Received: 24 October 2019 | Revised: 29 June 2020 | Accepted: 4 August 2020

DOI: 10.1002/nme.6519

RESEARCH ARTICLE

WILEY

Computational Morphogenesis: Morphologic constructions using polygonal discretizations

Habeun Choi[1] | **Heng Chi**[2,3] | **Kyoungsoo Park**[1] | **Glaucio H. Paulino**[2]

[1]Department of Civil and Environmental Engineering, Yonsei University, Seoul, Korea

[2]School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, Georgia

[3]Siemens Corporate Technology, Princeton, New Jerssy

Correspondence
Kyoungsoo Park, Department of Civil and Environmental Engineering, Yonsei University, 50 Yonsei-ro Seodaemun-gu, Seoul 120-749, Korea.
Email: k-park@yonsei.ac.kr

Funding information
Georgia Institute of Technology, Grant/Award Number: Raymond Allen Jones Chair; Korea Institute of Energy Technology Evaluation and Planning, Grant/Award Number: 20174030201480; National Research Foundation of Korea, Grant/Award Number: 2018R1A2B6007054

Summary

To consistently coarsen arbitrary unstructured meshes, a computational morphogenesis process is built in conjunction with a numerical method of choice, such as the virtual element method with adaptive meshing. The morphogenesis procedure is performed by clustering elements based on a posteriori error estimation. Additionally, an edge straightening scheme is introduced to reduce the number of nodes and improve accuracy of solutions. The adaptive morphogenesis can be recursively conducted regardless of element type and mesh generation counting. To handle mesh modification events during the morphogenesis, a topology-based data structure is employed, which provides adjacent information on unstructured meshes. Numerical results demonstrate that the adaptive mesh morphogenesis effectively handles mesh coarsening for arbitrarily shaped elements while capturing problematic regions such as those with sharp gradients or singularity.

KEYWORDS

adaptive mesh coarsening, error estimation, polygonal elements, unstructured mesh, virtual element method

1 | INTRODUCTION

Mesh adaptation has been widely utilized as an effective tool to improve the accuracy and efficiency of numerical solutions in various engineering applications. The main feature of this technique is to decrease the number of degrees of freedom (DOFs), which reduces computational cost, while ensuring a desired level of accuracy in the numerical analysis. For example, Babuška and Rheinboldt[1,2] developed adaptive mesh refinement schemes to minimize the energy norm of the error. Zienkiewicz and Zhu[3] utilized a recovered gradient, that is, the recovery-based error estimator, to construct meshes for adaptive refinement. Since then, adaptive mesh refinement has been employed to solve various engineering problems, for example, shear bands,[4–6] cohesive fracture,[7–10] structural optimization,[11–14] hydrodynamics,[15] magnetohydrodynamics,[16] hemodynamics,[17] compressible flow,[18] and Lattice Boltzmann simulation.[19] The aforementioned references are not exhaustive and just represent a very small sample of the field.

Most previous works focused on adaptive refinement while only a few works were conducted on adaptive coarsening. For example, an adaptive mesh refinement and coarsening library, named as libMesh, was developed using a hierarchical refinement and coarsening scheme in conjunction with a tree data structure.[20] Molinari and Ortiz[6] performed mesh coarsening with an edge-collapse operator while introducing local retriangulation tools to improve mesh quality. To avoid mesh quality issues during edge-collapse, Park et al[8] employed adaptive mesh coarsening by reversing the adaptive refinement in 4k structured meshes, and Alhadeff et al[7] parallelized the adaptive mesh refinement and coarsening process. Bassi et al[21,22] proposed mesh agglomeration of uniform quadrilateral meshes in conjunction with the mesh free concept. Recently, Antonietti and Pennesi[23] employed the discontinuous Galerkin method for the agglomeration process of arbitrarily-shaped polygonal elements with edge-coarsening. Because the mesh agglomeration resulted in elements with arbitrary shapes, a special rule for the numerical integration on coarsened elements was developed. In summary, most previous adaptive mesh coarsening investigations were limited to the use of structured meshes because of mesh quality and numerical integration issues.

Polygonal discretizations have emerged as a new frontier in computational mechanics. In this study, to investigate mesh coarsening for arbitrary unstructured meshes, computational morphogenesis scheme is introduced. In the adaptive mesh morphogenesis, new cells are adaptively generated by clustering adjacent cells based on a posteriori error estimation. Additionally, an edge straightening scheme is employed to effectively reduce the number of nodes and global error of a generated mesh. The proposed scheme is not limited to the size or shape of cells, for example, skew and nonconvex, and accurately identifies problematic regions in the domain. Furthermore, the morphogenesis results provide lower global error than the ones of uniform meshes on various displacement fields. Once a reliable error estimator is available, then we have a plethora of methods at our disposal, such as finite elements (polygonal based)[24,25], virtual element methods (VEM)[26], mimetic finite differences (MFD)[27–29], discontinuous Galerkin methods (DG)[23,30,31], hybridizable discontinuous Galerkin methods (HDG)[32,33], and hybrid higher order methods (HHO)[34,35], to name a few. Among those, we arbitrarily select the VEM.

The remainder of the article is organized as follows. Section 2 explains the basic VEM and provides its underlining formulation. Section 3 presents the adaptive mesh morphogenesis procedure. Section 4 addresses the computational implementation of the morphogenesis procedure. Section 5 discusses the overall computational framework. Finally, Section 6 summarizes the key findings of the present article and provides recommendations for further morphogenesis research.

2 | NUMERICAL METHOD OF CHOICE

To perform mesh coarsening on arbitrary unstructured meshes, the virtual element method (VEM)[26] is selected because it provides flexibility on element shapes, namely convex and nonconvex polyhedra.[36,38] In the VEM, an explicit form of shape functions is not required to evaluate the stiffness matrix. To approximate the solution space, projection operators are employed, which can be exactly computed,[26,39] and the evaluation of a discrete bilinear form consists of consistency and stability terms. The VEM has been utilized to solve various engineering problems such as linear elasticity,[40,41] linear elastodynamics,[37,42,43] inelasticity problems,[44–46] fracture problems,[47–50] Stokes problems,[51] and topology optimization.[52–54] Alternatively, one should note that polygonal and polyhedral elements were employed using harmonic shape functions,[55,56] shape functions from a constrained minimization process,[57,58] and maximum-entropy shape functions,[59,60] while numerical integration should be carefully performed for the construction of element stiffness matrices.

2.1 | Virtual Element Method Formulation

The virtual element formulation for linear elasticity problems is presented. We shall restrict our attention to the 2D case. For a given solid $\Omega \subset \mathbb{R}^2$ with $\partial\Omega$ being its boundary, one assumes that a displacement field $\boldsymbol{u}_0$ is prescribed on a portion of the boundary ($\Gamma^{\boldsymbol{u}}$), and a traction field $\boldsymbol{t}$ is subjected to the other portion ($\Gamma^{\boldsymbol{t}}$), such as $\Gamma^{\boldsymbol{u}} \cup \Gamma^{\boldsymbol{t}} = \partial\Omega$ and $\Gamma^{\boldsymbol{u}} \cap \Gamma^{\boldsymbol{t}} = \emptyset$. In addition, a body force $\boldsymbol{f}$ is applied in the interior.

In the continuum setting, the principle of virtual work states that the (unique) equilibrating displacement field $\boldsymbol{u}$ among the set of kinematically admissible displacement field $\mathscr{K}$ satisfies

$$\int_\Omega \epsilon(\boldsymbol{v}) : [\mathbb{C}\epsilon(\boldsymbol{u})]\mathrm{d}\boldsymbol{x} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\mathrm{d}\boldsymbol{x} + \int_{\Gamma^{\boldsymbol{t}}} \boldsymbol{t} \cdot \boldsymbol{v}\mathrm{d}s \quad \forall \boldsymbol{v} \in \mathscr{K}^0, \tag{1}$$

where $\epsilon(\cdot) = (\nabla \cdot + \nabla^{\mathrm{T}} \cdot)/2$ is the infinitesimal strain operator, $\mathbb{C}$ is the fourth order linear isotropic elasticity tensor, and $\mathscr{K}^0 \subset \mathscr{K}$ is the set of kinematically admissible displacements that vanishes on $\Gamma^{\boldsymbol{u}}$. In the following subsections, the local space for lower order virtual element, together with the essential L^2 projection operators, and the corresponding VEM approximation for the above variational principle will be presented.

2.2 | Virtual element spaces for 2D

Consider a generic polygon E with m vertices numbered in counterclockwise fashion as $\boldsymbol{x}_1 \dots \boldsymbol{x}_m$. We first introduce an auxiliary virtual space, denoted as $\widetilde{\mathscr{V}}(E)$, which is defined as

$$\widetilde{\mathscr{V}}(E) = \{v \in \mathscr{H}_1(E) : \Delta v \in \mathscr{P}_1(E) \text{ in } E \text{ and } v|_e \in \mathscr{P}_1(e),\ \forall e \in \partial E\}, \tag{2}$$

where $\mathscr{P}_1(\cdot)$ is the space of linear functions, Δ stands for the Laplacian operator, and e is a generic edge of element E. According to its definition, $\widetilde{\mathscr{V}}(E)$ contains functions whose Laplacians are linear in the interior of E and boundary variations are piecewise linear. From the definition, we can show that $\mathscr{P}_1(E) \subseteq \widetilde{\mathscr{V}}(E)$ and therefore, we can define a projection operator $\Pi_E^{\nabla} : \widetilde{\mathscr{V}}(E) \to \mathscr{P}_1(E)$, such that $\forall v \in \widetilde{\mathscr{V}}(E)$, we have

$$\begin{cases} \int_E \nabla(\Pi_E^{\nabla} v) \cdot \nabla p_1 \mathrm{d}\boldsymbol{x} = \int_E \nabla v \cdot \nabla p_1 \mathrm{d}\boldsymbol{x} & \forall p_1 \in \mathscr{P}_1(E) \\ \sum_{i=1}^m \Pi_E^{\nabla} v(\boldsymbol{x}_i) = \sum_{i=1}^m v(\boldsymbol{x}_i) \end{cases}. \tag{3}$$

By using the divergence theorem and Equation (3), one obtains the following expression, that is,

$$\int_E \nabla(\Pi_E^{\nabla} v) \mathrm{d}\boldsymbol{x} = \int_E \nabla v \mathrm{d}\boldsymbol{x} = \int_{\partial E} v \boldsymbol{n} \mathrm{d}s = \sum_{i=1}^m \int_{e_i} v \boldsymbol{n}_{e_i} \mathrm{d}s, \tag{4}$$

where $\boldsymbol{n}_{e_i}$ is the unit outward normal vector of edge e_i (connecting vertices $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i-1}$). Since any function v in $\widetilde{\mathscr{V}}(E)$ varies piecewise linear on each edge of e, we can further simplify Equation (4) as

$$\int_E \nabla(\Pi_E^{\nabla} v) \mathrm{d}\boldsymbol{x} = \sum_{i=1}^m v(\boldsymbol{x}_i) \frac{\boldsymbol{n}_{e_i} |e_i| + \boldsymbol{n}_{e_{i+1}} |e_{i+1}|}{2}, \tag{5}$$

where $|e_i|$ denotes the length of edge e_i and we adopt the convention that $m+1 \doteq 1$ and $1-1 \doteq m$. Putting together Equation (5) and the second condition of Equation (3), we can show that the projection $\Pi_E^{\nabla} v$ can be uniquely and exactly computed with values of v on vertices of E and some geometric information of E. With the auxiliary virtual space $\widetilde{\mathscr{V}}(E)$ and projection operator Π_E^{∇}, the formal definition of the virtual space $\mathscr{V}(E)$ is given by

$$\mathscr{V}(E) = \left\{ v \in \widetilde{\mathscr{V}}(E) : \int_E (\Pi_E^{\nabla} v)\, p_1 \mathrm{d}\boldsymbol{x} = \int_E v p_1 \mathrm{d}\boldsymbol{x}, \forall p_1 \in \mathscr{P}_1(E) \right\}. \tag{6}$$

Notice that, by definition, $\mathscr{V}(E)$ is a subspace of $\widetilde{\mathscr{V}}(E)$, yet, it contains $\mathscr{P}_1(E)$ as well. We can also show that the dimension of the virtual space $\mathscr{V}(E)$ is exactly m and the values of its functions at the m vertices of E form a complete set of DOFs of $\mathscr{V}(E)$.

2.3 | L^2 projection operators for virtual elements

Through the formal definition of $\mathscr{V}(E)$, we only know its functions on the boundary of E but we do not know the function in the interior of E unless we solve the partial differential equation (PDE). In order to construct VEM approximation, an essential ingredient is the local projections. In this work, an L^2 projection operator $\Pi_1 : \mathscr{V}(E) \rightarrow \mathscr{P}_1(E)$ is defined, such that $\forall v \in \mathscr{V}(E)$, we have

$$\int_E \Pi_1 v\, p_1 \mathrm{d}\boldsymbol{x} = \int_E v\, p_1 \mathrm{d}\boldsymbol{x}, \quad \forall p_1 \in \mathscr{P}_1(E). \tag{7}$$

Comparing Equation (7) with the formal definition of $\mathscr{V}(E)$ in Equation (6), we conclude that $\Pi_1 v = \Pi_E^{\nabla} v$ for all $v \in \mathscr{V}(E)$. Since we have shown that the projection $\Pi_E^{\nabla} v$ is exactly computable with only v at vertices of E, the L^2 projection $\Pi_1 v$ is exactly computable with its DOFs and geometric information of E, even without knowing v anywhere in the interior of E.

Having defined the L^2 projection operator for v, we also introduce a second L^2 projection operator, which projects ∇v onto $[\mathscr{P}_0(E)]^2$, as

$$\int_E \Pi_0(\nabla v) \cdot \boldsymbol{p}_0 \mathrm{d}\boldsymbol{x} = \int_E \nabla v \cdot \boldsymbol{p}_0 \mathrm{d}\boldsymbol{x}, \quad \forall \boldsymbol{p}_0 \in [\mathscr{P}_0(E)]^2, \tag{8}$$

where $\mathscr{P}_0(E)$ is the space of constant functions. Simplifying expression Equation (8) leads to

$$\int_E \Pi_0(\nabla v) \mathrm{d}\boldsymbol{x} = \int_E \nabla v \mathrm{d}\boldsymbol{x} = \int_E \nabla(\Pi_E^{\nabla} v)\, \mathrm{d}\boldsymbol{x}. \tag{9}$$

Therefore, the second L^2 projection is readily available, once we have computed $\Pi_1 v$ (or equivalently, $\Pi_E^{\nabla} v$).

