INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2010; **82**:671–698
Published online 3 December 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2763

Polygonal finite elements for topology optimization: A unifying paradigm

Cameron Talischi[1], Glaucio H. Paulino[1,*,†], Anderson Pereira[2] and Ivan F. M. Menezes[2]

[1]*Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, IL, U.S.A.*
[2]*Tecgraf, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil*

SUMMARY

In topology optimization literature, the parameterization of design is commonly carried out on uniform grids consisting of Lagrangian-type finite elements (e.g. linear quads). These formulations, however, suffer from numerical anomalies such as checkerboard patterns and one-node connections, which has prompted extensive research on these topics. A problem less often noted is that the constrained geometry of these discretizations can cause bias in the orientation of members, leading to mesh-dependent sub-optimal designs. Thus, to address the geometric features of the spatial discretization, we examine the use of unstructured meshes in reducing the influence of mesh geometry on topology optimization solutions. More specifically, we consider polygonal meshes constructed from Voronoi tessellations, which in addition to possessing higher degree of geometric isotropy, allow for greater flexibility in discretizing complex domains without suffering from numerical instabilities. Copyright © 2009 John Wiley & Sons, Ltd.

Received 7 April 2009; Revised 26 August 2009; Accepted 31 August 2009

KEY WORDS: topology optimization; polygonal elements; Voronoi tessellations; unstructured meshing

1. INTRODUCTION AND MOTIVATION

The goal of topology optimization is to find the most efficient distribution of a fixed volume of material in a specified design domain, without violating user-defined design constraints. The computational solution procedure for solving such problems begins with a spatial discretization of the design domain, based on which the set of design variables are defined (e.g. element volume fractions) and candidate designs are analyzed. If additional constraints are not imposed on the

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

Contract/grant sponsor: Office of Science and National Nuclear Security Administration; contract/grant number: DE-FG02-97ER25308
Contract/grant sponsor: Tecgraf (Group of Technology in Computer Graphics), PUC-Rio, Rio de Janeiro, Brazil

problem, the topology optimization solutions may not be convergent under mesh refinement because the oscillation of the design field is limited only by the grid scale. For example, the minimum compliance problem does not have a physical length scale, and is ill-posed in the continuum setting (for a review, see Reference [1]). Therefore, one would obtain designs with finer features as the mesh is refined. The issue of mesh dependency can be treated by imposing restrictions on the continuum problem. For example, it has been shown that placing an upper bound on the total variation of the design field, corresponding to the perimeter of the design, guarantees the existence of an optimal solution [2, 3]. However, Petersson [4] points out that the standard finite element formulation for the perimeter-controlled method on structured square meshes suffers from a 'rotational mesh-dependence', which causes the members to align with the element edges. Therefore, one cannot expect convergence of numerical results to the optimal solution. Here lies *another form of mesh dependency* that stems from the geometric features of the spatial discretization. Thus, there is need for better computational framework and techniques.

To simplify the discussion and avoid the ill-posed nature of the continuum problem, consider the discrete topology optimization problem defined on a fixed mesh. Assuming that the design variables take discrete values (what is often desired), then an optimal solution exists. Since the geometry of the mesh dictates the possible layout of material and orientation of members, one expects the solution obtained on a less-biased mesh to be better. Moreover, if the geometric attributes of the mesh are too restricting, certain characteristic patterns of the optimal solution (e.g. orthogonality of members in classic Michell problems) may be excluded from the final design. Similarly, artificial features specific to the choice of discretization may be introduced. A commonly encountered artifact is the one-node hinge that spuriously improves the performance of compliant mechanisms [5]. Therefore, it is important to examine the influence of geometric properties of the spatial discretization on topology optimization solutions. We remark that such investigations have been conducted in computational fracture mechanics, in particular, for crack propagation simulations, where quantities of interest such as crack path and crack length are sensitive to the mesh geometry. Park *et al.* [6] employed topological operators such as edge-swap and nodal perturbation to avoid undesirable crack patterns in 4K structured meshes. Papoulia *et al.* [7] used a 2D pinwheel mesh with the isoperimetric property to ensure convergence of crack path, while Bolander and Saito [8] proposed fracture simulations on spring networks with random geometry that maximize directional isotropy for crack propagation.

Another limitation of uniform grids is the difficulty in discretizing complex design domains and accurately representing loading and support boundary conditions. In finite element applications, complex domains are usually discretized using triangular or tetrahedral elements. However, such meshes exhibit one-node connections and are susceptible to checkerboard patterns in topology optimization applications. Indeed, one can use regularization schemes such as filtering to suppress the numerical instabilities, but these measures often involve heuristic parameters that can augment the optimization problem. Moreover, as Rozvany *et al.* [9] argue, these schemes can lead to significant weight increases. Therefore, there remains interest in obtaining checkerboard-free results without placing additional constraints. *Polygonal elements can be very useful in this respect since they naturally exclude checkerboard layouts* [10, 11], *and provide flexibility in discretizing complex domains* (*see Section 2*).

In this work, Voronoi diagrams are used as a natural and effective means for generating irregular polygonal meshes. These diagrams have been studied extensively in mathematics, computer science, and natural sciences, and the interested reader is referred to [12] for a survey on the topic. We adopt an extension of the scheme proposed in [8, 13] for meshing arbitrary two or three-dimensional

DOI: 10.1002/nme

domains using Voronoi tessellations. An attractive feature of the method is that randomness and subsequently higher levels of geometric isotropy are obtained as a byproduct of arbitrary seed placement. Furthermore, the use of Lloyd's algorithm [14] can remove excessive element distortion, and allows for construction of meshes that are uniform in size.

The remainder of this paper is organized as follows. In Section 2, we discuss the approach for generating Voronoi meshes. We review the finite element formulation for convex polygons in Section 3, and present the topology optimization solution scheme in Section 4. Next, we address some implementation issues in Section 5, and show several numerical results in Section 6. Finally, we conclude with some remarks in Section 7.

2. MESHING USING VORONOI TESSELLATIONS

In this work, we extend the method proposed in [8, 13] for generating polygonal meshes using Voronoi diagrams. The main idea is to choose a set of generating points or *seeds* such that the corresponding Voronoi tessellation incorporates the boundary of the domain. By requiring the Voronoi diagram to be centroidal, one can obtain high-quality meshes.

2.1. Concepts and definitions

Consider the set of seeds $P=\{\boldsymbol{p}_i\}_{i=1}^n$ in domain $\Omega\subseteq\mathbb{R}^d$. The *Voronoi tessellation* of P, denoted by $\mathscr{V}(P)$, is the partitioning of the domain Ω into n regions defined by

$$V_i=\bigcap_{\forall j,j\neq i}\{\boldsymbol{x}\in\Omega|\delta(\boldsymbol{x},\boldsymbol{p}_i)\leqslant\delta(\boldsymbol{x},\boldsymbol{p}_j)\},\quad i=1,\ldots,n \tag{1}$$

where $\delta(\cdot,\cdot)$ is the Euclidean distance [15]. That is, the *Voronoi cell* V_i is composed of points that are closer to $\boldsymbol{p}_i$ than any other point in P. Note that each V_i is non-empty since $\boldsymbol{p}_i\in V_i$. If Ω is unbounded (e.g. $\Omega=\mathbb{R}^2$), some cells are unbounded because the number of Voronoi cells is equal to the number of seeds. Note also that the bounded Voronoi cells are necessarily convex polygons as they are formed by finite intersection of half-planes. This is important because the finite element formulation presented in the next section is restricted to convex polygonal discretizations. A number of efficient algorithms are available that construct Voronoi tessellations directly, or based on its dual, the Delaunay triangulation of the same point set [15], rendering this fundamental geometric construct an attractive tool for mesh generation.

