation,, International Journal of Solids and Structures 40 (26) (2003) 7513–7537.

[32] E. Béchet, H. Minnebo, N. Moës, B. Burgardt, Improved implementation and robustness study of the X-FEM for stress analysis around cracks, International Journal for Numerical Methods in Engineering 64 (8) (2005) 1033–1056.

[33] P. Laborde, J. Pommier, Y. Renard, M. Salaün, High-order extended finite element method for cracked domains, International Journal for Numerical Methods in Engineering 64 (3) (2005) 354–381.

[34] P. Pommier, Y. Renard, Getfem++. An open source generic C++ library for finite element methods, Available at `http://www-gmm.insa-toulouse.fr/getfem`, 2005.

[35] B. L. Vaughan, Jr., B. G. Smith, D. L. Chopp, A comparison of the extended finite element method with the immersed interface method for elliptic equations with discontinuous coefficients and singular sources, Communications in Applied Mathematics and Computational Science 1 (1) (2006) 207–228.

[36] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Longman, Inc., Reading, MA, 1995.

[37] E. Haines, Point in polygon strategies, in: P. Heckbert (Ed.), Graphics Gems IV, Academic Press, 1994, pp. 24–46.

[38] J. W. Ratcliff, Efficient Polygon Triangulation, Available at `http://www.flipcode.com/cgi-bin/fcarticles.cgi?show=63943`, 2000.

[39] J. W. Demmel, J. R. Gilbert, X. S. Li, SuperLU user's guide, Tech. Rep. LBNL-44289, Lawrence Berkeley National Laboratory, Berkeley, CA. (2003).

[40] J. Yau, S. Wang, H. Corten, A mixed-mode crack analysis of isotropic solids using conservation laws of elasticity, Journal of Applied Mechanics 47 (1980) 335–341.

[41] B. Moran, C. F. Shih, Crack tip and associated domain integrals from momentum and energy balance, Engineering Fracture Mechanics 27 (1987) 615–642.

[42] H. Tada, P. C. Paris, G. R. Irwin, The Stress Analysis of Cracks Handbook, ASME Press, New York, N.Y., 2000.

[43] M. H. Aliabadi, D. P. Rooke, D. J. Cartwright, Mixed-mode Bueckner weight functions using boundary element analysis, International Journal of Fracture 34 (1987) 131–147.

[44] J. Bellec, J. E. Dolbow, A note on enrichment functions for modelling crack nucleation, Communications in Numerical Methods in Engineering 19 (12) (2003) 921–932.

[45] P. Bocca, A. Carpinteri, S. Valente, Size effects in the mixed mode crack propagation: softening and snap-back analysis, Engineering Fracture Mechanics 35 (1–3) (1990) 159–170.

[46] P. O. Bouchard, F. Bay, Y. Chastel, I. Tovena, Crack propagation modelling using an advanced remeshing technique, Computer Methods in Applied Mechanics and Engineering 189 (3) (2000) 723–742.