2.4 | Virtual element approximation for linear elasticity

With the ingredients defined in the preceding subsections, we are ready to introduce the virtual element approximation for linear elasticity problems. First, consider tessellation T_h of the domain Ω into nonoverlapping polygons. We further assume that T_h is conforming on both $\Gamma^{\boldsymbol{u}}$ and $\Gamma^{\boldsymbol{t}}$. Then, we can denote $\Gamma_h^{\boldsymbol{u}}$ and $\Gamma_h^{\boldsymbol{t}}$ as the unions of edges in T_h which belongs to $\Gamma^{\boldsymbol{u}}$ and $\Gamma^{\boldsymbol{t}}$, respectively. On tessellation T_h, the global displacement space, denoted as $\mathscr{K}_h$, is defined as

$$\mathscr{K}_h \doteq \{\boldsymbol{v}_h \in \mathscr{K} : \boldsymbol{v}_{h|E} \in [\mathscr{V}(E)]^2 \;\; \forall E \in T_h\}. \tag{10}$$

According to the above definition, each component of the local displacement field $\boldsymbol{v} = [v_x, v_y]^T$ in any element E belongs to the local virtual element space $[\mathscr{V}(E)]^2$ and has DOFs located at the vertices of E. Additionally, we define $\boldsymbol{\Pi}_1\boldsymbol{v}$ and $\boldsymbol{\Pi}_0\nabla\boldsymbol{v}$ as the actions of the projection operators Π_1 and Π_0 on each component of $\boldsymbol{v}$ and $\nabla\boldsymbol{v}$, respectively, given as:

$$\boldsymbol{\Pi}_1\boldsymbol{v} \doteq \begin{bmatrix} \Pi_1 v_x \\ \Pi_1 v_y \end{bmatrix}^T \text{ and } \boldsymbol{\Pi}_0\nabla\boldsymbol{v} \doteq \begin{bmatrix} (\Pi_0 \nabla v_x)^T \\ (\Pi_0 \nabla v_y)^T \end{bmatrix}. \tag{11}$$

Based on $\boldsymbol{\Pi}_0 \nabla \boldsymbol{v}$, we also define the L^2 projection of strain tensor $\boldsymbol{\Pi}_0\boldsymbol{\epsilon}(\boldsymbol{v})$ as

$$\boldsymbol{\Pi}_0\boldsymbol{\epsilon}(\boldsymbol{v}) \doteq \frac{1}{2}[\boldsymbol{\Pi}_0\nabla\boldsymbol{v} + (\boldsymbol{\Pi}_0\nabla\boldsymbol{v})^T]. \tag{12}$$

To construct the VEM approximation for linear elasticity, a major step is to construct $a_h^E(\boldsymbol{u}_h, \boldsymbol{v}_h)$, which is the VEM approximation to the following element-level bilinear form, that is,

$$a^E(\boldsymbol{u}_h, \boldsymbol{v}_h) = \int_E \boldsymbol{\epsilon}(\boldsymbol{v}_h) : [\mathbb{C}\boldsymbol{\epsilon}(\boldsymbol{u}_h)] \mathrm{d}\boldsymbol{x}, \tag{13}$$

where $\boldsymbol{u}_h$ is the approximated solution of VEM in a discretized domain. Based on the virtual element space and projection operators, we decompose $a^E(\boldsymbol{u}_h, \boldsymbol{v}_h)$ into

$$a_h^E(\boldsymbol{u}_h, \boldsymbol{v}_h) = |E| \boldsymbol{\Pi}_0 \boldsymbol{\epsilon}(\boldsymbol{v}_h) : [\mathbb{C}\, \boldsymbol{\Pi}_0 \boldsymbol{\epsilon}(\boldsymbol{u}_h)] + \alpha^E S^E(\boldsymbol{u}_h - \boldsymbol{\Pi}_1 \boldsymbol{u}_h, \boldsymbol{v}_h - \boldsymbol{\Pi}_1 \boldsymbol{v}_h), \tag{14}$$

where the first and second terms on the right-hand side are, respectively, named as the "consistency" and "stability" terms. The "consistency" term is responsible for the capturing the part of the bilinear form $a^E(\boldsymbol{u}_h, \boldsymbol{v}_h)$ that is essential for convergence. Because the "consistency" term only involves the projections of the local strain fields, it can be exactly computed without numerical integration. With only the consistency term, the system of equation resulting from the VEM approximation will be singular because of the existence of nonphysical zero-energy eigenmodes (eigenmodes which are not rigid-body modes in our context). Thus, the basic idea of the stabilization term in VEM is to stabilize those nonphysical spurious modes. To ensure that it does not affect the convergence rate, the choice of stabilization term $S^E(\boldsymbol{u}_h, \boldsymbol{v}_h)$ needs to scale like the energy norm $a^E(\boldsymbol{u}_h, \boldsymbol{v}_h)$ in terms of the element size[26].

In this work, we adopt the typically choice for "stability" term in the VEM literature, which consists of a bilinear form $S^E(\boldsymbol{u}_h, \boldsymbol{v}_h)$ and a scalar α^E defined as

$$S^E(\boldsymbol{u}_h, \boldsymbol{v}_h) = \sum_{\boldsymbol{x}_v \in E} \boldsymbol{u}_h(\boldsymbol{x}_v) \cdot \boldsymbol{v}_h(\boldsymbol{x}_v) \text{ and } \alpha^E = \frac{1}{4}\mathrm{tr}\mathbb{C} = \frac{1}{4} C_{ijij}, \tag{15}$$

respectively. Note that C_{ijkl} are components of the isotropic elasticity tensor $\mathbb{C}$ having both major and minor symmetries. The basic idea behind this choice of "stability" term is that both $S^E(\boldsymbol{u}_h, \boldsymbol{v}_h)$ and α^E are simple to evaluate. On the other hand, the VEM approximation of the loading terms related to body force $\boldsymbol{f}$ and surface traction $\boldsymbol{t}$ are defined as

$$\langle \boldsymbol{f}, \boldsymbol{v}_h \rangle_h = \sum_{E \in T_h} \boldsymbol{f}(\boldsymbol{x}_c^E) \cdot (\boldsymbol{\Pi}_1 \boldsymbol{v}_h)|_{\boldsymbol{x}=\boldsymbol{x}_c^E}, \tag{16}$$

and

$$\langle \boldsymbol{t}, \boldsymbol{v}_h \rangle_h = \sum_{e \in \Gamma_h^t} \oint_e \boldsymbol{t} \cdot \boldsymbol{v}_h \mathrm{d}s, \tag{17}$$

respectively, where $\boldsymbol{x}_c^E$ stands for the centroid of element E and $\oint_e$ could be any numerical integration scheme on e that is exact for any linear integrand. Notice that, by definition, $\boldsymbol{v}_h$ is linear on each edge e.

Finally, the VEM approximation for the linear elasticity problem Equation (1) consists of seeking $\boldsymbol{u}_h \in \mathscr{K}_h$ such that

$$\sum_{E \in T_h} a_h^E(\boldsymbol{u}_h, \boldsymbol{v}_h) = \langle \boldsymbol{f}, \boldsymbol{v}_h \rangle_h + \langle \boldsymbol{t}, \boldsymbol{v}_h \rangle_h \qquad \forall \boldsymbol{v}_h \in \mathscr{K}_h^0. \tag{18}$$

3 | ADAPTIVE MESH MORPHOGENESIS

An adaptive mesh morphogenesis strategy is proposed to investigate mesh coarsening for arbitrary unstructured meshes. The concept of the adaptive morphogenesis is simply to merge elements, which have relatively lower errors. In the following subsections, the adaptive morphogenesis procedure and a geometrical example are presented.

3.1 | Generation procedure

The adaptive mesh morphogenesis procedure consists of four steps: (1) identifying elements for coarsening, (2) clustering target elements, (3) edge straightening, and (4) removing skinny elements, as shown in Figure 1. First, elements which need coarsening, for example, the gray elements in Figure 1A, are searched based on a posteriori error estimation. Specifically, one searches and flags the elements whose normalized errors ($\epsilon_{E,n}$) are lower than a user-defined threshold (θ_{clst}), such that

$$\epsilon_{E,n} = \frac{\epsilon_E}{\max(\epsilon_E)} < \theta_{\text{clst}}, \tag{19}$$

where the normalized element error is the ratio of the element error (ϵ_E) to the maximum element error in the domain ($\max(\epsilon_E)$).

Next, for each flagged element, one checks whether its adjacent elements, which share nodes with the flagged element, are also marked or not. If normalized errors of the adjacent elements are also lower than the threshold, those elements are considered as a patch for the clustering. After defining the patches for all the flagged elements in the domain, elements of each patch are merged into a new single element. If there are flagged elements which do not have any adjacent elements with low errors, one excludes them from the mesh clustering procedure. A clustered element consists of nodes along the patch boundary (see Figure 1B). After setting the element connectivities of clustered elements, one removes original elements of the patches and inserts the new elements. Then, nodes, located inside of coarsened elements, do not have connections with elements. To maintain the topological consistency of a mesh, those nodes are removed. For example, one can select three elements, that is, E_1, E_2, and E_3 in Figure 2A, as a patch for the mesh clustering. The patch consists of 13 nodes, that is, $N_1, N_2, \dots, N_{13}$. Through the mesh clustering process, those three elements are merged into one element as illustrated in Figure 2B. The clustered element consists of 12 nodes, that is, $N_1, N_2, \dots, N_{12}$, and then N_{13} is removed.

After clustering elements, one introduces an edge straightening process to improve the quality of clustered elements and decrease the number of nodes in a domain. This process is achieved by selectively removing nodes which have two adjacent edges. One removes interior nodes, which share only two edges, regardless of those directions. For boundary nodes, to maintain the geometry of a domain, one only eliminates them when the adjacent edges are collinear. Interior nodes which have only two adjacent edges, for example, the white circles which are shared by dashed lines in Figure 1B, can be found after the mesh clustering procedure. By removing those nodes, the two adjacent edges are naturally replaced with a new straight edge, for example, the thick blue lines in Figure 1C. In the case of boundary nodes, one should consider directions of adjacent edges. For example, as shown in Figure 1C, the white square nodes share two collinear edges, and thus they are removed. However, the white triangular node describes the corner of the domain, and thus the edges, which share the white triangular node, are not collinear. Therefore, the white triangular node is not removed. Then, one achieves a coarsened mesh while maintaining the original geometry (see Figure 1D).

During the adaptive morphogenesis, triangular elements with a large aspect ratio may be generated, for example, the blue element in Figure 1E. Because such elements do not generally improve the accuracy of solution, the skinny element is merged with its adjacent element which shares the longest edge of the skinny element (see Figure 1F). In this study, triangular elements whose angles are greater than 130° or less than 20° are considered as the skinny elements.

3.2 | An illustrative example

An example of the adaptive mesh morphogenesis is illustrated by defining element errors in a domain. A rounded rectangular domain has the size of 3×1 with the corner radius of 0.5 and the "VEM" shaped holes, as shown in Figure 3. Element errors (ϵ_E) are arbitrary defined in the domain with small perturbations (ξ), given as

$$\epsilon_E(\boldsymbol{x}_c) = 2x_c - y_c^2 + 0.5 + \xi, \tag{20}$$

where $\boldsymbol{x}_c = [x_c, y_c]^T$ is the centroid of an element. A pseudorandom value ξ is obtained from the standard uniform distribution in the open interval of (0,1). Based on Equation (20), the elements have normalized errors within a range from 0.024 to 1.0 according to the centroid of each element. For an initial mesh discretization, a centroid Voronoi tessellation (CVT) mesh is utilized with 7001 elements and 13 946 nodes, as shown in Figure 4A. Higher element error is expected on the right- and bottom-region, and thus finer elements would be used in the corresponding region. To demonstrate a smooth transition from fine elements to coarse elements, one monotonically decreases the clustering threshold θ_{clst} in this example. First, one sets the clustering threshold as 0.831. Then, the mesh morphogenesis scheme is applied on regions where normalized element error is lower than 0.831. As illustrated in Figure 4B, most elements are merged while the elements in the right- and bottom-region are remained as they are. The number of elements is 2059 after the first mesh generation. Next, the element errors are updated on the basis of the centroid of merged

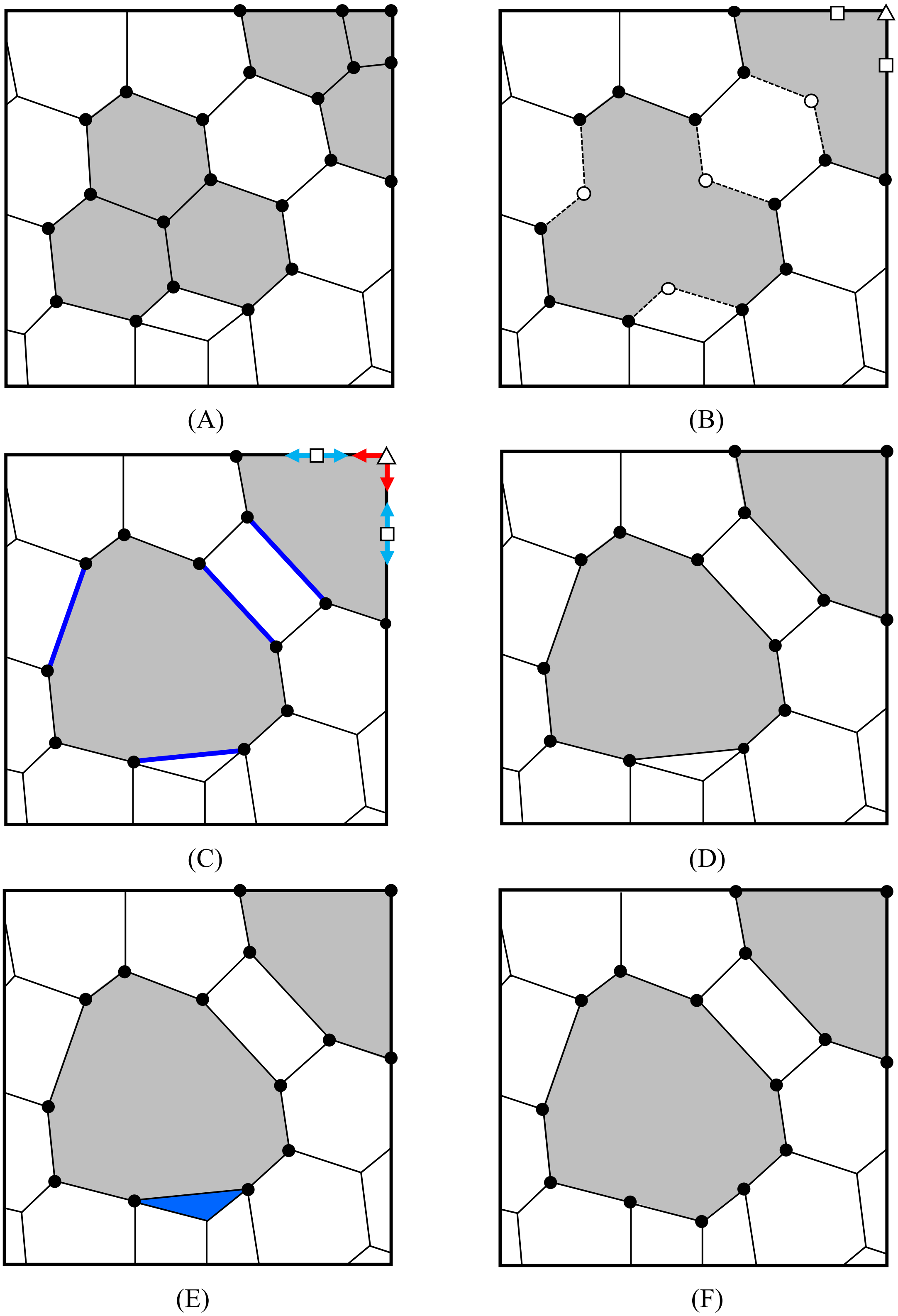


FIGURE 1 Schematics of the mesh morphogenesis procedure: A, identifying elements for coarsening, B, mesh clustering, C, edge straightening for interior nodes, D, edge straightening for boundary nodes, E, searching a skinny triangular element, and F, merging the skinny element with its adjacent element [Colour figure can be viewed at wileyonlinelibrary.com]

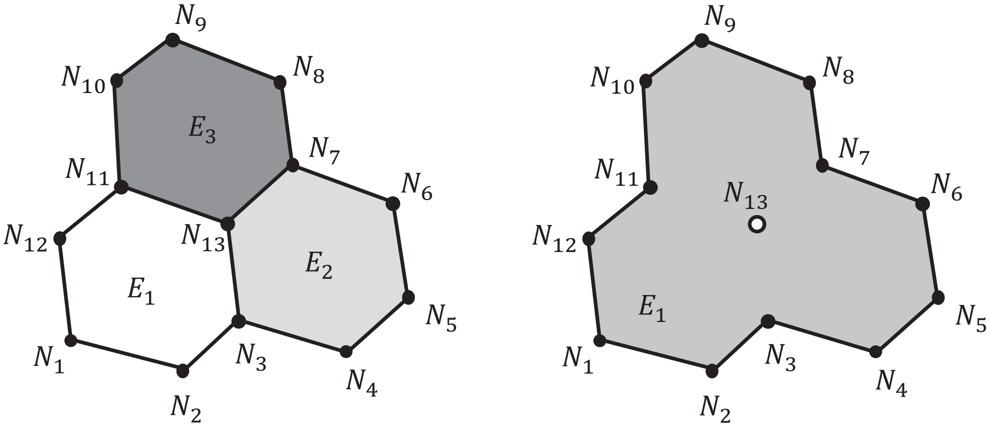


FIGURE 2 An example of a patch for the mesh clustering and its element connectivity: A, original element and B, clustered element

elements, and the morphogenesis is performed with the clustering threshold value of 0.625, which leads to the second mesh generation. Approximately 66.1% of the domain area is coarsened and the number of elements decreases by 1333, as shown in Figure 4C. Afterward, one more morphogenesis is conducted with the threshold value of 0.413. As expected, larger elements are generated on the left part of the domain (see Figure 4D). The generated mesh has 1243 elements after the third mesh generation. Note that the rounded boundary is maintained by small linear edges. This is because boundary nodes are eliminated only when the adjacent edges are collinear during the edge straightening process.