A Voronoi tessellation is *centroidal* if each generating point coincides with the centroid of the corresponding Voronoi cell, i.e.

$$\boldsymbol{p}_i=\overline{\boldsymbol{p}}_i\quad\forall i=1,\ldots,n\quad\text{where }\overline{\boldsymbol{p}}_i:=\frac{\int_{V_i}\boldsymbol{x}\mu(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\int_{V_i}\mu(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}} \tag{2}$$

and $\mu(\boldsymbol{x})$ is a given density function over domain Ω [16]. When $\mu(\boldsymbol{x})\equiv 1$, $\overline{\boldsymbol{p}}_i$ is the geometric center of V_i. Another description of Centroidal Voronoi tessellations (CVTs) is based on the following 'energy' functional, which is a measure of the deviation of each cell from the corresponding seed:

$$\mathscr{E}(P)=\sum_{i=1}^n\int_{V_i}|\boldsymbol{x}-\boldsymbol{p}_i|^2\mu(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \tag{3}$$

DOI: 10.1002/nme

The minimizer of this functional is necessarily a CVT [16]. The converse, however, is not true. Square and regular hexagonal tessellations are both CVTs on $\Omega = \mathbb{R}^2$, while only the hexagonal tessellation minimizes the energy [16, 17].

One of the most popular algorithms for constructing CVTs is an iterative scheme called Lloyd's algorithm [14]. Since each iteration is guaranteed to reduce the energy, Lloyd's algorithm is locally convergent [18]. The deterministic version of Lloyd's algorithm is as follows:

1. Construct the Voronoi diagram $\mathscr{V}(P)$ associated with $P = \{\boldsymbol{p}_i\}_{i=1}^n$.
2. Compute the centroid of each Voronoi cell $\overline{\boldsymbol{p}}_i$.
3. Replace the original point set P with the set of centroids $\overline{P} = \{\overline{\boldsymbol{p}}_i\}_{i=1}^n$ and go to step 1 unless convergence is reached.

The convergence criterion can be based on the reduction in energy or the movement of seeds in the last iteration. For example, consider the Voronoi diagram defined over a square domain shown in Figure 1. Clearly, the centroids of the Voronoi cells do not coincide with the generating point set (Figure 1(a)). Figure 1(b) shows the Voronoi diagram generated by the centroids of the original cells, i.e. after one iteration of Lloyd's algorithm. After 10 iterations, the Voronoi diagram is nearly centroidal (see Figure 1(c) and (d)).

2.2. *Meshing algorithm*

A polygonal mesh can be generated from a given set of points in the domain Ω by including additional points such that the resulting Voronoi tessellation incorporates an approximation to the boundary $\partial\Omega$. The procedure proposed in [8, 13] is summarized as follows:

1. The interior of Ω is populated with a desired number of generating seeds. We denote this point set by P_{int}.
2. To establish an approximation to the boundary of the domain, the interior points are reflected about the edges of the domain. This set of *auxiliary* points is denoted by P_{aux}.
3. The Voronoi diagram of point set $P = P_{\text{int}} \cup P_{\text{aux}}$ is constructed.
4. A polygonal discretization of the domain is given by the cells associated with the points in P_{int}.

Note that if a seed and its reflection have adjacent Voronoi cells, the common edge between them lines up with the boundary. This procedure is illustrated in Figure 2 for discretizing a four-sided region with 10 points. It is clear that some auxiliary points do not affect the final Voronoi mesh and therefore are unnecessary. By reflecting each point only about the 'closest' boundary, it is possible to avoid including such unnecessary auxiliary points.

The main steps of this meshing procedure, namely seed placement and reflection, can be implemented for general domains based on an implicit description of the geometry. One such approach is to construct a signed distance function, $d(\boldsymbol{x})$, that gives the distance to the closest boundary of the domain. In [19], the construction of distance functions for some simple geometries, as well as operations for combining geometries (e.g. union, intersection, and subtraction) are discussed. One can determine whether a randomly generated seed is inside the domain using the sign of the distance function, which by convention is taken to be negative in the interior of the domain. Moreover, the direction to the nearest boundary (i.e. outward normal at the closest

DOI: 10.1002/nme

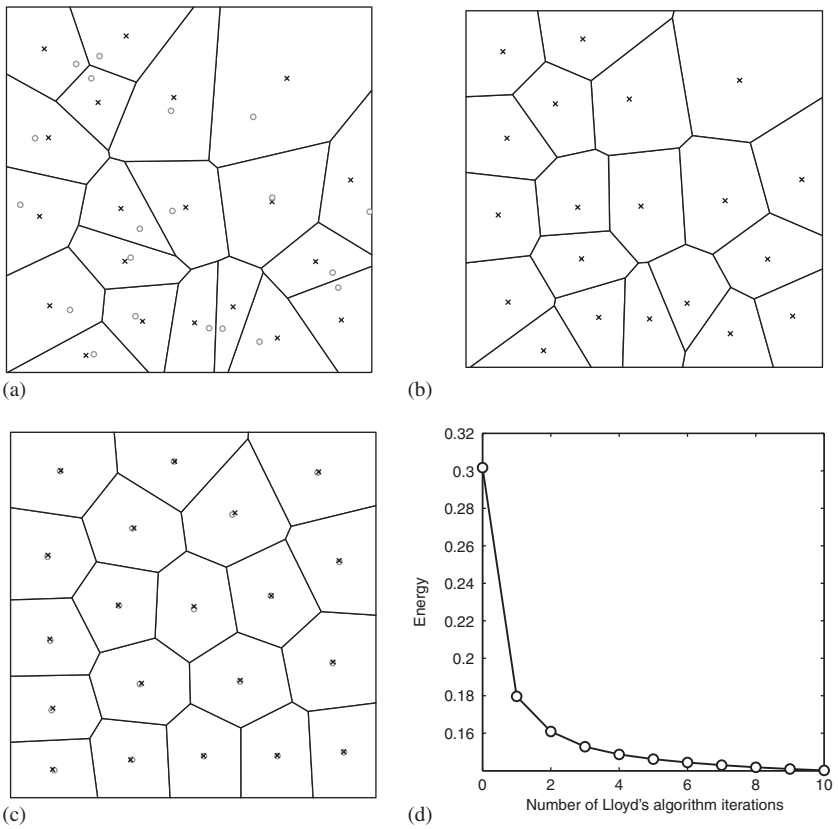


Figure 1. Illustration of Lloyd's algorithm: (a) initial distribution of seeds (circles), the corresponding Voronoi diagram ($\mathscr{E}=0.3017$), and the centroid of the Voronoi cells (crosses); (b) the Voronoi diagram after one iteration. The distribution of points is substantially more regular ($\mathscr{E}=0.1796$); (c) the Voronoi diagram after 10 iterations. The generating seeds (circles) and centroids (crosses) are nearly coincident ($\mathscr{E}=0.1402$); and (d) convergence of Lloyd's algorithm in terms of energy.

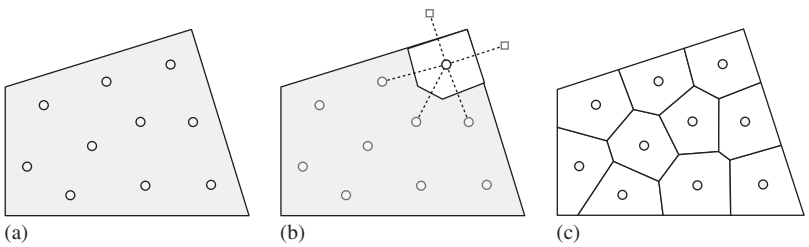


Figure 2. Meshing procedure: (a) interior of the domain is populated with desired number of seeds (circles); (b) the Voronoi diagram of the seeds and the corresponding auxiliary points (squares) are obtained (unnecessary auxiliary points are not shown); and (c) the Voronoi discretization is given by the cells associated with the seeds.

DOI: 10.1002/nme

boundary point) is given by the gradient of the distance function at that point (Figure 3). This means that the reflection of point $\boldsymbol{x} \in \Omega$ can be generically computed as:

$$\overline{\boldsymbol{x}} = \boldsymbol{x} - 2d(\boldsymbol{x}) \frac{\nabla d(\boldsymbol{x})}{|\nabla d(\boldsymbol{x})|} \tag{4}$$

Figure 4 illustrates example (7) of Reference [19], where the distance function is determined from the level set description of the domain boundary. Note that in order to accurately represent the corners (where there is a jump in the outward normal), it is necessary to reflect the nearby seeds about both boundary segments incident on that corner.