Additionally, mesh statistics for the generated meshes are illustrated in Figures 5 and 6. Figure 5 demonstrates the composition of generated meshes, while the numbers of occurrences are normalized with respect to the sum of possible events. For the initial discretization, the domain consists of 66.5% hexagon, 21.7% quadrilateral, 10.7% heptagon, 10.4% pentagon, and the others, as illustrated in Figure 5. As the adaptive morphogenesis is performed, the distribution of the mesh composition is wider than those of the initial discretization. In the third generation, the portions of the hexagon and quadrilateral decrease by 35.8% and 10.5%, respectively. More pentagons are generated, and those normalized appearance is 28.1%. Figure 6 shows the normalized edge length distribution. At the initial discretization, the average edge length is about 0.0122. After the performing of the adaptive morphogenesis, longer edges are generated due to merged elements, and the average edge length is 0.0222. In summary, the proposed adaptive morphogenesis scheme effectively represents mesh coarsening based on the element errors with various types and sizes of polygons.

4 | COMPUTATIONAL IMPLEMENTATION

In the adaptive morphogenesis, accurately estimating element errors and effectively retrieving adjacent information are essential. Thus, in this section, one first presents a posteriori error estimator,[61] which is utilized to identify elements with lower errors for coarsening. To handle mesh modification events, the topology-based data structure (TopS)[62,63] is employed, which provides adjacent information of topological entities. Finally, the morphogenesis procedure is summarized in Section 4.3.

4.1 | Error estimator

Based on the definition of the H^1-type skeletal norm, the skeletal error of the displacement from VEM simulation, that is, original displacement gradient error $\epsilon_{\mathbf{u},s}$, is given as,

$$\epsilon_{\mathbf{u},s} = \left[\sum_{E\in\Omega_h} h_E \sum_{e\in\partial E} \int_e (\nabla \boldsymbol{u} \cdot \boldsymbol{\tau}_e - \nabla \boldsymbol{u}_h \cdot \boldsymbol{\tau}_e) \cdot (\nabla \boldsymbol{u} \cdot \boldsymbol{\tau}_e - \nabla \boldsymbol{u}_h \cdot \boldsymbol{\tau}_e) \mathrm{d}e \right]^{1/2}, \tag{21}$$

where h_E is a characteristic size of an element, and $\boldsymbol{\tau}_e$ is the unit tangent vector of edge e. In most cases, the original error cannot be computed because the exact displacement gradient $\nabla \boldsymbol{u}$ is unknown. Alternatively, Chi et al[61] proposed a recovery-based a posteriori error estimator to approximate the original skeletal errors. The exact displacement gradient

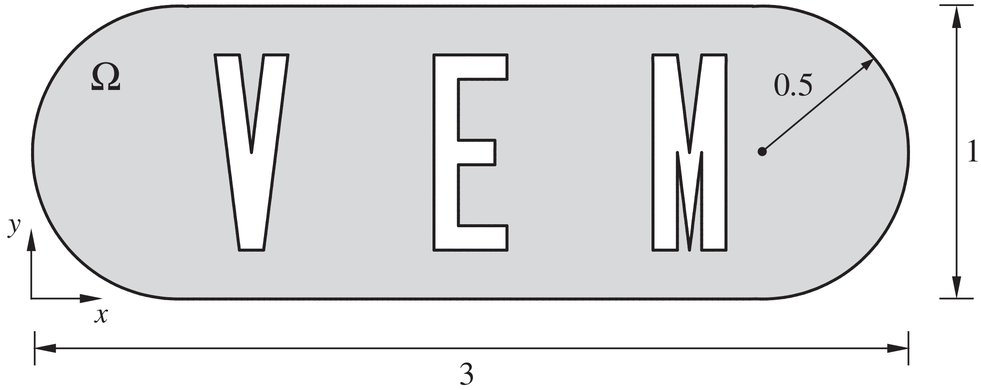


FIGURE 3 Domain description of the rounded rectangular domain with "VEM" shaped holes

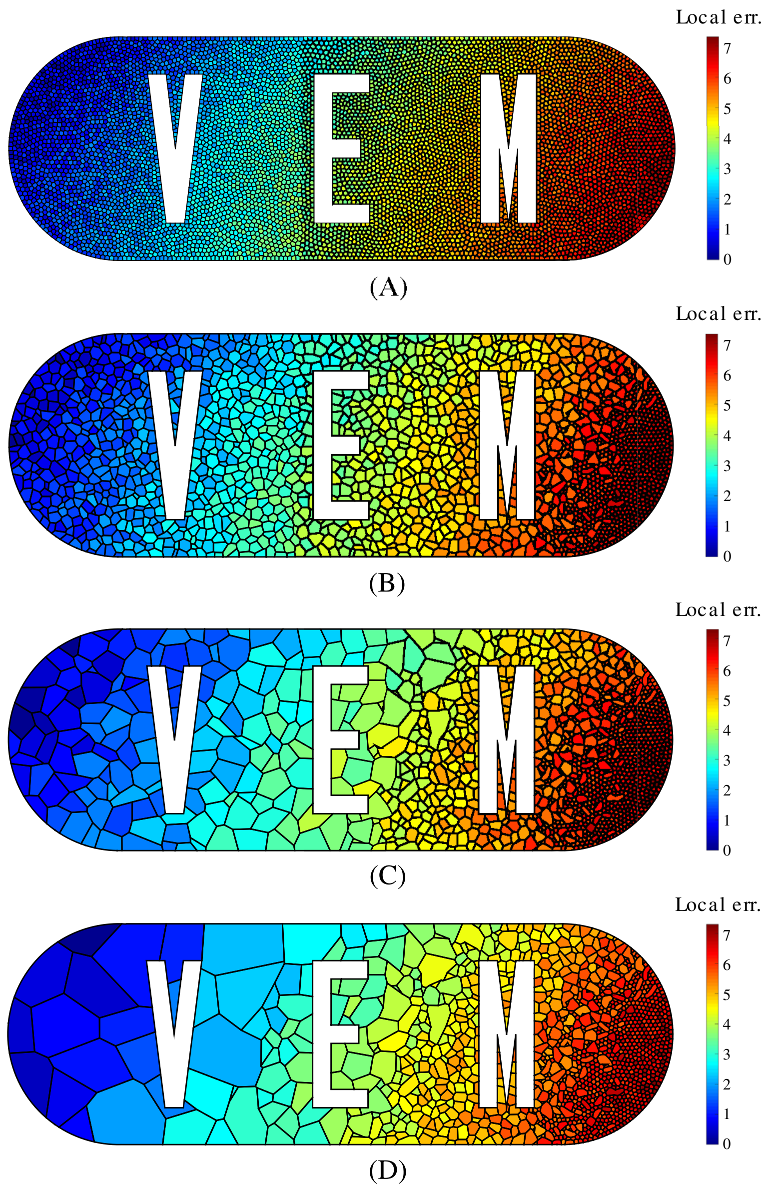


FIGURE 4 Generated meshes and those element errors during the adaptive morphogenesis: A, initial mesh, B, first mesh generation, C, second generation, and D, third generation [Colour figure can be viewed at wileyonlinelibrary.com]

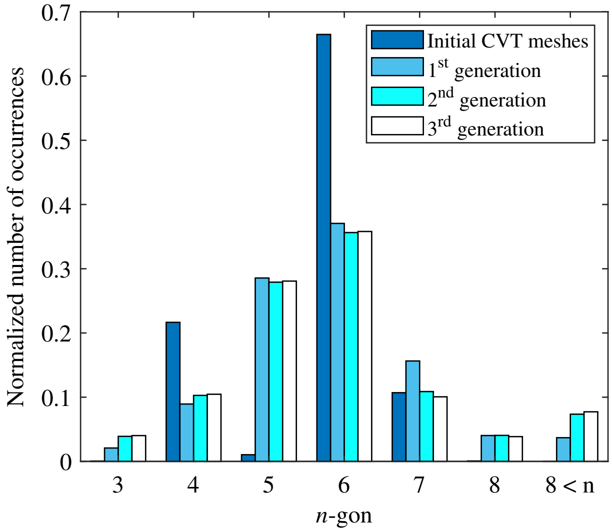


FIGURE 5 Composition of the initial and generated meshes with respect to the adaptive morphogenesis [Colour figure can be viewed at wileyonlinelibrary.com]

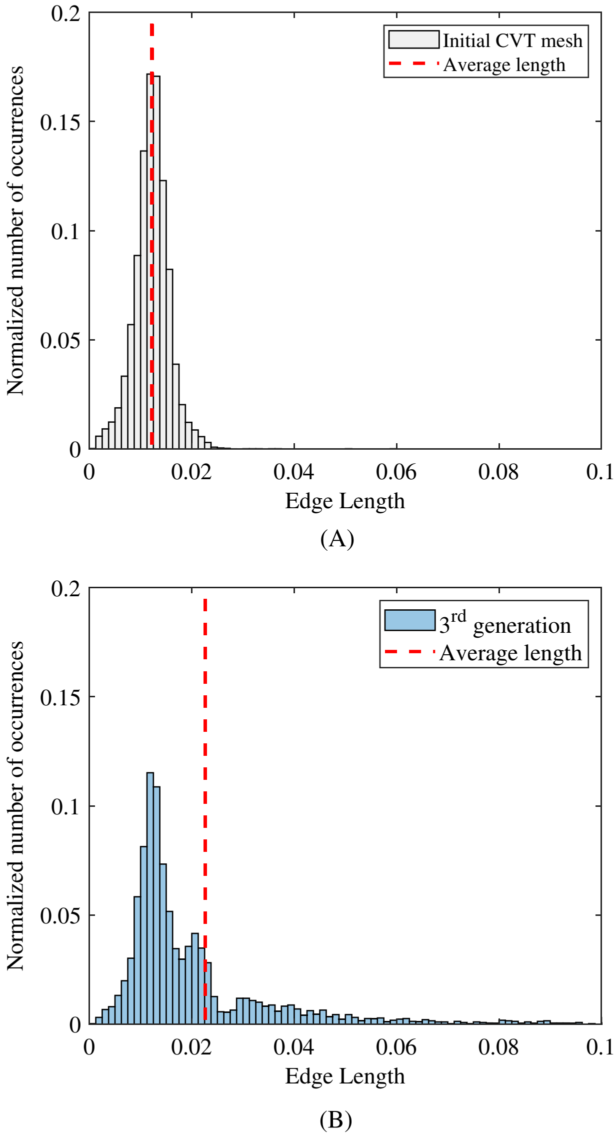


FIGURE 6 Edge length distribution according to the adaptive morphogenesis: A, the initial CVT mesh and B, third generation. Note that the red dashed line represents the average edge length for each mesh [Colour figure can be viewed at wileyonlinelibrary.com]

($\nabla \boldsymbol{u}$) is estimated by introducing a reconstructed displacement gradient ($G_h \boldsymbol{u}_h$). The a posteriori estimated element error $\widetilde{\epsilon}_{\mathbf{u},s}\big|_E$ is expressed as

$$\widetilde{\epsilon}_{\mathbf{u},s}\big|_E = \left[h_E \sum_{e \in \partial E} \int_e (G_h \boldsymbol{u}_h \cdot \boldsymbol{\tau}_e - \nabla \boldsymbol{u}_h \cdot \boldsymbol{\tau}_e) \cdot (G_h \boldsymbol{u}_h \cdot \boldsymbol{\tau}_e - \nabla \boldsymbol{u}_h \cdot \boldsymbol{\tau}_e) \mathrm{d}e \right]^{1/2}. \tag{22}$$

The reconstructed gradient is computed at each node in the domain, and the variation of $G_h \boldsymbol{u}_h$ along an edge is assumed to be linear. By summing all the estimated element errors in the domain Ω_h, the estimated global error $\widetilde{\epsilon}_{\mathbf{u},s}$ is evaluated as follows

$$\widetilde{\epsilon}_{\mathbf{u},s} = \left[\sum_{E \in \Omega_h} (\widetilde{\epsilon}_{\mathbf{u},s}\big|_E)^2 \right]^{1/2}. \tag{23}$$

The accuracy of the reconstructed displacement gradient and the error estimator are verified through various numerical examples.[61]

For the computation of the reconstructed displacement gradient at a given node $\boldsymbol{x}_i = [x_i, y_i]^T$, a patch of elements ($w_i$), which shares the node $\boldsymbol{x}_i$, is first defined. Within the patch (w_i), the reconstructed displacement gradient is given as

$$G_h \boldsymbol{u}_h(\boldsymbol{x}_i) = \nabla \mathbf{p}^i(\boldsymbol{x}_i) = \begin{bmatrix} \frac{\partial p_x^i}{\partial x}(\boldsymbol{x}_i) & \frac{\partial p_x^i}{\partial y}(\boldsymbol{x}_i) \\ \frac{\partial p_y^i}{\partial x}(\boldsymbol{x}_i) & \frac{\partial p_y^i}{\partial y}(\boldsymbol{x}_i) \end{bmatrix}, \tag{24}$$

where $\mathbf{p}^i = [p_x^i, p_y^i]^T$ is a quadratic vector field within w_i. The components of $\mathbf{p}^i$ are computed by minimizing the difference between a function (ξ_x, ξ_y) and the VEM solutions ($u_{h,x}$, $u_{h,y}$) along the x- and y-directions, that is,

$$p_x^i = \underset{\xi \in \mathscr{P}_2(w_i)}{\operatorname{argmin}} \sum_{j=1}^{n_w} [\xi_x(\boldsymbol{x}_j) - u_{h,x}(\boldsymbol{x}_j)]^2, \tag{25}$$

and

$$p_y^i = \underset{\xi \in \mathscr{P}_2(w_i)}{\operatorname{argmin}} \sum_{j=1}^{n_w} [\xi_y(\boldsymbol{x}_j) - u_{h,y}(\boldsymbol{x}_j)]^2, \tag{26}$$

where ξ_x and ξ_y are in the set of second degree polynomial functions [$\mathscr{P}_2(w_i)$], and n_w is the total number of nodes in w_i. In this study, Equations (25) and (26) are computed using the least square method.