2.3. Mesh quality

In References [8, 13], it is recommended that the domain interior be populated in a quasi-random manner by enforcing a minimum allowable distance between the interior points. This minimum separation eliminates the bunching up of the random points, and can also be used to generate graded meshes. Even with such a measure, the method may produce distorted elements not suitable for use in finite element analysis. An attractive alternative is to incorporate Lloyd's algorithm so that the resulting Voronoi mesh is centroidal. The modified procedure is as follows:

1. Perform one iteration of Lloyd's algorithm on the point set P_{int} to obtain the new set of interior points $\overline{P}_{\text{int}}$.
2. Generate the auxiliary point set $\overline{P}_{\text{aux}}$ corresponding to $\overline{P}_{\text{int}}$.
3. Replace P with $\overline{P} = \overline{P}_{\text{int}} \cup \overline{P}_{\text{aux}}$ and go to step 1 until desired level of regularity is reached.

We compare the above-mentioned approaches by considering the discretization of a square domain with a circular hole using random and quasi-random seed placement, and the proposed procedure. Meshes shown in Figures 5–7 have $n = 1000$ elements. In the case of quasi-random discretization, we use a minimum allowable distance of $\sqrt{0.68ab/n}$, as prescribed by Bolander and Saito [8] for a rectangular domain with dimensions a and b. The CVT mesh was constructed with 100 iterations of Lloyd's algorithm. As a measure of mesh quality, the coefficient of variation of edge lengths for each element is plotted in gray-scale. The coefficient of variation for a regular polygon is zero (indicated as white in the figures) since all the edges have the same length. We observe that the CVT mesh is superior to the other two meshes in terms of element quality. The histograms of interior angles (at the triple joints) and element areas are also shown in these figures. These plots indicate that Lloyd's iterations drive the Voronoi mesh toward a regular hexagonal packing. Moreover, we note that the CVT mesh is significantly more uniform in size over the domain.

Before concluding this section, we remark that since the seeds are placed randomly, the element and node numbering of the resulting mesh will be random. This, in turn, may cause the stiffness matrix to exhibit an undesirable sparsity pattern and suffer from a large bandwidth. A remedy to this problem consists of using the heuristic reverse Cuthill–McKee (RCM) algorithm, which is designed to reduce the bandwidth and profile (skyline) of sparse symmetric matrices by systematically reordering the node numbers [20]. For example, a typical mesh with 500 elements and 1002 nodes has the sparsity pattern shown in Figure 8. After applying RCM, the bandwidth is reduced from 966 (almost full) to 75. This leads to substantial savings in computational time needed for solving the linear systems with direct solvers. Other reordering algorithms (e.g. [21, 22]) and other solvers, including direct sparse solvers (see, for example, [23]) and iterative solvers tailored for topology optimization problems [24], can also be explored in this work.

DOI: 10.1002/nme

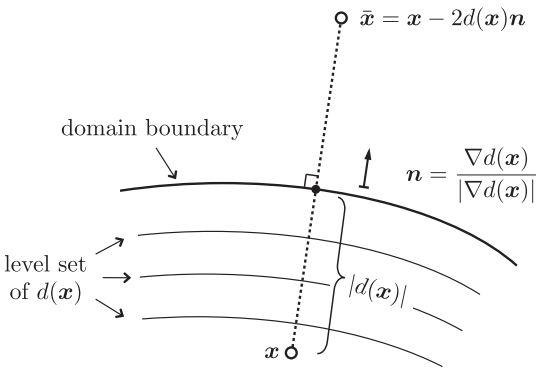


Figure 3. Reflecting points about domain boundary using the distance function $d(\boldsymbol{x})$.

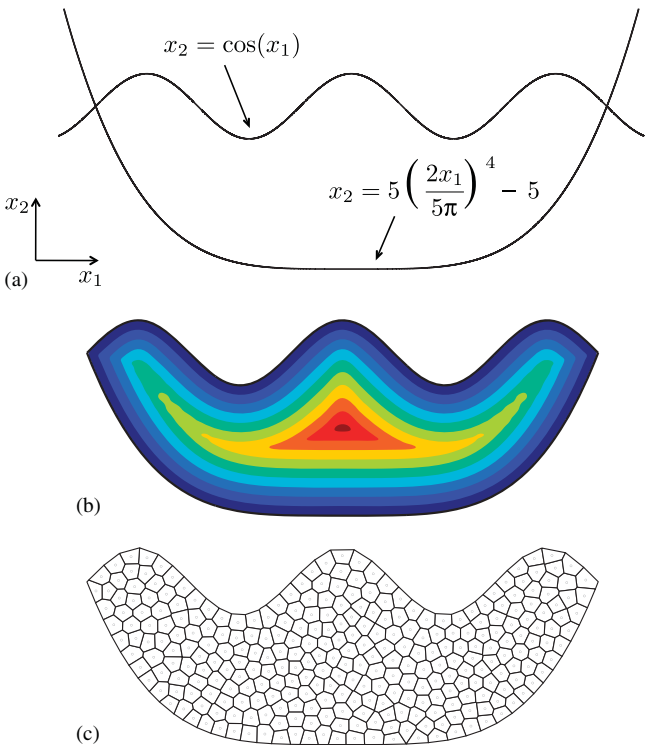


Figure 4. Implicit description of meshing domain: (a) example domain boundaries (taken from [19]); (b) generated distance function; and (c) CVT mesh constructed using the distance function.

3. FINITE ELEMENT SCHEME

In this section, we briefly review the finite element scheme for convex n-gons outlined in [25]. This approach constructs a conforming approximation space on polygonal meshes using natural

DOI: 10.1002/nme

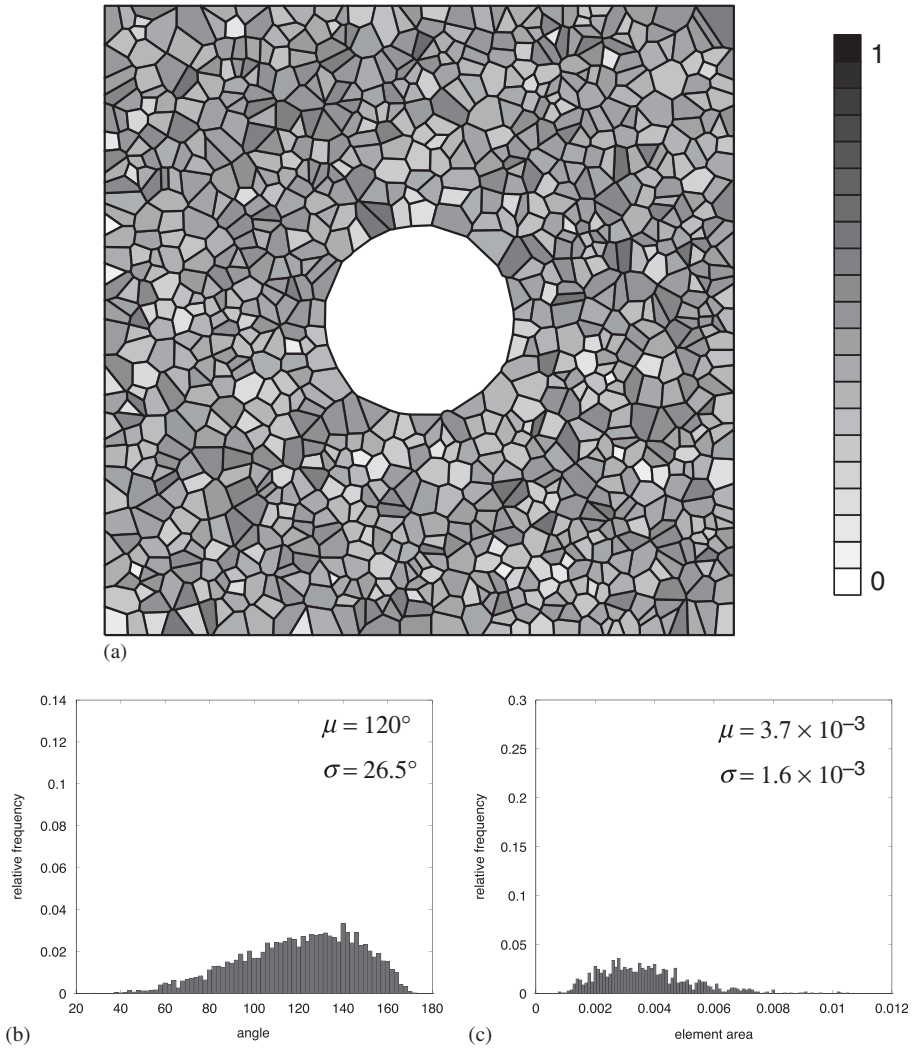


Figure 5. Random seed placement: (a) resulting mesh with coefficient of variation of edge lengths plotted in gray-scale; (b) histogram of interior angles of the mesh; and (c) histogram of element areas.

neighbor interpolation functions and isoparametric transformations. For $n=3$ and $n=4$, the resulting finite element is identical to the constant strain triangle and bilinear quadrilateral, respectively. We assess the accuracy of the polygonal finite elements by considering a verification problem.

3.1. Formulation

Consider a set of nodes $\{\boldsymbol{q}_i\}$ and point $\boldsymbol{p}$ where we would like to interpolate the nodal data. Points $\boldsymbol{p}$ and $\boldsymbol{q}_i$ are 'natural neighbors' if their Voronoi cells have a common edge [25, 26]. We define the

DOI: 10.1002/nme

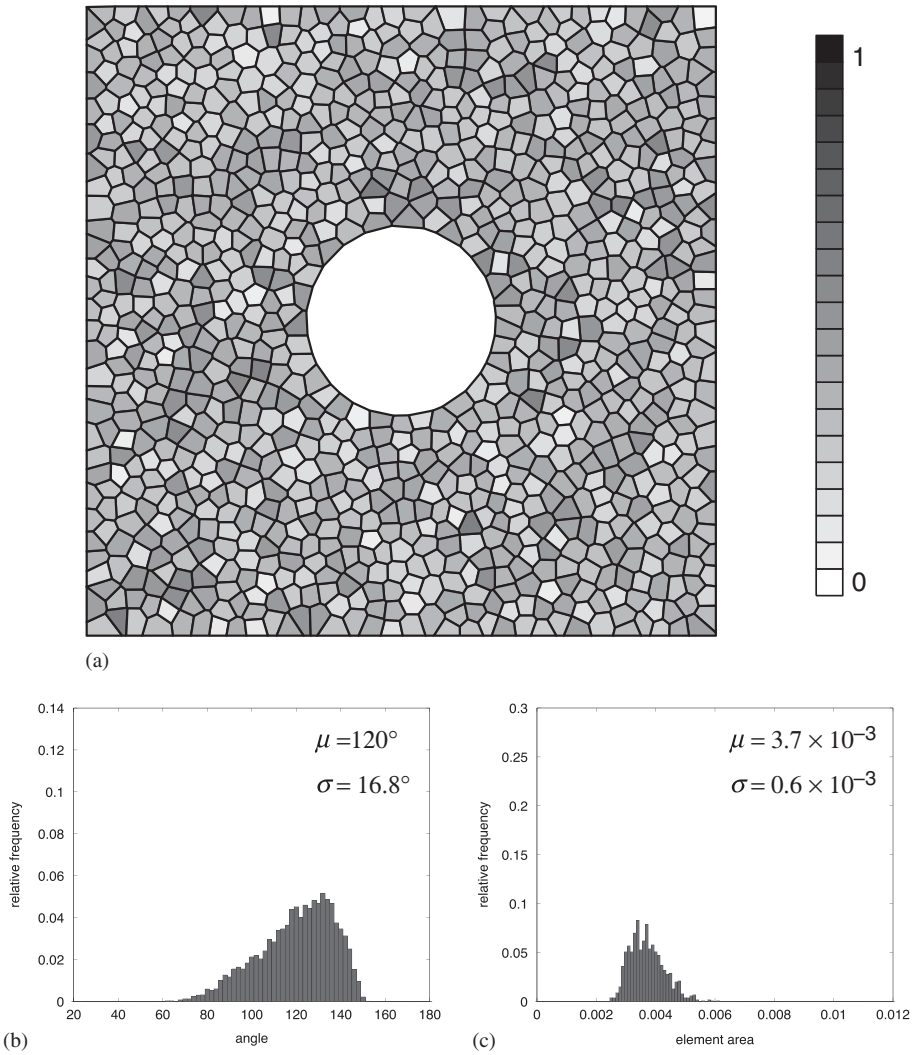


Figure 6. Quasi-random seed placement: (a) resulting mesh with coefficient of variation of edge lengths plotted in gray-scale; (b) histogram of interior angles of the mesh; and (c) histogram of element areas.

set of natural neighbors of $\boldsymbol{p}$ as follows:

$$\mathscr{I}(\boldsymbol{p}) = \{i \,|\, V_i \cap V_{\boldsymbol{p}} \neq \emptyset\} \tag{5}$$

where V_i and $V_{\boldsymbol{p}}$ denote the Voronoi cells of $\boldsymbol{q}_i$ and $\boldsymbol{p}$, respectively. The *Laplace interpolant* corresponding to $\boldsymbol{q}_i$ is given by:

$$\phi_i(\boldsymbol{x}) = \frac{w_i(\boldsymbol{x})}{\sum_{j \in \mathscr{I}} w_j(\boldsymbol{x})} \quad \text{where } w_i(\boldsymbol{x}) = \frac{s_i(\boldsymbol{x})}{h_i(\boldsymbol{x})} \tag{6}$$

DOI: 10.1002/nme

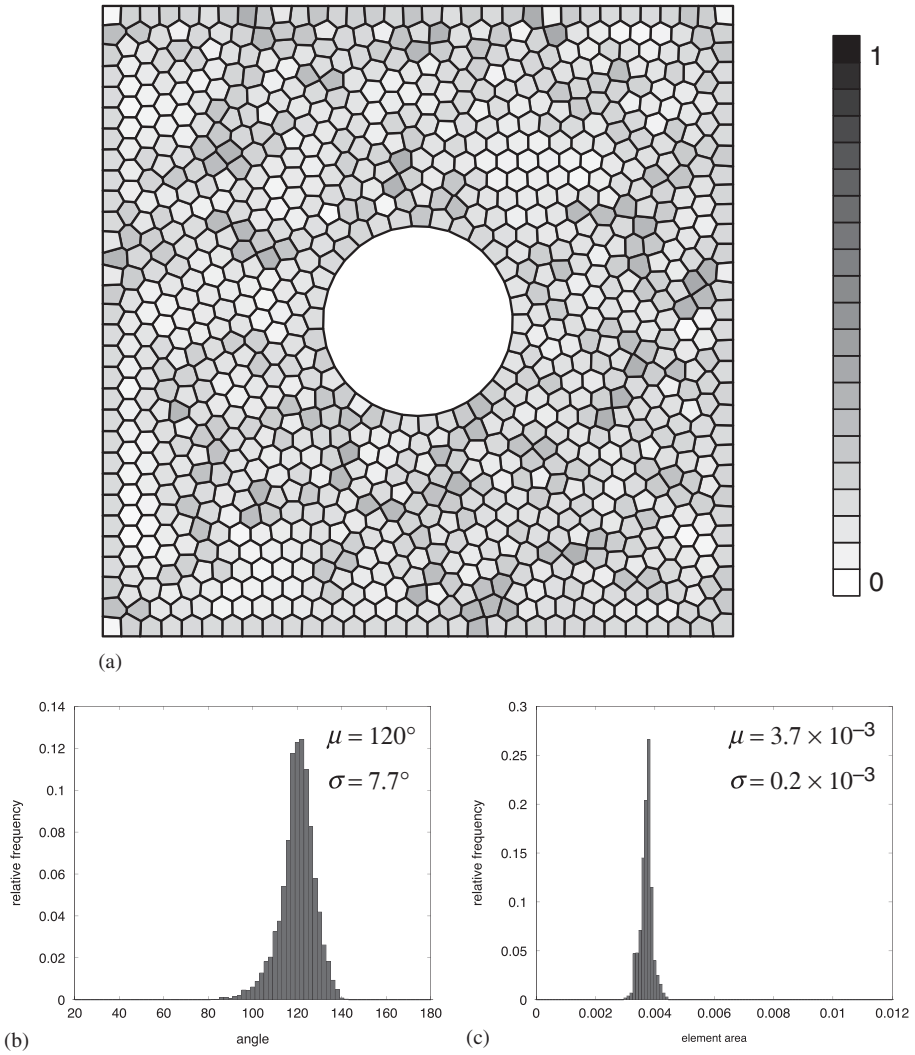


Figure 7. CVT mesh generation: (a) mesh with coefficient of variation of edge lengths plotted in gray-scale; (b) histogram of interior angles of the mesh; and (c) histogram of element areas.