To check the robustness and accuracy of the error estimator during the mesh morphogenesis, the estimated error $\widetilde{\epsilon}_{\mathbf{u},\,s}$ is compared with two error measures, that is, original errors $\epsilon_{\mathbf{u},s}$, and recovered errors $\epsilon_{\widetilde{\mathbf{u}},s}$, as discussed in Section 5.1. The recovered error is the skeletal error of the reconstructed displacement gradient, defined as

$$\epsilon_{\widetilde{\mathbf{u}},s} = \left[\sum_{E \in \Omega_h} h_E \sum_{e \in \partial E} \int_e (\nabla \boldsymbol{u} \cdot \boldsymbol{\tau}_e - G_h \boldsymbol{u}_h \cdot \boldsymbol{\tau}_e) \cdot (\nabla \boldsymbol{u} \cdot \boldsymbol{\tau}_e - G_h \boldsymbol{u}_h \cdot \boldsymbol{\tau}_e) \mathrm{d}e \right]^{1/2}. \tag{27}$$

4.2 | Topology-based data structure (TopS)

The adaptive mesh morphogenesis procedure demands an efficient data structure in order to handle adjacent information during the mesh modification events, that is, mesh clustering, node removal, treating skinny elements, and others. In this study, the topology-based data structure, named TopS,[62,63] is utilized to retrieve adjacency information. TopS consists of topological entities, for example, node, element, vertex, edge, and facet. The data structure explicitly represents nodes

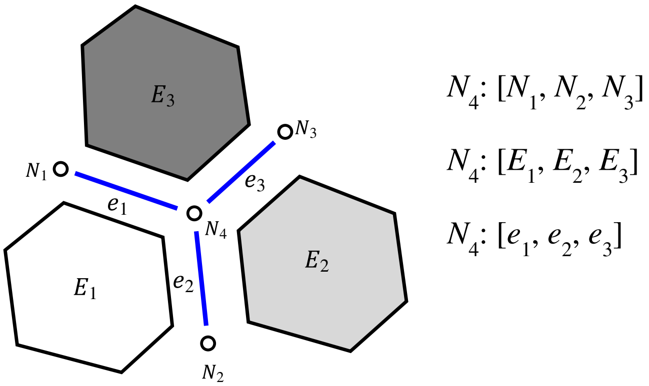


FIGURE 7 Schematics of topological entities for 2D polygonal elements [Colour figure can be viewed at wileyonlinelibrary.com]

and elements, while vertices, edges, and facets are implicitly represented. One advantage of TopS is that all the adjacent entities can be accessed from any given entity. For example, for a given node, one can identify all adjacent nodes, elements, and edges. As shown in Figure 7, the node N_4 has three types of adjacent entities, that is, nodes (N_1, N_2, N_3), elements (E_1, E_2, E_3), and edges (e_1, e_2, e_3). Note that edges are identical to facets in 2D while nodes and vertices are equivalent for linear elements. Additionally, a client-server approach is used to consistently maintain the data structure when mesh modification events occur.[64]

4.3 | Summary

The VEM analysis with the adaptive morphogenesis is outlined in Algorithm 1. Initially, one provides input data of a boundary value problem such as nodes, elements, material properties, and boundary conditions. Next, a system of equations is solved, and then the solution corresponds to a displacement field. Based on the evaluated displacement field, gradient errors of elements are estimated using the a posteriori estimator. Then, the adaptive morphogenesis is performed based on the estimated element errors. This process is recursively performed until the number of nodes (n_i) at the ith mesh generation is smaller than a given value ($n_{\min}$). When $n_i < n_{\min}$, a generated mesh is considered as sufficiently coarse, and then one terminates the analysis.

Algorithm 1. VEM analysis with the adaptive mesh morphogenesis

Input initial mesh, material properties and boundary conditions
Set $n_{\min}$ // $n_{\min}$: Minimum number of nodes for the generated meshes
$i = 0$, $n_i = n_0$ // n_0: Number of nodes at the initial mesh discretization
while ($n_i > n_{\min}$)
  Solve a system of equations
  Evaluate a posteriori gradient errors of each element ($\widetilde{\epsilon}_{\mathbf{u},s}\big|_E$)
  Perform the adaptive morphogenesis
  $i = i + 1$
  Update n_i // n_i: Number of nodes at the i^{th} generation
end while

The detailed procedure of the adaptive morphogenesis is described in Algorithm 2. The morphogenesis consists of four steps: (1) searching target elements, (2) mesh clustering, (3) edge straightening, (4) removing skinny elements, as discussed in Section 3.1. After finishing the morphogenesis, the clustering threshold θ_{clst} is adaptively updated according to the change of the number of nodes during the adaptive morphogenesis (see Algorithm 2). If the relative difference between the numbers of nodes of the ith and $i+1$th mesh generations is less than 0.5%, θ_{clst} increases to accelerate the mesh coarsening process. On the other hand, if the relative difference is larger than 5%, the value of θ_{clst} is initialized to prevent excessive coarsening over the whole domain. In this study, the initial value of the clustering threshold is set as 0.001. When the relative difference is less than 0.5%, the clustering threshold is enlarged with the increment of 0.001.

Algorithm 2. Procedure of adaptive mesh morphogenesis

1 : **Input:** Mesh at the ith generation, $\widetilde{\epsilon}_{\mathbf{u},s}\big|_E$, θ_{clst}, n_i
2 : Search target elements using the clustering criterion, i.e., $\widetilde{\epsilon}_{\mathbf{u},s}\big|_E \leq \theta_{\mathrm{clst}} \max(\widetilde{\epsilon}_{\mathbf{u},s}\big|_E)$
3 : Cluster the targeted elements together
4 : Perform the edge straightening
5 : Remove skinny elements
6 : Update the clustering threshold
6-1 : Evaluate n_{i+1}
6-2 : **if** $((n_i - n_{i+1})/n_{i+1} < 0.005)$ **then**
6-3 : Increase the clustering threshold θ_{clst}
6-4 : **else if** $((n_i - n_{i+1})/n_{i+1} > 0.05)$ **then**
6-5 : Initialize the clustering threshold θ_{clst}
6-6 : **else; continue; end if**
7 : **Output:** Mesh at the i+1th generation, θ_{clst}, n_{i+1}

During the morphogenesis, the target elements for clustering are identified on the basis of the a posteriori error estimation, as explained in Section 4.1. The detailed computational procedure for the error estimator is shown in Algorithm 3. First, a patch of elements (w_i) is defined for a given node ($\boldsymbol{x}_i$). Then, the reconstructed displacement field ($\mathbf{p}^i = [p_x^i, p_y^i]^{\mathrm{T}}$) is established as a linear combination of basis functions, given as

$$p_x^i = \mathbf{m}^w(\boldsymbol{x})\mathbf{q}^{w,x} \ \text{ and } \ p_y^i = \mathbf{m}^w(\boldsymbol{x})\mathbf{q}^{w,y}, \tag{28}$$

where $\mathbf{q}^{w,x} = [q_1^{w,x}, \cdots, q_6^{w,x}]^{\mathrm{T}}$ and $\mathbf{q}^{w,y} = [q_1^{w,y}, \cdots, q_6^{w,y}]^{\mathrm{T}}$ are the vectors containing the coefficients. The basis functions ($\mathbf{m}^w$) of $\mathbf{p}^i$ are assumed as second order polynomials, that is,

$$\mathbf{m}^w(\boldsymbol{x}) = \left[1 \left(\frac{x - x_w}{h_w}\right) \left(\frac{y - y_w}{h_w}\right) \left(\frac{(x - x_w)(y - y_w)}{{h_w}^2}\right) \left(\frac{x - x_w}{h_w}\right)^2 \left(\frac{y - y_w}{h_w}\right)^2\right], \tag{29}$$

where $\boldsymbol{x}_w = [x_w, y_w]^{\mathrm{T}}$ is the centroid of patch (w_i) and h_w is a characteristic size of the patch (w_i). By considering nodes ($\boldsymbol{x}_1, \boldsymbol{x}_2 \cdots \boldsymbol{x}_{n_w}$) in the patch ($w_i$), one can rewrite Equation (29) as a matrix form, which is given by,

$$\mathbf{P} = \begin{bmatrix} \mathbf{m}^w(\boldsymbol{x}_1) \\ \mathbf{m}^w(\boldsymbol{x}_2) \\ \vdots \\ \mathbf{m}^w(\boldsymbol{x}_{n_w}) \end{bmatrix}.$$

Then, one finds the coefficients of basis functions ($\mathbf{q}^{w,x}$, $\mathbf{q}^{w,y}$) through minimizing the residuals $\mathbf{r}_x$ and $\mathbf{r}_y$ which are expressed as

$$\mathbf{r}_x = \mathbf{P}\mathbf{q}^{w,x} - \mathbf{b}^{w,x} \ \text{ and } \ \mathbf{r}_y = \mathbf{P}\mathbf{q}^{w,y} - \mathbf{b}^{w,y}, \tag{31}$$

where $\mathbf{b}^{w,x} = [u_{h,x}(\boldsymbol{x}_1), \cdots, u_{h,x}(\boldsymbol{x}_{n_w})]^{\mathrm{T}}$ and $\mathbf{b}^{w,y} = [u_{h,y}(\boldsymbol{x}_1), \cdots, u_{h,y}(\boldsymbol{x}_{n_w})]^{\mathrm{T}}$ are vectors containing nodal displacements of VEM solution. To solve the minimization problem, one sets the normal equations up for $\mathbf{q}^{w,x}$ and $\mathbf{q}^{w,y}$ as follows,

$$(\mathbf{P}^{\mathrm{T}}\mathbf{P})\mathbf{q}^{w,x} = \mathbf{P}^{\mathrm{T}}\mathbf{b}^{w,x} \ \text{ and } \ (\mathbf{P}^{\mathrm{T}}\mathbf{P})\mathbf{q}^{w,y} = \mathbf{P}^{\mathrm{T}}\mathbf{b}^{w,y}. \tag{32}$$

After computing the vectors $\mathbf{q}^{w,x}$ and $\mathbf{q}^{w,y}$, the quadratic functions p_x^i and p_y^i can be evaluated using Equation (28). Finally, the nodal reconstructed displacement gradients $G_h\boldsymbol{u}_h(\boldsymbol{x}_i)$ are obtained according to Equation (24).

Algorithm 3. Element error evaluation using the a posteriori error estimator

1: **Input:** $\boldsymbol{u}_h$
2: Define a patch of elements (w_i) for each node $(\boldsymbol{x}_i)$
3: Compute a quadratic vector field $\mathbf{p}^i$ for each node
3-1: Set the basis functions and compute $\mathbf{P}$, $\mathbf{b}^{w,x}$, and $\mathbf{b}^{w,y}$
3-2: Evaluate the coefficient vectors, i.e., $\mathbf{q}^{w,x}$ and $\mathbf{q}^{w,y}$
3-3: Obtain the quadratic functions p_x^i and p_y^i
4: Evaluate nodal reconstructed gradients $G_h\boldsymbol{u}_h(\boldsymbol{x}_i)$
5: Compute $G_h\boldsymbol{u}_h$ on mesh skeleton
6: Compute the estimated element errors, $\widetilde{\epsilon}_{\mathbf{u},s}\big|_E$
7: **Output:** $\widetilde{\epsilon}_{\mathbf{u},s}\big|_E$

5 | NUMERICAL EXAMPLES

To verify the robustness and effectiveness of the adaptive morphogenesis, three numerical examples are illustrated. In the first example, one assumes an exact displacement field which includes sharp gradients. Next, two boundary value problems are solved, which have strong gradient and singularity, respectively. Note that plane strain condition is assumed, and consistent units are utilized for all the numerical examples.

5.1 | Prescribed displacement in an octagonal domain

In the octagonal domain illustrated in Figure 8A, an exact displacement field is assumed as

$$u_x = 16x(1-x)y(1-y)\mathrm{atan}\left(\frac{25(x-4y+2)}{2}\right), \tag{33}$$

$$u_y = 6xy, \tag{34}$$

where u_x and u_y are the displacement magnitudes in the x-y Cartesian coordinate system, respectively. Due to the arctangent term in Equation (33), the domain has a sharp gradient along the line $x-4y+2=0$, that is, the red line in Figure 8A. The elastic modulus and Poisson's ratios are selected as 1 and 0.3, respectively. In the numerical simulation, body forces, which satisfy equilibrium condition with the linear elastic assumption, are calculated and applied on the domain, while the assumed displacements are prescribed on the domain boundary. In this example, two types of discretizations are utilized, that is, centroid Voronoi tessellation (CVT) mesh, and structured mesh with quadrilateral and triangular elements, as shown in Figures 8B,C, respectively.

To check the accuracy of the a posteriori error estimator $\widetilde{\epsilon}_{\mathbf{u},s}$, one compares it with two exact error measures, that is, original error $\epsilon_{\mathbf{u},s}$, and recovered error $\epsilon_{\widetilde{\mathbf{u}},s}$ for uniform CVT and structured meshes. The number of nodes for each CVT mesh is 42, 255, 1001, 1997, 4981, 9932, 19 927, 49 813, and 79 714, while the number of nodes for each structured mesh is 37, 129, 481, 1057, 2092, 5212, 10 261, 19 137, and 52 117. Figure 9 illustrates that the reconstructed displacement gradient provides a higher rate of convergence than the gradient of the VEM solution, and the estimated error well agrees with the original error as the meshes are refined. Thus, one can say that the error estimator is accurate because the estimated error reproduces the original error.

Next, the validity of the error estimator is tested during the morphogenesis procedure on the CVT and structured meshes. At the initial discretization, the uniform CVT and structured meshes are utilized with the numbers of nodes are 49 813 and 52 117, respectively. The error estimator provides similar gradient errors to the original errors during the mesh generation, as shown in Figure 10. Although the estimated error is deviated from the original errors after certain mesh generation, the estimated error evolution is similar to that of the original error. Additionally, the errors of the uniform meshes are also plotted in Figure 10 for the comparison purpose. The computational results with the morphogenesis provide smaller error than the results from the uniform mesh. For example, to reach the estimated error level of 0.4, the minimum number of nodes for the uniform CVT mesh is approximately 31 800, while that for the morphogenesis is about

7800. In summary, the error estimator provides reasonable accuracy on the error evaluation and can be utilized on the proposed adaptive morphogenesis scheme. Furthermore, the morphogenesis scheme can effectively reduce the number of degrees of freedom in this example.

To demonstrate the change of discretization during the morphogenesis, one utilizes a uniform CVT mesh with 9932 nodes and a structured mesh with 10 261 nodes at the initial discretization. The generated meshes and their estimated element errors are illustrated in Figures 11 and 12. On the region with the sharp gradient, that is, along the line of $x-4y+2=0$, the initial fine elements are maintained because of relatively large errors. Coarse elements are generated on the upper and lower parts of the region where relatively smooth gradient is expected. Although some of clustered elements have relatively larger errors than initial fine elements, the global error of generated meshes is lower than that of the uniform meshes, as discussed previously. In summary, the error estimator accurately pinpoints the problematic regions, and the adaptive morphogenesis scheme generates coarse meshes when and where they are needed based on the estimated errors.

5.2 | Short cantilever

A short cantilever example[65,66] is employed, which has strong displacement gradients on the corners of the support region. The geometry and boundary conditions of the cantilever are illustrated in Figure 13A. The domain is fixed on the left edge and a distributed load of $q=1$ is applied along the top surface of the domain. The elastic modulus and Poisson's ratio are 1 and 0.3, respectively. In this numerical example, two types of discretizations are utilized, that is, CVT and rectangular meshes, as shown in Figures 13B,C.

To demonstrate the effectiveness of the proposed scheme, the computational results with the adaptive morphogenesis are compared with the results obtained from the uniform meshes. For the adaptive morphogenesis scheme, the numbers of nodes are 9983 and 10 201 for the CVT and rectangular meshes, respectively, at the initial discretization. Five uniform meshes are utilized for each discretization type. The number of nodes for each CVT mesh is 501, 1024, 2025, 5041, and 10 201, while the number of nodes for each rectangular mesh is 501, 996, 1996, 4994, and 9983. Figure 14 demonstrates that the computational results with the adaptive morphogenesis provide lower errors than the results with the uniform meshes for both CVT and rectangular mesh types. Additionally, during the adaptive morphogenesis procedure, the effects of the edge straightening (ES) on the estimated global error are investigated. When the edge straightening scheme is not employed, the estimated global error rapidly increases, and becomes higher than the error obtained from the uniform meshes. More advanced stabilization schemes[67] do not improve the trend of error evolution. For example, even if a matrix-based stabilization scheme[43] is used, the error evolution trend does not change in this study. Basically, the edge straightening in the morphogenesis improves the element quality, which reduces the estimated errors of coarsened elements.

The mesh generation and the estimated element errors during the morphogenesis are illustrated for the CVT mesh, rectangular mesh, and rectangular mesh without using ES, as shown in Figures 15, 16, and 17, respectively. For each

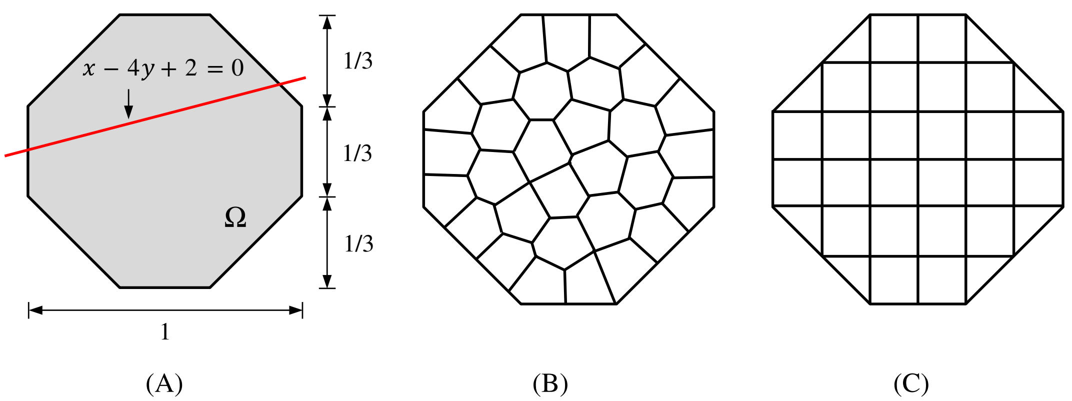


FIGURE 8 A, Geometry of the octagon domain. B, An example of the CVT mesh. C, An example of the structured mesh with quadrilateral and triangular elements [Colour figure can be viewed at wileyonlinelibrary.com]

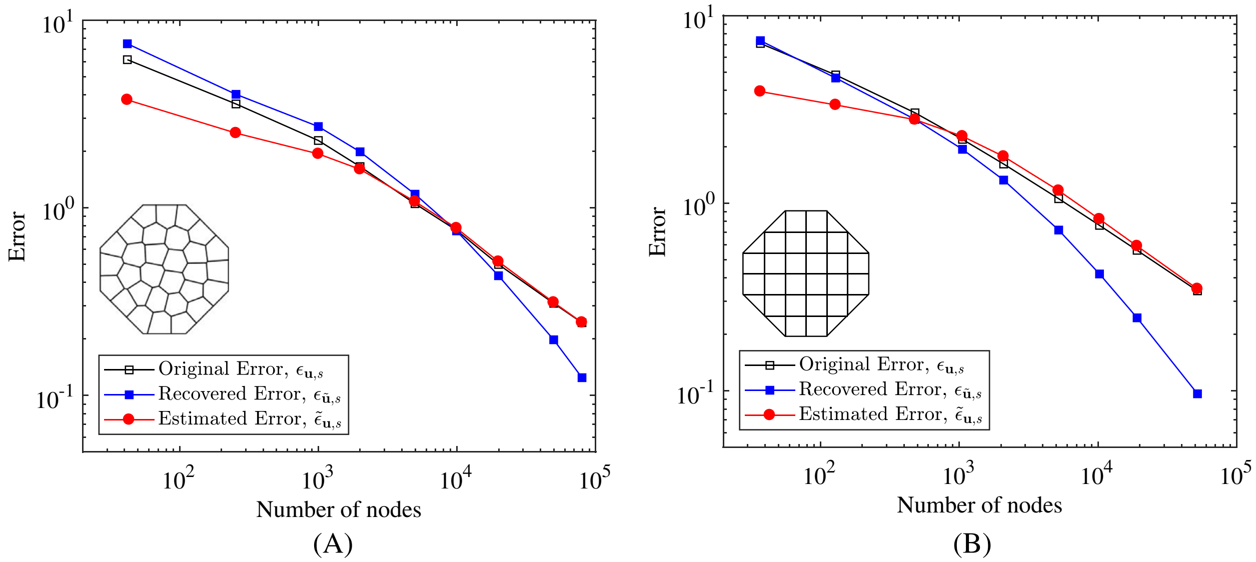


FIGURE 9 Comparison of the accuracy among the original errors ($\epsilon_{\mathbf{u},s}$), the recovered errors ($\epsilon_{\tilde{\mathbf{u}},s}$), and the estimated errors ($\widetilde{\epsilon}_{\mathbf{u},s}$) for A, uniform CVT meshes and B, uniform structured meshes [Colour figure can be viewed at wileyonlinelibrary.com]

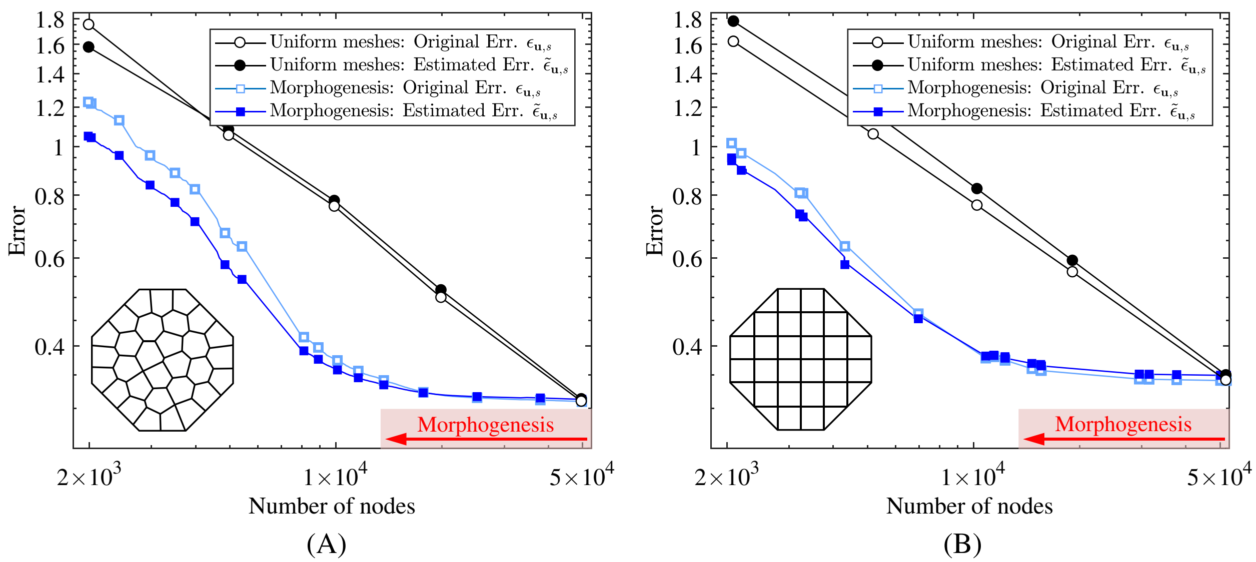


FIGURE 10 Global error evolution of the generated meshes in comparisons with those of uniform meshes: A, CVT meshes and B, structured meshes [Colour figure can be viewed at wileyonlinelibrary.com]

figure, three representative meshes are selected. During the morphogenesis, fine elements are remained on the top- and bottom-left corners which have strong gradients. Coarse elements are generated on the top- and bottom-right corners, and the center of the domain which have relatively lower errors than other regions. When the edge straightening scheme is not applied for the initial rectangular mesh (see Figure 17), all edges of the generated meshes only have horizontal or vertical directions. In addition, for the rectangular meshes, the mesh statistic and the global estimated errors during the adaptive morphogenesis are summarized in Table 1. For example, at the 12th morphogenesis, the number of nodes with ES decreases by 55%, while the global estimated error slightly increases by 7.4% compared with the initial discretization. The number of elements with ES is similar to the number of elements without ES, and the global errors are also similar to each other up to the 50th generation. However, the number of nodes without ES is larger than that with ES, which leads to the increase of the degrees of freedom.

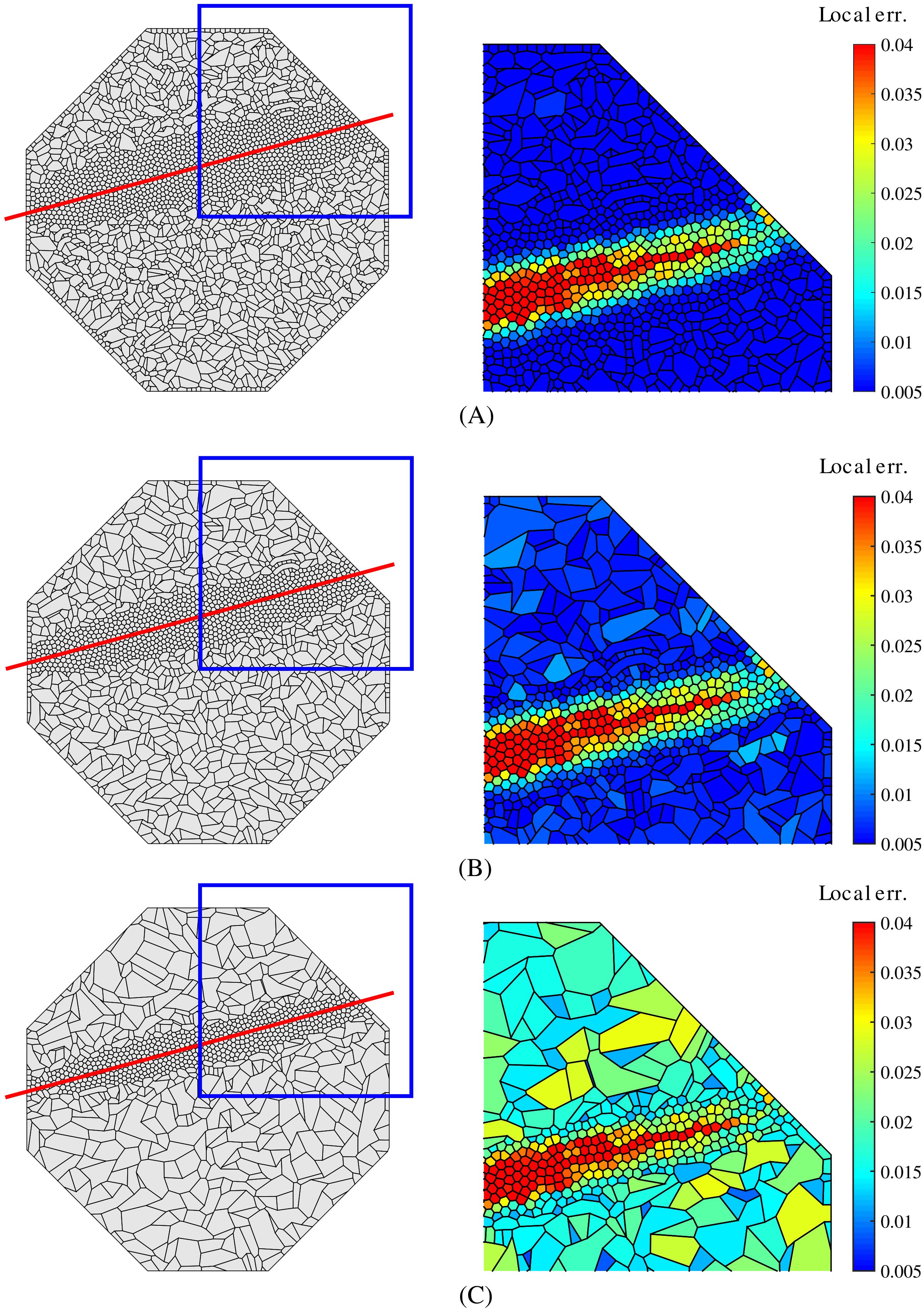


FIGURE 11 Generated meshes and element errors using the adaptive morphogenesis on the CVT mesh. The number of nodes of each figure is A, 5189, B, 3423, and C, 2093 [Colour figure can be viewed at wileyonlinelibrary.com]

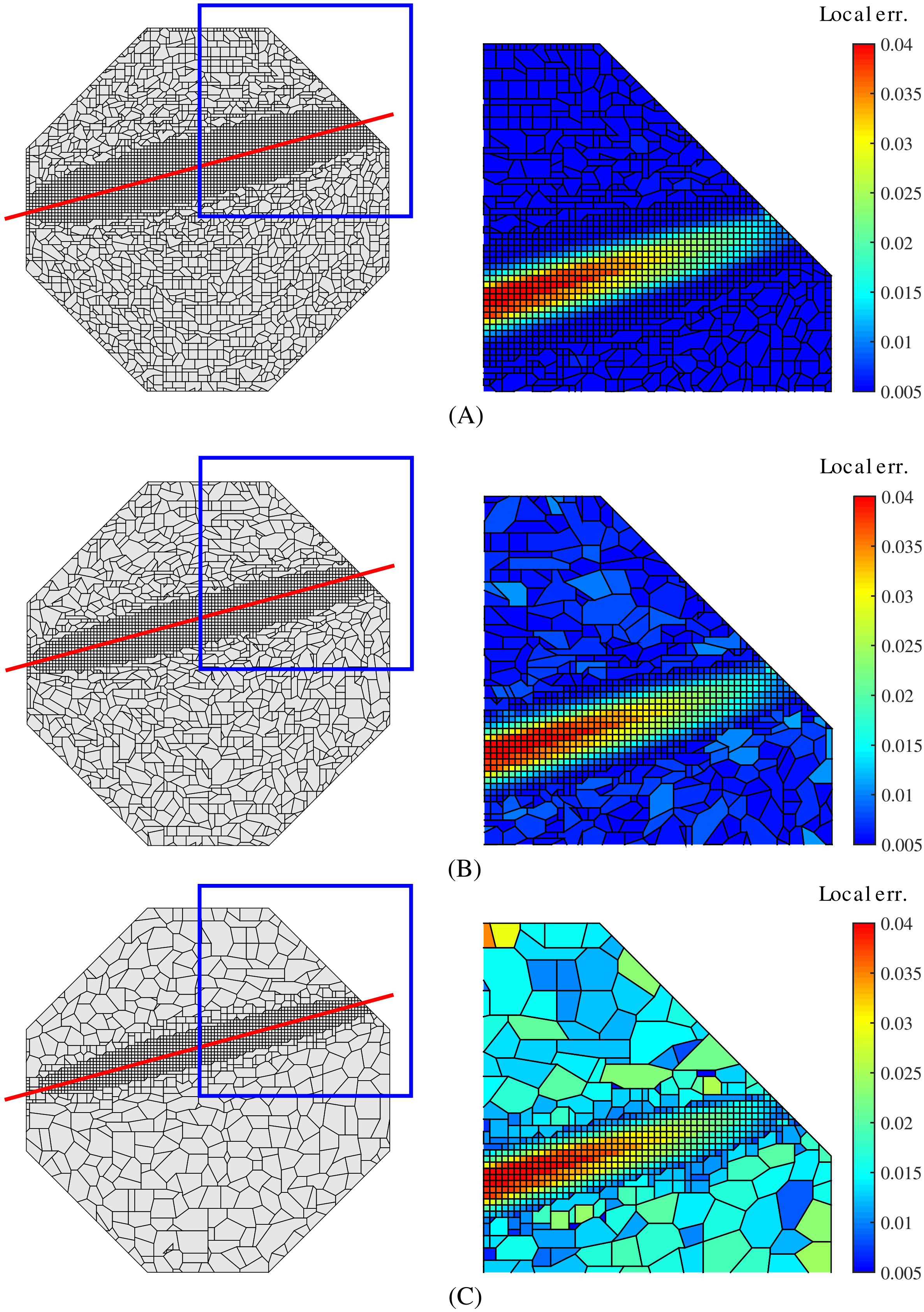


FIGURE 12 Generated meshes and element errors using the adaptive morphogenesis on the structured mesh. The number of nodes of each figure is A, 5038, B, 3428, and C, 2087 [Colour figure can be viewed at wileyonlinelibrary.com]