Here $\boldsymbol{x}$ is the location of $\boldsymbol{p}$, s_i is the length of the Voronoi edge common to V_i and $V_{\boldsymbol{p}}$, and h_i denotes the distance between $\boldsymbol{p}$ and $\boldsymbol{q}_i$, as illustrated in Figure 9. By construction, the Laplace functions are non-negative, bounded and satisfy partition of unity:

$$0 \leqslant \phi_i(\boldsymbol{x}) \leqslant 1, \quad \sum_{i \in \mathscr{I}} \phi_i(\boldsymbol{x}) = 1 \tag{7}$$

DOI: 10.1002/nme

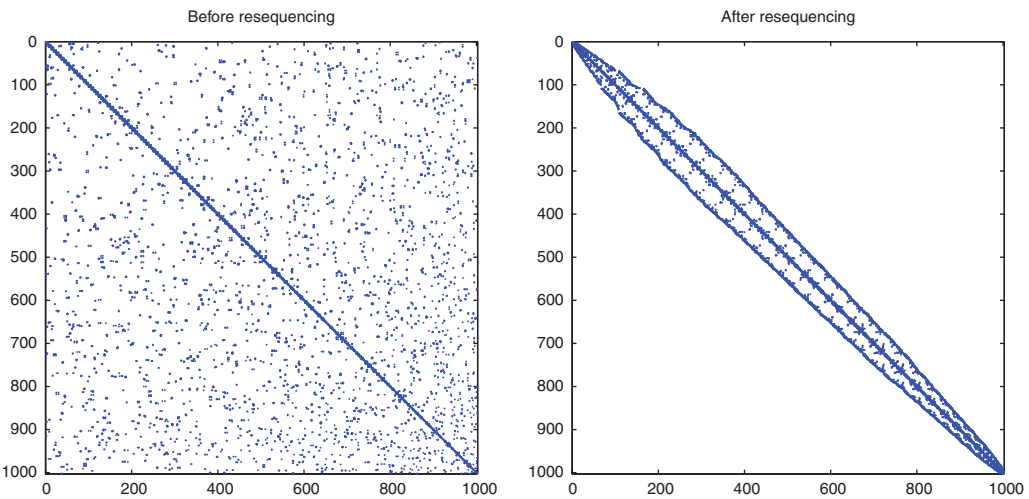


Figure 8. Sparsity pattern for the stiffness of polygonal mesh with 500 elements and 1002 nodes before RCM resequencing (left) and after resequencing (right). The bandwidth is reduced from 966 to 75.

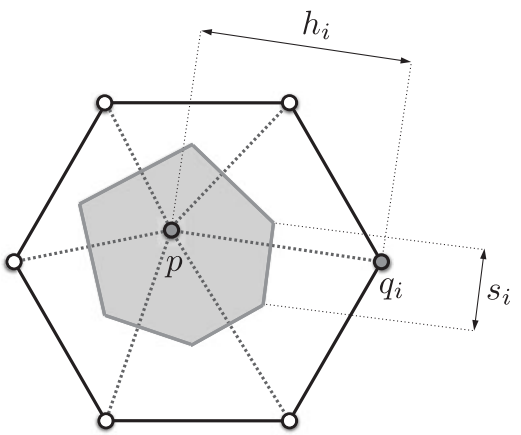


Figure 9. For a convex polygon, every interior point is a natural neighbor to all the vertices. The geometric quantities s_i and h_i used to define the Laplace shape functions are shown here.

Furthermore, it can be shown that these functions are linearly precise:

$$\sum_{i \in \mathscr{I}} \boldsymbol{x}_i \phi_i(\boldsymbol{x}) = \boldsymbol{x} \tag{8}$$

In this expression, $\boldsymbol{x}_i$ represents the location of node $\boldsymbol{q}_i$. This property along with constant precision (partition of unity) ensures the convergence of the Galerkin method for second-order partial

DOI: 10.1002/nme

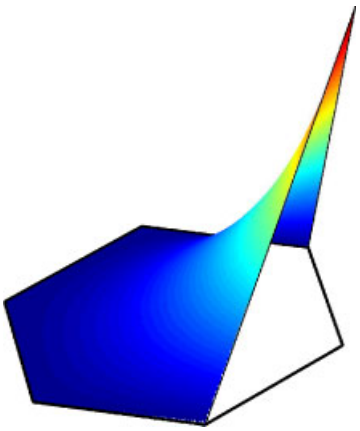


Figure 10. Typical Laplace shape function for a regular hexagon.

differential equations [27]. Moreover, Laplace functions are linear on the boundary of the convex hull of $\{\boldsymbol{q}_i | i \in \mathscr{I}\}$ [28], and satisfy the Kronecker-delta property:

$$\phi_i(\boldsymbol{x}_j) = \delta_{ij} \tag{9}$$

This means that the interpolated value at a node is equal to the nodal value. A typical Laplace shape function is shown in Figure 10.

If the nodes are located at the vertices of a convex polygon, any interior point of this polygon has $\{\boldsymbol{q}_i\}$ as its natural neighbors. Therefore, Laplace shape functions corresponding to $\{\boldsymbol{q}_i\}$ constitute a finite element for that polygon. Furthermore, an isoparametric mapping from regular n-gons (the so-called 'parent' element) to any convex polygon can be constructed using these shape functions (Figure 11). Since the interpolated field varies linearly on the boundary, the resulting approximation is conforming. Following the usual approach in the finite element community, the shape functions are defined on the parent domain,[‡] where the weak form integrals are evaluated numerically. The reference n-gon is divided into n triangles (by connecting the centroid to the vertices) and well-known quadrature rules are used on each triangle. For the verification problem in the next section, we have used three integration points per triangle. Alternatively, numerical integration can be carried out using special quadrature rules developed for polygonal domains (see, for example, [30–32]).

3.2. Verification

The numerical verification example chosen here is Cook's problem [33, 34], which consists of a tapered swept panel subjected to uniform shear loading (Figure 12). The quantity of interest is the tip deflection at mid-depth (point C), which is computed on several quadrilateral and polygonal discretizations, some of which are shown in Figure 12. Note that the refinement for quadrilateral

[‡]The closed-form expressions for these shape functions can be found in the appendix of Reference [29].

DOI: 10.1002/nme

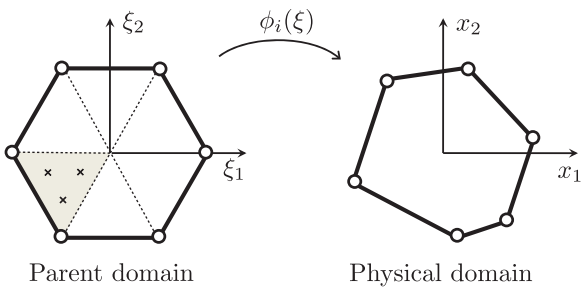


Figure 11. Isoparametric mapping from the parent (reference) domain (regular n-gon) to the physical domain. The weak form integration is carried out in the parent domain by triangulating the n-gon and using triangular quadrature rules.

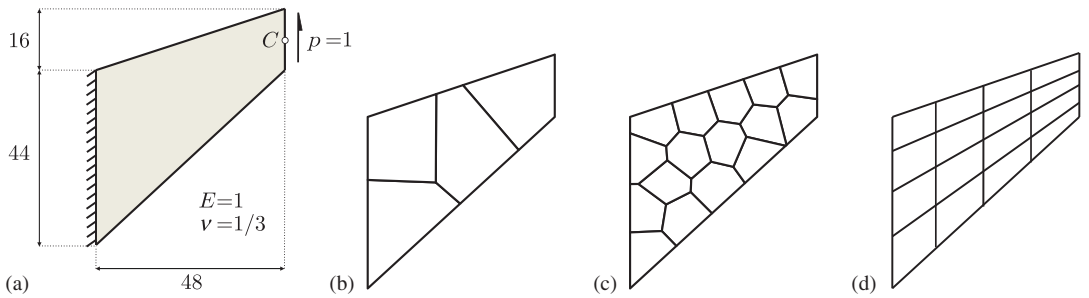


Figure 12. (a) The geometry, boundary conditions, and material properties for Cook's problem; (b) polygonal mesh with 4 elements; (c) polygonal mesh with 16 elements; and (d) typical quadrilateral mesh with 16 elements (each edge is divided evenly).

meshes is progressive (i.e. the finer meshes are embedded in the coarser ones), whereas the polygonal meshes are constructed independently. The results are compared with the reference value of 23.96, reported in Reference [34]. The values for deflection and relative error are summarized in Table I and plotted in Figure 13.

As expected, convergence to the exact solution is monotonic for both types of elements. However, polygonal elements are not as stiff as the quadrilateral elements and produce better results, especially with coarser meshes. It is interesting to observe that even though the polygonal mesh with four elements is made of three quadrilaterals and only one pentagon, it gives a significantly more accurate deflection value than the corresponding quadrilateral mesh. In fact, the accuracy obtained on this mesh is comparable to that of the quadrilateral mesh with 16 elements. Since numerical instabilities such as checkerboard patterns are caused by their artificial stiffness in the finite element approximation, polygonal elements are expected to be less susceptible to such pathologies. In fact, it was shown in a recent study that undesirable fine scale patterns (resembling checkerboard) are appropriately penalized in the compliance minimization problem when modeled by hexagonal finite elements [10]. We note that better accuracy and in some cases superconvergence of polygonal and

DOI: 10.1002/nme

Table I. Summary of results for Cook's problem.

	Polygonal			Quadrilateral		
# elements	# nodes	Deflection	% error	# nodes	Deflection	% error
4	10	17.4494	27.17	9	11.8090	50.71
16	34	21.9240	8.497	25	18.2902	23.66
64	130	23.4488	2.134	81	22.0781	7.854
256	514	23.8204	0.583	289	23.4303	2.211
1024	2050	23.9257	0.143	1089	23.8176	0.594
4096	8194	23.9539	0.026	4225	23.9245	0.148

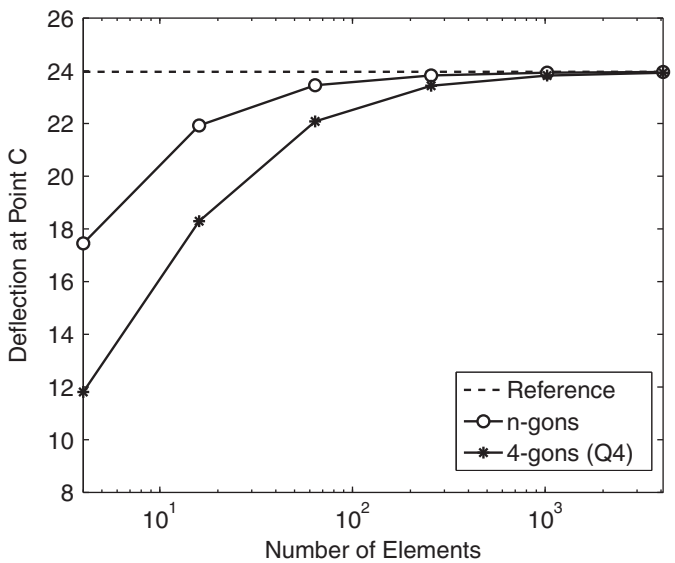


Figure 13. Illustration of the convergence of numerical results for Cook's problem.

triangular finite element discretizations constructed using CVTs have been previously observed in References [35, 36].

4. TOPOLOGY OPTIMIZATION

In this work, we consider the compliance minimization problem, which consists of finding the stiffest (i.e. least compliant) configuration of material in an extended design domain Ω subject to a given set of loading and support conditions. We assume that the constituent material is linear elastic and isotropic with stiffness tensor $\boldsymbol{C}_0$. Furthermore, we follow the solid isotropic material

DOI: 10.1002/nme

with penalization (SIMP) method, in which the design field is characterized by a density function $\rho(\boldsymbol{x})$, and the stiffness tensor follows the power-law relation [37]:

$$\boldsymbol{C}(\boldsymbol{x}) = [\rho(\boldsymbol{x})]^p \boldsymbol{C}_0, \quad 0 < \rho(\boldsymbol{x}) \leqslant 1 \tag{10}$$

To penalize the intermediate densities, the penalty exponent p is taken greater than 1. The minimum compliance problem is given by

$$\inf_{\rho} \ell(\boldsymbol{u}) \quad \text{subject to} \int_{\Omega} \rho(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \leqslant \overline{V} \tag{11}$$

where $\overline{V}$ is a given upper bound for the total volume of structure and $\boldsymbol{u}$ is the displacement field corresponding to the equilibrium configuration:

$$a(\boldsymbol{u}, \boldsymbol{v}) = \ell(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathscr{U} \tag{12}$$

Here $\mathscr{U} = \{\boldsymbol{u} \in H^1(\Omega) | \boldsymbol{u} = 0 \text{ on } \partial\Omega_{\boldsymbol{u}}\}$ is the space of admissible displacement fields and

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}) &= \int_{\Omega} \boldsymbol{C}(\boldsymbol{x}) \boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, \mathrm{d}\boldsymbol{x} \\ \ell(\boldsymbol{v}) &= \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{v}\, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_t} \boldsymbol{v} \cdot \boldsymbol{t}\, \mathrm{d}\boldsymbol{s} \end{aligned} \tag{13}$$

are the energy bilinear form and linear form, respectively. In the above expression, $\boldsymbol{f}$ is the body force, $\boldsymbol{t}$ is the traction applied to $\partial\Omega_{\boldsymbol{t}} = \partial\Omega \backslash \partial\Omega_{\boldsymbol{u}}$ and $\boldsymbol{\varepsilon}$ is the linearized strain given by:

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \tfrac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\mathrm{T}}) \tag{14}$$

To solve this problem numerically, we discretize the displacement field using polygonal finite elements as described in Sections 2 and 3. The discretization of the density is implicitly carried out on the same mesh by assuming a constant element density inside each displacement finite element Ω_e:

$$\rho^h(\boldsymbol{x}) = \sum_{e=1}^{n} \chi_e(\boldsymbol{x}) \rho_e \quad \text{where } \chi_e(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in \Omega_e \\ 0, & \boldsymbol{x} \notin \Omega_e \end{cases} \tag{15}$$

With this discretization, $\{\rho_e\}_{e=1}^n$ is the set of design variables. Alternatively, a continuous variation of density can be obtained by sampling the density at the nodal locations and interpolating it with the shape functions [38, 39]. As mentioned before, the optimization problem (11) does not have a physical length scale. However, the mesh size induces a length scale on ρ^h, which effectively translates the ill-posedness of (11) into the mesh dependency of the discrete system.

We can avoid this issue by imposing an explicit length scale on the problem independent of the mesh size. One possibility is to use the projection method [40, 41] which introduces a minimum

DOI: 10.1002/nme

member size, $r_{\min}$, into the discrete problem by assigning the weighted average of the nearby nodal densities to each element. Thus, the projected element density becomes:

$$\rho_e = \frac{\sum_{i=1}^{m} w_e^i \rho^i}{\sum_{i=1}^{m} w_e^i} \tag{16}$$

Here $\{\rho^i\}_{i=1}^m$ is the set of nodal densities (taken as the design variable for the optimization problem) and w_e^i is the (linear) weight function defined as:

$$w_e^i = \max\left(\frac{r_{\min} - r_e^i}{r_{\min}}, 0\right) \tag{17}$$

As shown in Figure 14, r_e^i is the distance from the centroid of element e to node i. We remark that this projection is effectively a convolution of the density function with the weight functions. In the continuum setting, this amounts to regularizing the density field, which inherits the smoothness properties of the weight functions. According to the analysis by Bourdin [42], this convolution removes the ill-posedness of the continuum problem and mesh dependency of the discrete problem.