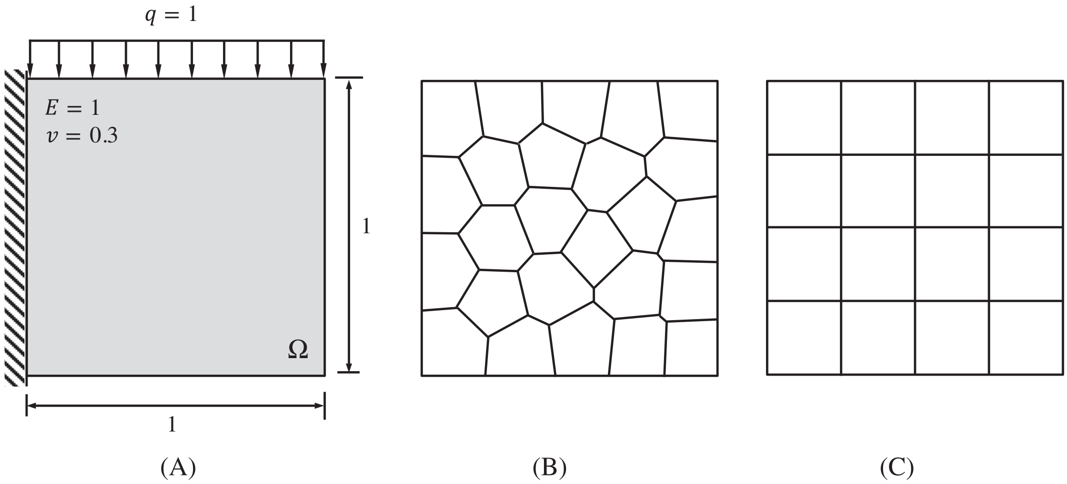


FIGURE 13 A, Geometry and boundary conditions of the short cantilever. B, An example of the CVT mesh. C, An example of the rectangular mesh

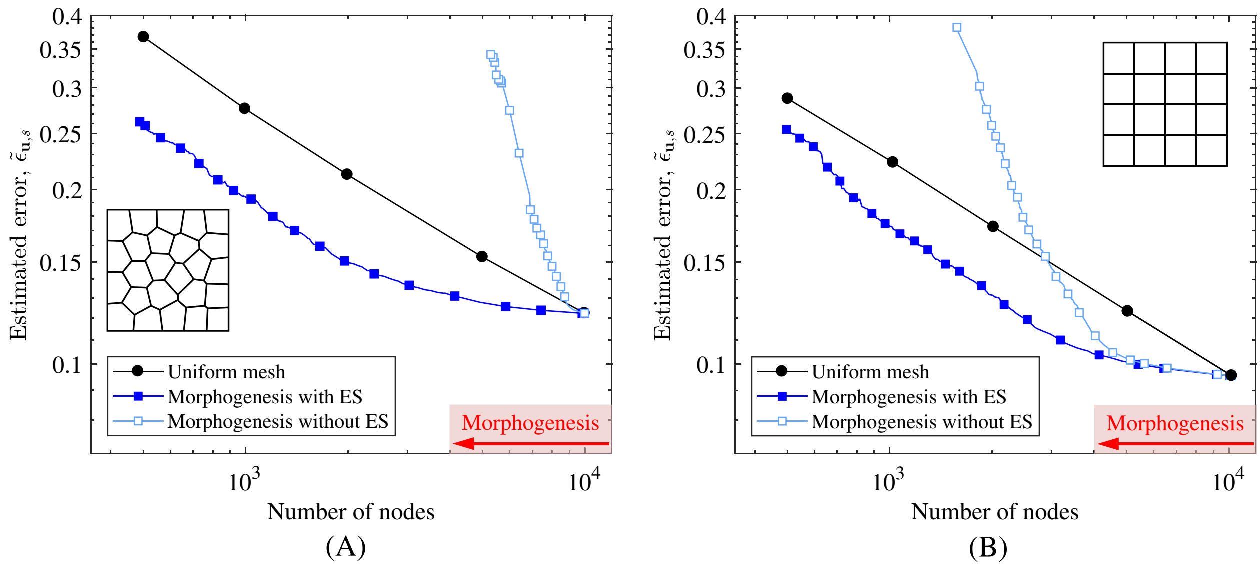


FIGURE 14 Effects of the edge straightening on the global error during the morphogenesis: A, CVT mesh and B, rectangular mesh [Colour figure can be viewed at wileyonlinelibrary.com]

5.3 | L-shaped beam

An L-shaped beam has a nonconvex corner which leads to a singularity in the displacement gradients. In the L-shaped beam, a uniform shear traction of $\tau = 1$ is applied on the right edge while the displacement is fixed on the top edge (see Figure 18). The elastic modulus is 10, and the Poisson's ratio is 0.35. Two types of discretizations, that is, CVT and rectangular meshes, are utilized, as in the previous examples. For initial meshes to perform the adaptive morphogenesis, two CVT meshes with 19 976 and 84 704 nodes, and two rectangular meshes with 17 176 and 77 441 nodes are employed.

The estimated errors obtained from the morphogenesis are compared with those from the uniform meshes. Figure 19 shows that the results from the morphogenesis have lower values of the global errors than the ones of uniform mesh. Although the morphogenesis starts from the different numbers of nodes at the initial discretization, the relation between the global error and the number of nodes becomes similar when the number of nodes is smaller than a certain level. In addition, the zooms of the CVT and rectangular meshes are illustrated according to the morphogenesis, as shown in Figures 20 and 21, respectively. On the reentrant corner which has singularity [Figures 20B and 21B], the initial fine elements are remained during the morphogenesis. Coarse elements are generated on the left- and right-bottom corners, for

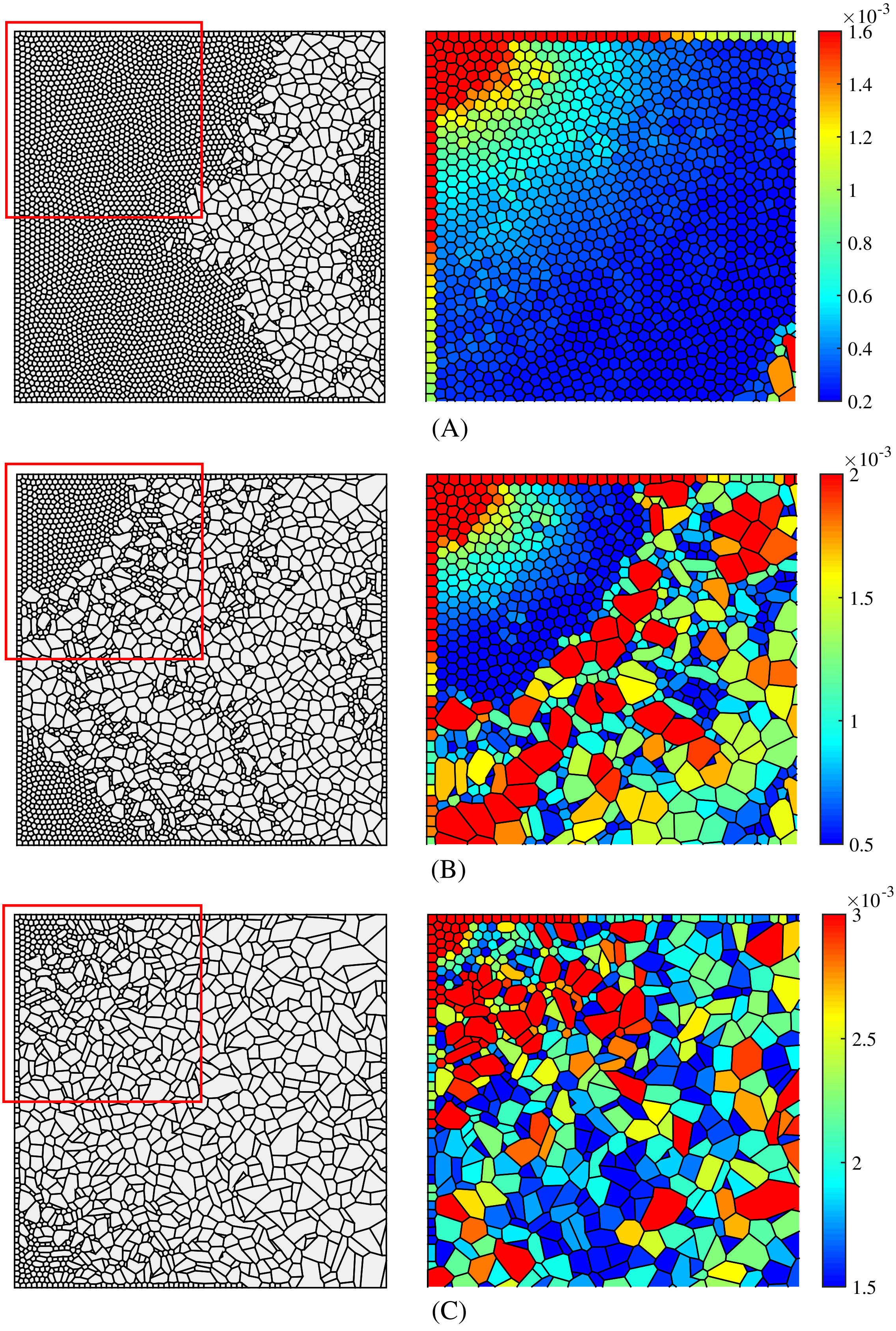


FIGURE 15 The mesh generation and estimated element errors during the morphogenesis on the CVT mesh when the edge straightening scheme is applied: A, fourth generation, B, 12th generation, and C, 50th generation [Colour figure can be viewed at wileyonlinelibrary.com]

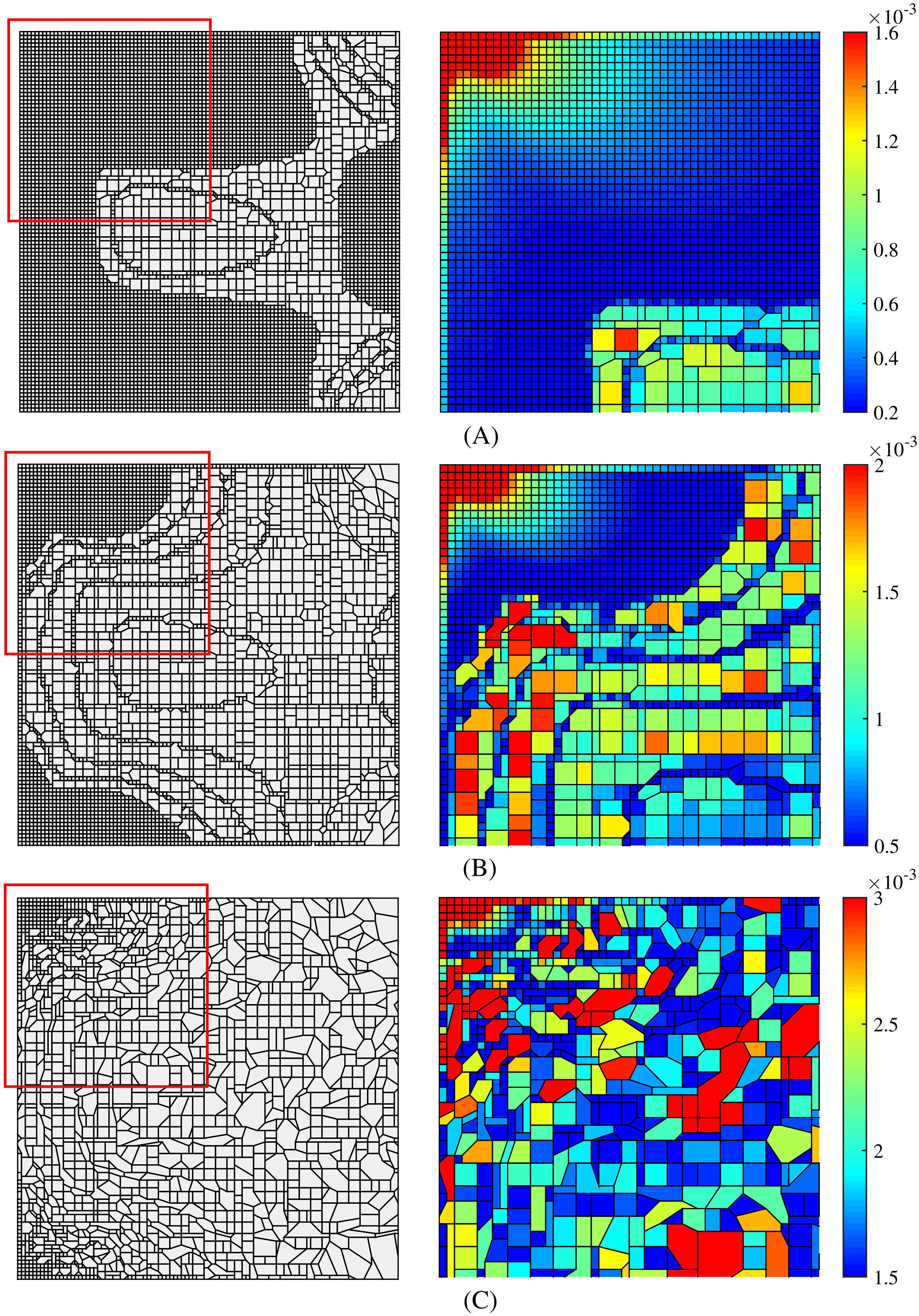


FIGURE 16 The mesh generation and estimated element errors during the morphogenesis on the rectangular mesh when the edge straightening scheme is applied: A, fourth generation, B, 12th generation, and C, 50th generation [Colour figure can be viewed at wileyonlinelibrary.com]