5. IMPLEMENTATION ISSUES

In this work, the polygonal finite element meshes are represented by a compact topological data structure called TopS [43–45]. TopS is designed to provide direct and efficient access to local topological information about the mesh, and it does so by relying on static element topology templates. A template consists of a table with all the topological relationships within an element and is defined for each element type. Among the information stored in the template are the number of nodes, vertex-, edge- and facet-uses of the element; the number and local indices of the nodes incident to a given edge- or facet-use; and the number and local indices of the edge- and facet-uses incident to a given vertex-use.

For the most element types (e.g. $T3$, $T6$, $Q4$, $Q8$, Tetra4, Tetra10), topological information is common to all the elements of the same type and thus can be efficiently represented by a static template. However, polygonal elements contain a variable number of nodes, vertices, and edges. In order to represent these elements in TopS, we have extended the topological framework with support for dynamic templates. In this manner, if topological information is requested from a polygonal element, a virtual method of the element, rather than a static template, is invoked in order to provide the required data. Moreover, nodal and adjacency data of each polygonal element are allocated dynamically. Although this approach introduces additional costs for polygonal elements, it has a very simple implementation and allows one to represent arbitrary elements in a seamless way using the existing template-based topological framework of TopS.

The optimization problem described in the previous section is solved using the globally convergent method of moving asymptotes (GCMMA) [46, 47]. The algorithm requires the sensitivities of the objective and constraint functions, which are straightforward to compute for the minimum compliance problem (this can be found in several references, for example, see [48]). To avoid converging to local minima, we perform a continuation on the penalty parameter p by gradually increasing its value from $p=1$ to $p=5$. For each value of p, we terminate the iterations when the maximum change in the design variables is less than a given tolerance. Unless otherwise stated,

DOI: 10.1002/nme

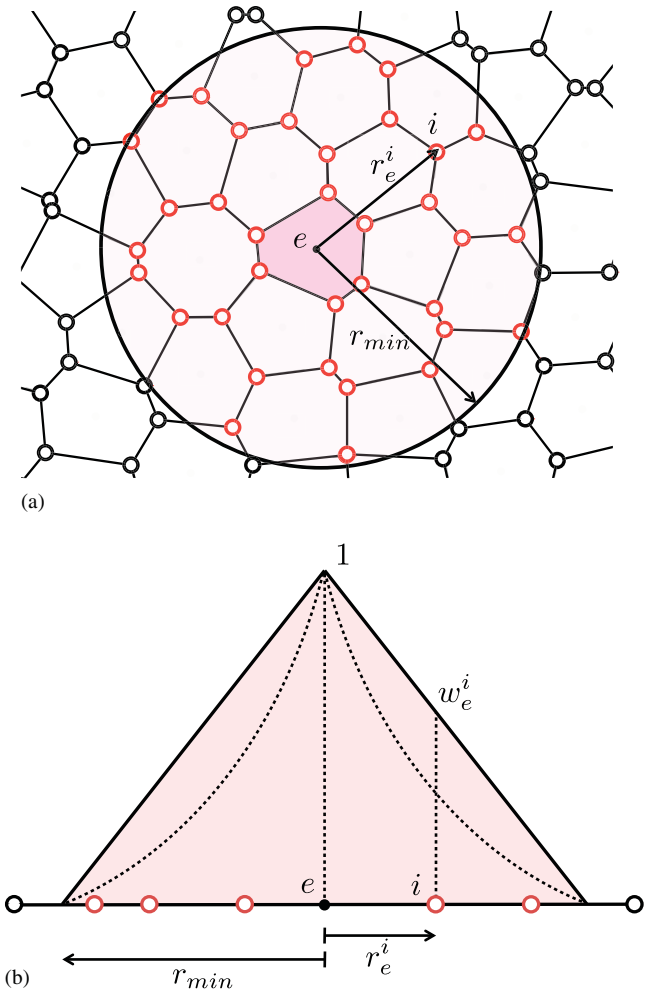


Figure 14. Illustration of the projection scheme: (a) radius of projection $r_{\min}$ and nodes that contribute to density of element e and (b) the solid line indicates the linear weight function and the dashed line indicates a non-linear weight function.

increments of $\Delta p = 0.5$, tolerance of 0.01, minimum of 15 iterations, and maximum of 75 iterations per value of p were used for the numerical results in Section 6.

6. NUMERICAL RESULTS

In this section, we assess the performance of the polygonal discretization by considering various numerical examples. We begin with a geometric study in which we compare polygonal meshes

DOI: 10.1002/nme

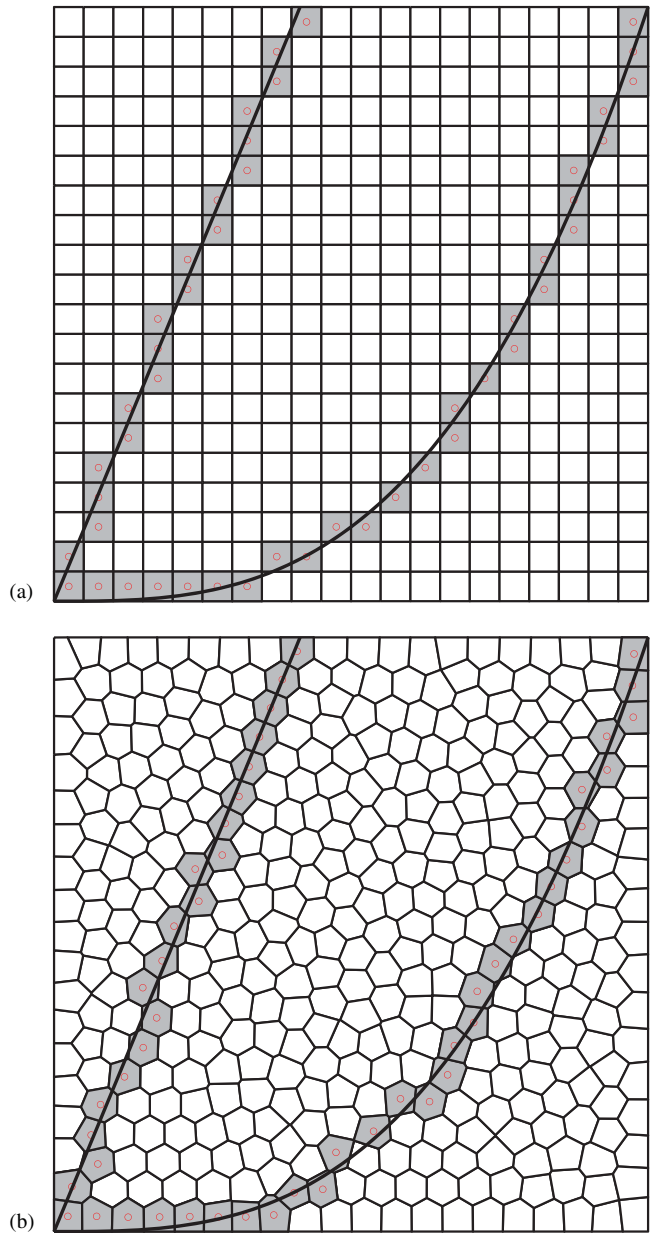


Figure 15. Geometric study: approximation of curve $y=(\tan 3\pi/4)x$ and $y=x^3$ using (a) regular square mesh and (b) CVT mesh.

DOI: 10.1002/nme

Table II. Summary of results for the geometric study.

	Straight line		Cubic line	
Type	CVT mesh	Square mesh	CVT mesh	Square mesh
Average vertex distance	2.18×10^{-3}	2.41×10^{-3}	2.29×10^{-3}	2.43×10^{-3}
Length of approx. path	1.151	1.178	1.649	1.649

and structured square grids in their ability to represent arbitrary curves. We then present topology optimization results for benchmark Michell-type problems.

6.1. Geometric study

As discussed in the introduction, the geometric properties of the spatial discretizations affect the quality of topology optimization results one could obtain. With higher levels of geometric isotropy, arbitrary curves (i.e. structural members) can be better approximated. Subsequently there would be less restriction on the formation of the optimal design. In this study, we consider two simple curves: a straight line with slope of 67.5° with respect to the horizontal axis and a cubic curve $y=x^3$ (Figure 15); and we find approximations to these curves on a square grid and a CVT mesh. We define the 'closest' approximation as the sequence of elements with the least deviation from the given curve. The deviation is computed as the sum of the distances of the element centroids to the curve. The problem of finding the closest approximation can be formulated as a shortest path problem (in a graph) if we view the element centroids as the graph nodes, and the element adjacencies as the graph connectivity. In this setting, the cost associated with going from one graph node to the next is the distance of the centroid of the target element to the curve. The shortest path problem is solved using Dijkstra's algorithm [49], and results are shown in Figure 15.