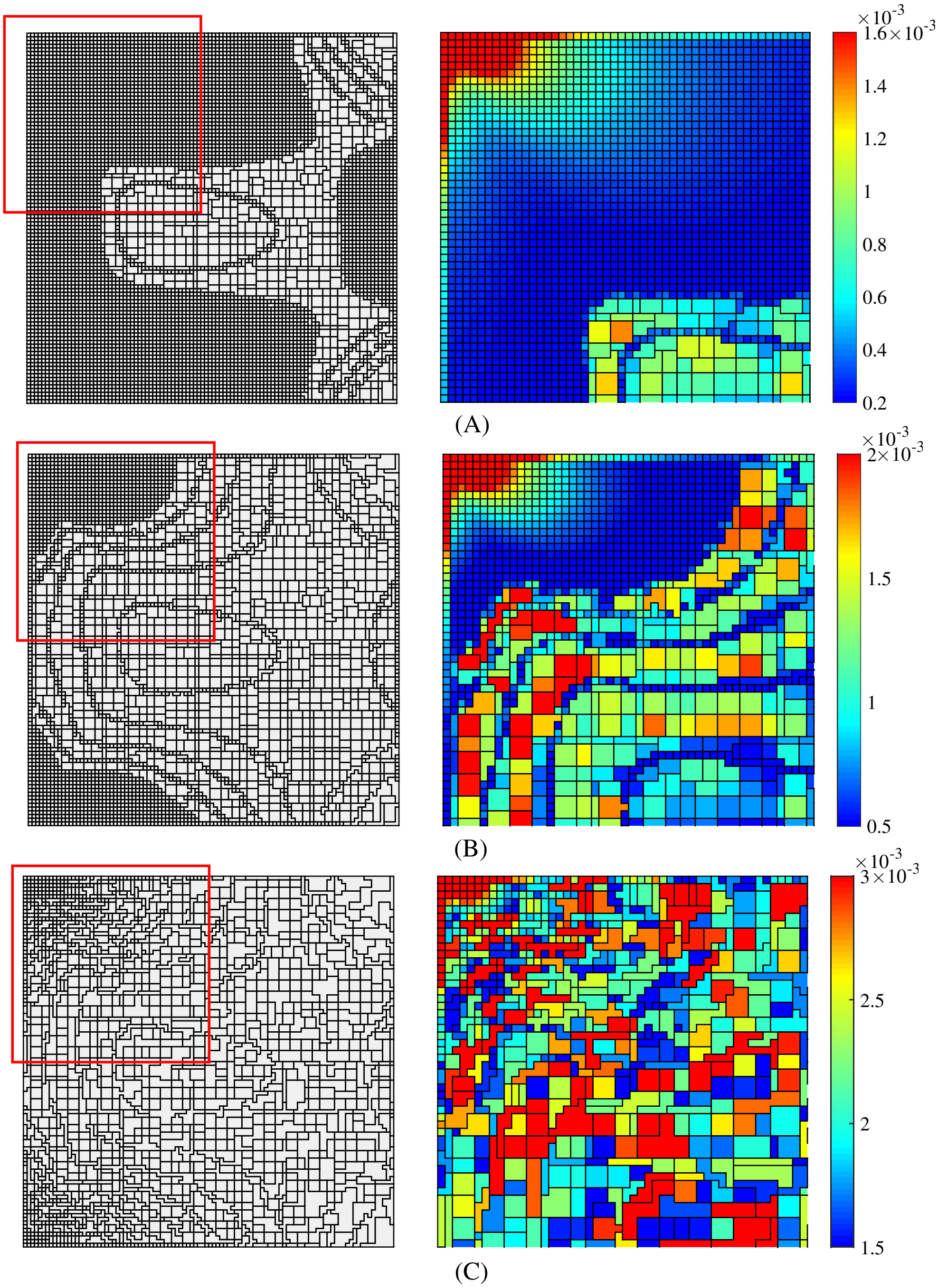


FIGURE 17 The mesh generation and estimated element errors during the morphogenesis on the rectangular mesh when the edge straightening scheme is not applied: A, fourth generation, B, 12th generation, and C, 50th generation [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Mesh statistics and the global estimated errors according to the adaptive morphogenesis for the rectangular meshes

		Use of edge straightening			Nonuse of edge straightening		
	Initial mesh	**fourth gen.**	**12th gen.**	**50th gen.**	**fourth gen.**	**12th gen.**	**50th gen.**
Number of elements	10 000	7274	3541	1674	7272	3546	1625
Number of nodes	10 201	7868	4568	2544	8033	4910	3546
Estimated error, $\tilde{\epsilon}_{\mathbf{u},s}$	0.095	0.097	0.102	0.119	0.098	0.103	0.127

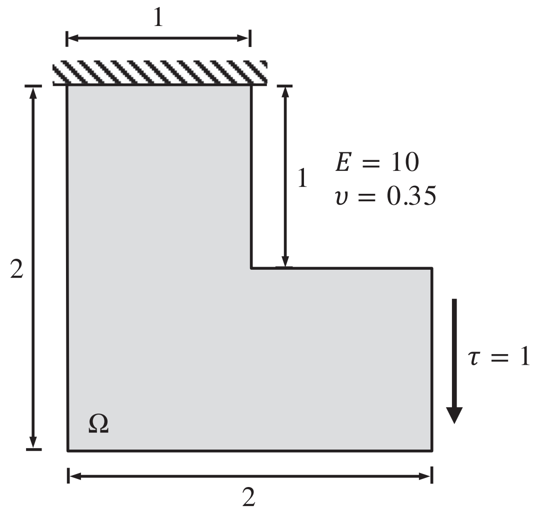


FIGURE 18 Domain description of the L-shaped beam

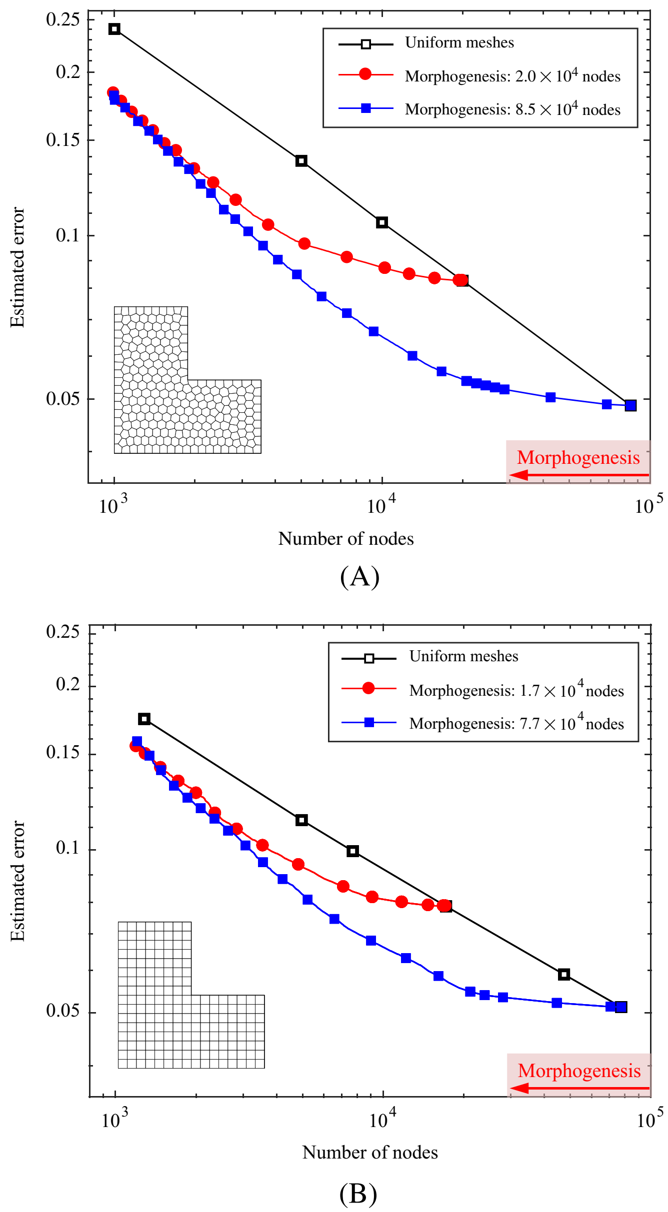


FIGURE 19 Global estimated errors of the generated meshes in comparisons with those of uniform meshes: A, CVT mesh and B, rectangular mesh [Colour figure can be viewed at wileyonlinelibrary.com]

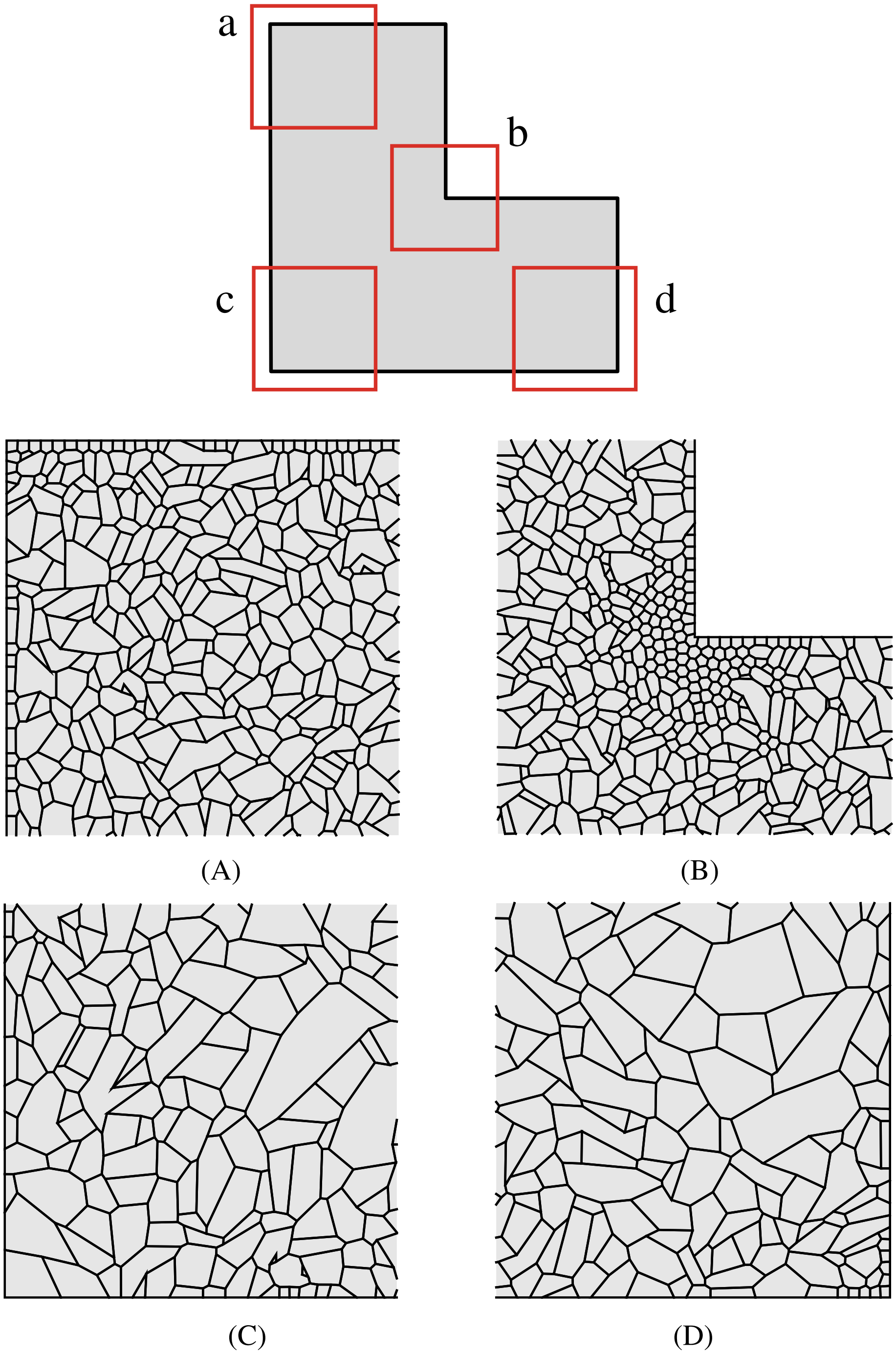


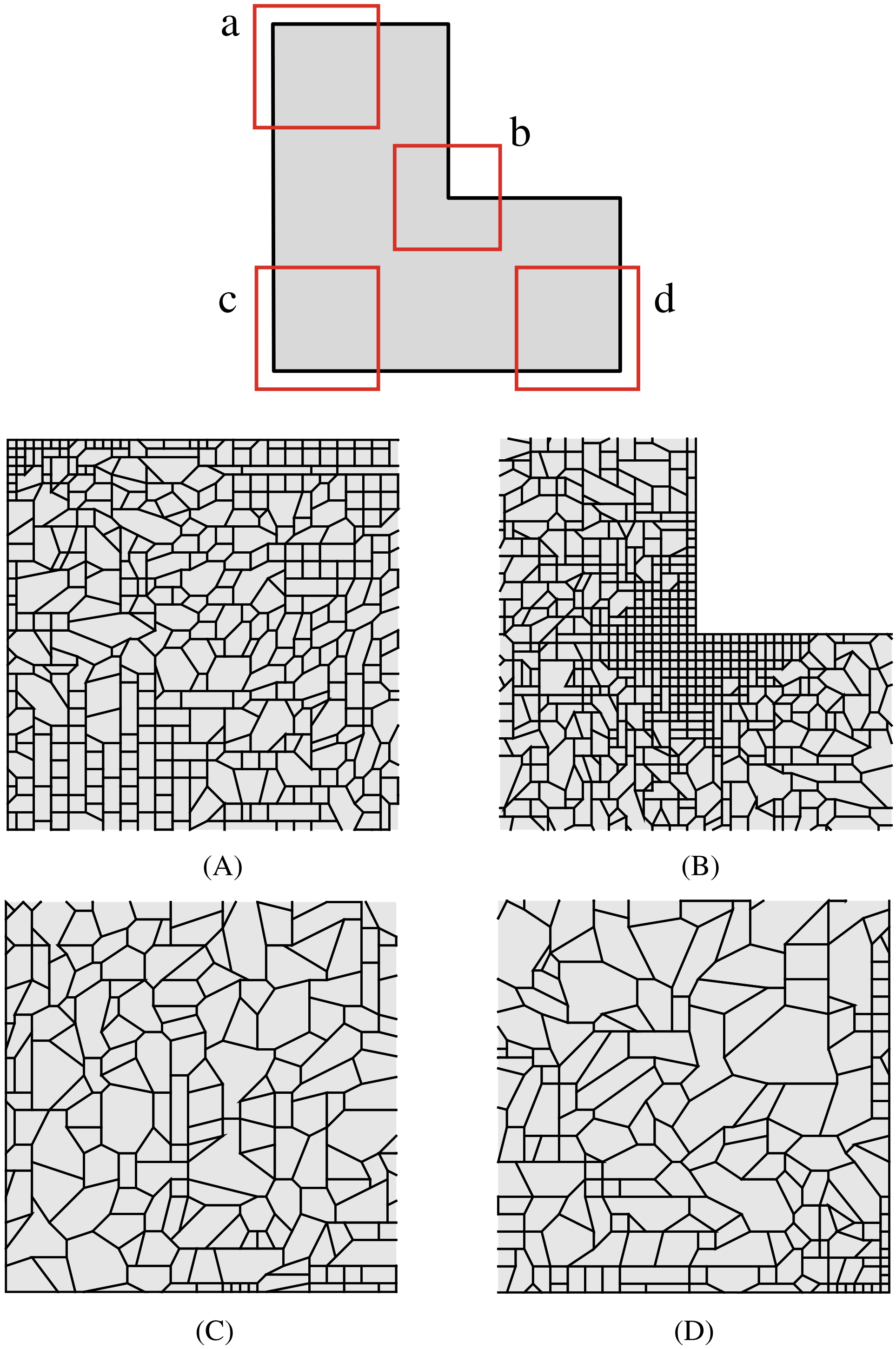
FIGURE 20 Zooms of the generated mesh according to the morphogenesis on the CVT mesh: A, top-left corner, B, reentrant corner, C, bottom-left corner, and D, bottom-right corner [Colour figure can be viewed at wileyonlinelibrary.com]

example, Figures 20C,D and 21C,D, where relatively low element errors are expected. In summary, the adaptive morphogenesis effectively reduces the number of nodes, while maintaining fine elements on the problematic regions to compute an accurate solution.

6 | CONCLUDING REMARKS AND EXTENSION

To investigate mesh coarsening for arbitrary unstructured meshes, the present study employs a VEM-based adaptive mesh morphogenesis strategy. Guided by an a posteriori error estimator, the elements which have low errors are searched, and merged with those adjacent elements. The main contributions of the present article are summarized as follows:

FIGURE 21 Zooms of the generated mesh according to the morphogenesis on the rectangular mesh: A, top-left corner, B, reentrant corner, C, bottom-left corner, and D, bottom-right corner [Colour figure can be viewed at wileyonlinelibrary.com]



- The adaptive mesh morphogenesis strategy consists of four steps: (1) identifying elements for coarsening, (2) clustering target elements, (3) edge straightening, and (4) eliminating skinny elements. One should note that the mesh morphogenesis strategy is not limited to the size or shape of the elements. In other words, skew and nonconvex elements can be utilized to represent mesh coarsening in the computational framework.
- To search elements which need coarsening, normalized errors of elements are evaluated using an a posteriori error estimator,[61] which is associated with displacement gradient errors based on the H^1-type skeletal norm. When the normalized errors are lower than a given threshold, the corresponding elements are classified as target elements for mesh clustering.

- The edge straightening scheme is introduced to improve the mesh quality of the clustered elements. When the edge straightening scheme is not applied during the adaptive morphogenesis procedure, the global errors of the generated meshes increase rapidly with coarsening. On the other hand, when the edge straightening scheme is employed, the meshes are generated with lower errors than those of uniform meshes.
- To demonstrate the proof-of-concept for the adaptive morphogenesis, element errors are arbitrarily defined within a rounded rectangular domain having "VEM" shaped holes. The computational results illustrate that the proposed scheme effectively coarsen unstructured meshes while maintaining fine meshes on high error regions.
- To verify the proposed computational framework, three numerical examples are simulated, which include sharp gradients or singularity on the displacement fields. The computational results demonstrate that the adaptive morphogenesis framework provides lower global errors than the results with uniform meshes. The problematic regions of the domain are captured by using the error estimator. Thus, fine elements are used in the problematic regions, while other regions are adaptively coarsened based on the estimated errors.

As indicated before, the morphogenesis strategy can be utilized with any feasible polygonal discretization in conjunction with reliable error estimators. Besides VEM, these methods include finite elements (polygonal based), MFD, DG, HDG, and HHO. Although the present study focuses on VEM, we hope that the morphogenesis strategy will be explored with different numerical methods such as the aforementioned ones. Besides mesh coarsening, the morphogenesis strategy can be explored in conjunction with mesh refinement, and also coupled with both coarsening and refinement. We hope that these schemes will be extended to three-dimensional problems. Finally, these ideas are promising and provide the basis for novel adaptive schemes in computational mechanics.

ACKNOWLEDGEMENTS

Habeun Choi and Kyoungsoo Park acknowledge the supports from the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (grant number: 2018R1A2B6007054), and from the Korea Institute of Energy Technology Evaluation and Planning (KETEP) funded by the Ministry of Trade, Industry & Energy (grant number: 20174030201480). Heng Chi and Glaucio H. Paulino acknowledge support from the Raymond Allen Jones Chair at the Georgia Institute of Technology.

ORCID

Habeun Choi https://orcid.org/0000-0001-7863-110X
Heng Chi https://orcid.org/0000-0003-3331-0614
Kyoungsoo Park https://orcid.org/0000-0001-6243-5016
Glaucio H. Paulino https://orcid.org/0000-0002-3493-6857

REFERENCES

1. Babuška I, Rheinboldt WC. Adaptive approaches and reliability estimations in finite element analysis. *Comput Methods Appl Mech Eng*. 1979;17:519-540.
2. Babuška I, Rheinboldt WC. A-posteriori error estimates for the finite element method. *Int J Numer Methods Eng*. 1978;12(10):1597-1615.
3. Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. *Int J Numer Methods Eng*. 1987;24(2):337-357.
4. Batra R, Ko K-I. An adaptive mesh refinement technique for the analysis of shear bands in plane strain compression of a thermoviscoplastic solid. *Comput Mech*. 1992;10(6):369-379.
5. Borouchaki H, Laug P, Cherouat A, Saanouni K. Adaptive remeshing in large plastic strain with damage. *Int J Numer Methods Eng*. 2005;63(1):1-36.
6. Molinari J, Ortiz M. Three-dimensional adaptive meshing by subdivision and edge-collapse in finite-deformation dynamic–plasticity problems with application to adiabatic shear banding. *Int J Numer Methods Eng*. 2002;53(5):1101-1126.
7. Alhadeff A, Leon SE, Celes W, Paulino GH. Massively parallel adaptive mesh refinement and coarsening for dynamic fracture simulations. *Eng Comput*. 2016;32(3):533-552.
8. Park K, Paulino GH, Celes W, Espinha R. Adaptive mesh refinement and coarsening for cohesive zone modeling of dynamic fracture. *Int J Numer Methods Eng*. 2012;92(1):1-35.
9. Spring DW, Leon SE, Paulino GH. Unstructured polygonal meshes with adaptive refinement for the numerical simulation of dynamic cohesive fracture. *Int J Fract*. 2014;189(1):33-57.
10. Chen L, Lingen EJ, de Borst R. Adaptive hierarchical refinement of NURBS in cohesive fracture analysis. *Int J Numer Methods Eng*. 2017;112(13):2151-2173.

11. Chau KN, Chau KN, Ngo T, Hackl K, Nguyen-Xuan H. A polytree-based adaptive polygonal finite element method for multi-material topology optimization. *Comput Methods Appl Mech Eng*. 2018;332:712-739.
12. Costa JCA Jr, Alves MK. Layout optimization with h-adaptivity of structures. *Int J Numer Methods Eng*. 2003;58(1):83-102.
13. Lambe AB, Czekanski A. Topology optimization using a continuous density field and adaptive mesh refinement. *Int J Numer Methods Eng*. 2018;113(3):357-373.
14. Zhang XS, Chi H, Paulino GH. Adaptive multi-material topology optimization with hyperelastic materials under large deformations: A virtual element approach. *Comput Methods Appl Mech Eng*. 2020;370:112976.
15. Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *J Comput Phys*. 1989;82(1):64-84.
16. Balsara DS. Divergence-free adaptive mesh refinement for magnetohydrodynamics. *J Comput Phys*. 2001;174(2):614-648.
17. Botti L, Piccinelli M, Ene-Iordache B, Remuzzi A, Antiga L. An adaptive mesh refinement solver for large-scale simulation of biological flows. *Int J Numer Methods Biomed Eng*. 2010;26(1):86-100.
18. Bayona-Roa C, Codina R, Baiges J. Variational multiscale error estimators for the adaptive mesh refinement of compressible flow simulations. *Comput Methods Appl Mech Eng*. 2018;337:501-526.
19. Neumann P, Neckel T. A dynamic mesh refinement technique for lattice Boltzmann simulations on octree-like grids. *Comput Mech*. 2013;51(2):237-253.
20. Kirk BS, Peterson JW, Stogner RH, Carey GF. libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Eng Comput*. 2006;22(3–4):237-254.
21. Bassi F, Botti L, Colombo A. Agglomeration-based physical frame dG discretizations: an attempt to be mesh free. *Math Models Methods Appl Sci*. 2014;24(08):1495-1539.
22. Bassi F, Botti L, Colombo A, Di Pietro DA, Tesini P. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. *J Comput Phys*. 2012;231(1):45-65.
23. Antonietti PF, Pennesi G. V-cycle multigrid algorithms for discontinuous Galerkin methods on non-nested polytopic meshes. *J Sci Comput*. 2019;78(1):625-652.
24. Sukumar N, Tabarraei A. Conforming polygonal finite elements. *Int J Numer Methods Eng*. 2004;61(12):2045-2066.
25. Chi H, Talischi C, Lopez-Pamies O, Paulino GH. Polygonal finite elements for finite elasticity. *Int J Numer Methods Eng* 2015;101(4):305–328.
26. Beirão da Veiga L, Brezzi F, Cangiani A, Manzini G, Marini LD, Russo A. Basic principles of virtual element methods. *Math Models Methods Appl Sci*. 2013;23(01):199-214.
27. Beirao da Veiga L, Lipnikov K, Manzini G. *The Mimetic Finite Difference Method for Elliptic Problems*. Heidelberg: Springer; 2014.
28. Kuznetsov Y, Lipnikov K, Shashkov M. The mimetic finite difference method on polygonal meshes for diffusion-type problems. *Comput Geosci*. 2004;8(4):301-324.
29. Lipnikov K, Manzini G, Shashkov M. Mimetic finite difference method. *J Comput Phys*. 2014;257:1163-1227.
30. Antonietti PF, Houston P, Hu X, Sarti M, Verani M. Multigrid algorithms for hp-version interior penalty discontinuous Galerkin methods on polygonal and polyhedral meshes. *Cal*. 2017;54(4):1169-1198.
31. Cangiani A, Georgoulis EH, Houston P. Hp-version discontinuous Galerkin methods on polygonal and polyhedral meshes. *Math Models Methods Appl Sci*. 2014;24(10):2009-2041.
32. Cockburn B, Gopalakrishnan J, Lazarov R. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J Numer Anal*. 2009;47(2):1319-1365.
33. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–stokes equations. *J Comput Phys*. 2011;230(4):1147-1170.
34. Di Pietro DA, Ern A. A hybrid high-order locking-free method for linear elasticity on general meshes. *Comput Methods Appl Mech Eng*. 2015;283:1-21.
35. Zhang Y, Mei L. A hybrid high-order method for a class of quasi-Newtonian stokes equations on general meshes. *Appl Math Comput*. 2020;366:124741.
36. Beirão da Veiga L, Brezzi F, Marini LD, Russo A. Serendipity nodal VEM spaces. *Comput Fluids*. 2016;141:2-12.
37. Park K, Chi H, Paulino GH. On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration. *Comput Methods Appl Mech Eng*. 2019;356:669-684.
38. Paulino GH, Gain AL. Bridging art and engineering using Escher-based virtual elements. *Struct Multidiscip Optim*. 2015;51(4):867-883.
39. Beirão da Veiga L, Brezzi F, Marini LD, Russo A. The hitchhiker's guide to the virtual element method. *Math Models Methods Appl Sci*. 2014;24(08):1541-1573.
40. Beirão da Veiga L, Brezzi F, Marini LD. Virtual elements for linear elasticity problems. *SIAM J Numer Anal*. 2013;51(2):794-812.
41. Gain AL, Talischi C, Paulino GH. On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Comput Methods Appl Mech Eng*. 2014;282:132-160.
42. Zhang B, Yang Y, Feng M. Mixed virtual element methods for elastodynamics with weak symmetry. *J Comput Appl Math*. 2019;353:49-71.
43. Park K, Chi H, Paulino GH. Numerical recipes for elastodynamic virtual element methods with explicit time integration. *Int J Numer Methods Eng*. 2020;121(1):1-31.
44. Chi H, Beirão da Veiga L, Paulino GH. Some basic formulations of the virtual element method (VEM) for finite deformations. *Comput Methods Appl Mech Eng*. 2017;318:148-192.
45. Beirão da Veiga L, Lovadina C, Mora D. A virtual element method for elastic and inelastic problems on polytope meshes. *Comput Methods Appl Mech Eng*. 2015;295:327-346.

46. Artioli E, Beirão da Veiga L, Lovadina C, Sacco E. Arbitrary order 2D virtual elements for polygonal meshes: part II, inelastic problem. *Comput Mech*. 2017;60(4):643-657.
47. Aldakheel F, Hudobivnik B, Hussein A, Wriggers P. Phase-field modeling of brittle fracture using an efficient virtual element scheme. *Comput Methods Appl Mech Eng*. 2018;341:443-466.
48. Benedetto MF, Caggiano A, Etse G. Virtual elements and zero thickness interface-based approach for fracture analysis of heterogeneous materials. *Comput Methods Appl Mech Eng*. 2018;338:41-67.
49. Nguyen-Thanh VM, Zhuang X, Nguyen-Xuan H, Rabczuk T, Wriggers P. A virtual element method for 2D linear elastic fracture analysis. *Comput Methods Appl Mech Eng*. 2018;340:366-395.
50. Hussein A, Aldakheel F, Hudobivnik B, Wriggers P, Guidault P-A, Allix O. A computational framework for brittle crack propagation based on an efficient virtual element method. *Finite Elements Anal Des*. 2019;159:15-32.
51. Antonietti P, Beirão da Veiga L, Mora D, Verani M. A stream function formulation of the stokes problem for the virtual element method. *SIAM J Numer Anal*. 2014;52:386-404.
52. Antonietti P, Bruggi M, Scacchi S, Verani M. On the virtual element method for topology optimization on polygonal meshes: a numerical study. *Comput Math Appl*. 2017;74(5):1091-1109.
53. Gain AL, Paulino GH, Duarte LS, Menezes IF. Topology optimization using polytopes. *Comput Methods Appl Mech Eng*. 2015;293:411-430.
54. Chi H, Pereira A, Menezes IF, Paulino GH. Virtual element method (VEM)-based topology optimization: an integrated framework. *Struct Multidiscip Optim*. 2020;62:1089-1114.
55. Martin S, Kaufmann P, Botsch M, Wicke M, Gross M. Polyhedral finite elements using harmonic basis functions. *Comput Graph Forum*. 2008;27(5):1521-1529.
56. Bishop JE. A displacement-based finite element formulation for general polyhedra using harmonic shape functions. *Int J Numer Methods Eng*. 2014;97(1):1-31.
57. Rashid M, Gullett P. On a finite element method with variable element topology. *Comput Methods Appl Mech Eng*. 2000;190(11–12):1509-1527.
58. Dohrmann CR, Rashid M. Polynomial approximation of shape function gradients from element geometries. *Int J Numer Methods Eng*. 2002;53(4):945-958.
59. Sukumar N. Quadratic maximum-entropy serendipity shape functions for arbitrary planar polygons. *Comput Methods Appl Mech Eng*. 2013;263:27-41.
60. Hormann K, Sukumar N. Maximum entropy coordinates for arbitrary polytopes. *Comput Graph Forum*. 2008;27(5):1513-1520.
61. Chi H, Beirão da Veiga L, Paulino GH. A simple and effective gradient recovery scheme and a posteriori error estimator for the virtual element method (VEM). *Comput Methods Appl Mech Eng*. 2019;347:21-58.
62. Celes W, Paulino GH, Espinha R. A compact adjacency-based topological data structure for finite element mesh representation. *Int J Numer Methods Eng*. 2005;64(11):1529-1556.
63. Celes W, Paulino GH, Espinha R. Efficient handling of implicit entities in reduced mesh representations. *J Comput Inf Sci Eng*. 2005;5(4):348-359.
64. Paulino GH, Park K, Celes W, Espinha R. Adaptive dynamic cohesive fracture simulation using nodal perturbation and edge-swap operators. *Int J Numer Methods Eng*. 2010;84(11):1303-1343.
65. Zhu J, Zienkiewicz O. Adaptive techniques in the finite element method. *Commun Appl Numer Methods*. 1988;4(2):197-204.
66. Cook RD. *Finite Element Modeling for Stress Analysis*. New York: John Wiley & Sons; 1995.
67. Beirão da Veiga L, Vacca G. Sharper error estimates for virtual elements and a bubble-enriched version. *arXiv Preprint arXiv*. 2020;12009.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Choi H, Chi H, Park K, Paulino GH. Computational Morphogenesis: Morphologic constructions using polygonal discretizations. *Int J Numer Methods Eng*. 2021;122:25–52. https://doi.org/10.1002/nme.6519