Qualitatively, the approximations on the CVT mesh appear to follow the curve more closely. The approximate curves on the square mesh suffer, as expected, from one-node connections, while CVT approximations enjoy greater uniformity. To quantify this, we have computed the average distance of the element vertices to the curve. As shown in Table II, the approximate path on the CVT mesh, on an average, is closer to the given curve. We also computed the length of the approximate path as the sum of lengths of line segments connecting the centroids in the path. In both cases, this length is larger than that of the true path, but the error is slightly larger for the square mesh, indicating more zig-zagging around the curves.

6.2. Benchmark examples

For the examples in this section, we used the Poisson's ratio of $\nu=0.3$ and Young's modulus of $E_0=1$ for the solid phase. The first example is the MBB-beam problem [50], whose design domain and boundary conditions are shown in Figure 16. To examine the effects of mesh regularity, we compared the results obtained on random, quasi-random, and CVT discretizations, each made up of 2000 elements (see Section 2). The final designs with compliance values are shown in Figure 17. The results generally agree with the analytic least-weight Michell truss (Figure 16). The CVT design has a smoother flow of members and also a slightly lower compliance. We point out that a direct comparison of compliance values may not be appropriate since they are computed on

DOI: 10.1002/nme

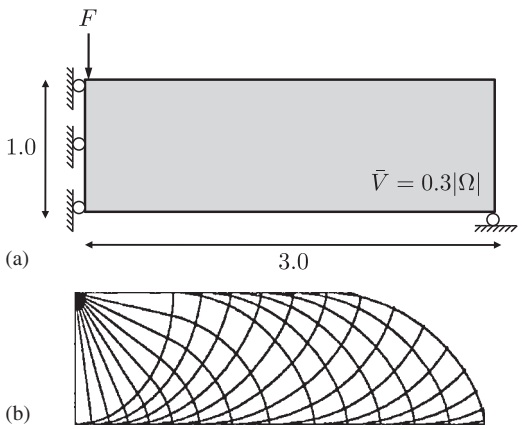


Figure 16. MBB beam problem: (a) the extended design domain and loading and support conditions and (b) the corresponding Michell solution taken from Reference [51].

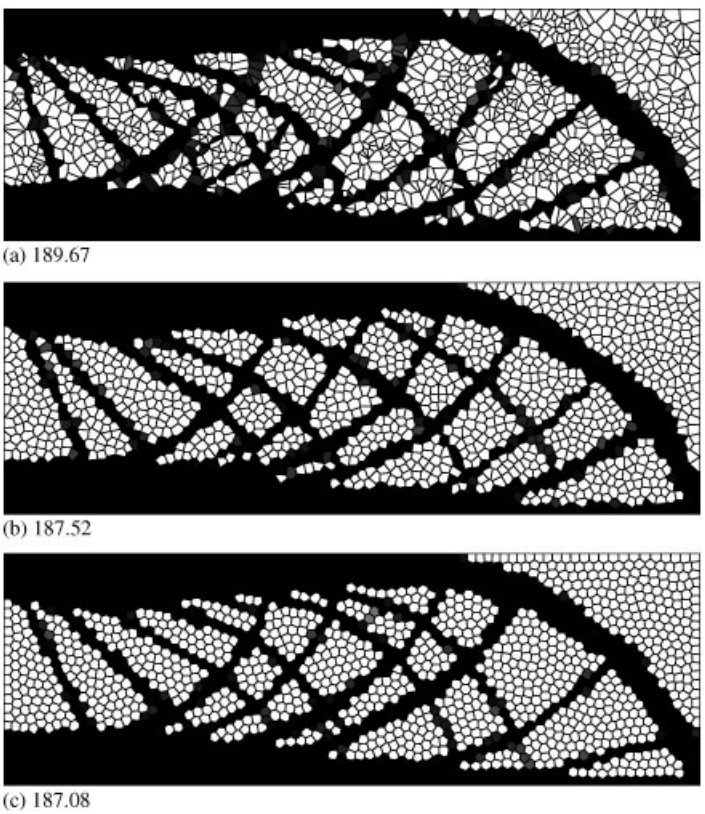


Figure 17. Effects of mesh regularity: (a) random; (b) quasi-random; and (c) CVT meshes. The final compliance value is shown below each result.

DOI: 10.1002/nme

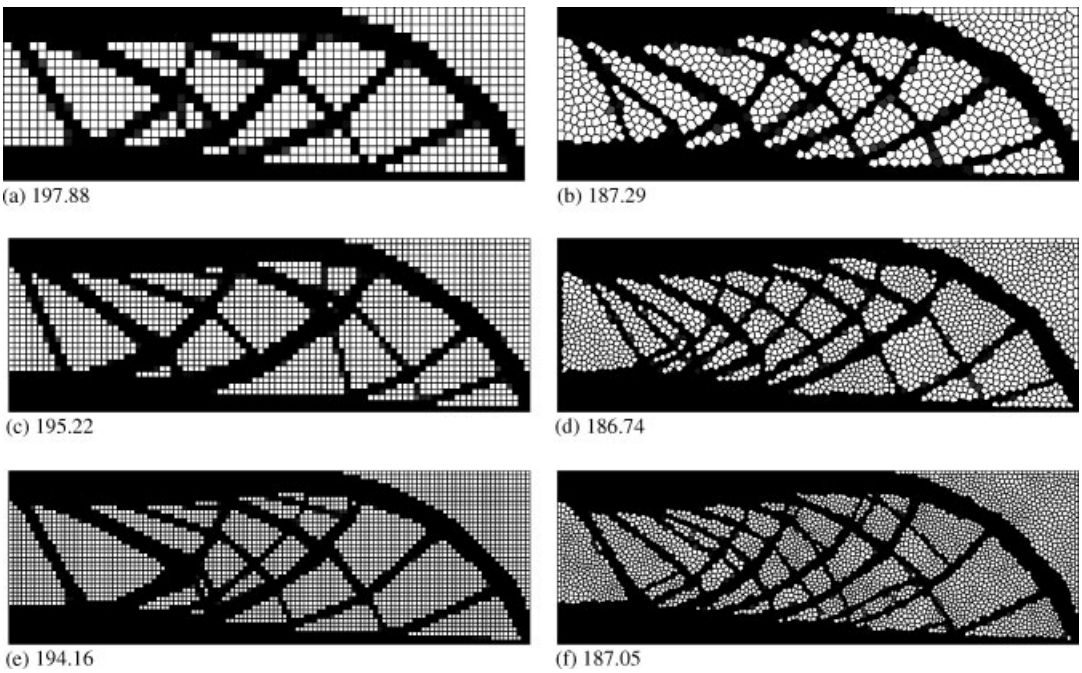


Figure 18. MBB beam problem: left-hand-side shows $Q8$ meshes (a)(c)(e) and right-hand-side shows CVT meshes; (b)(d)(f) with 1200, 2700, and 4800 elements (from top to bottom). The final compliance value is shown below each result.

different finite element meshes. It is also noteworthy that the design on the 'random' mesh follows the expected layout despite the poor quality of the finite elements.

Next we compare the results obtained on standard structured square grids with those obtained with CVT meshes (Figure 18). To alleviate the checkerboard problem, we use $Q8$ elements for the rectangular mesh. We point out that this choice increases the size of problem considerably. For example, the $Q8$ mesh with 4800 elements has 29 360 unknown degrees of freedom, whereas the CVT mesh with the same number of elements has 19 163 unknowns. As shown in Table III, the overall cost of polygonal elements is lower even though shape function computations are more expensive. Given that the same number of elements is used for both meshes, CVT results seem to have higher resolution: there are more (but finer) members in the CVT results, indicating a more even distribution of material. Also, we can see instances of one-node connections and diagonal element chains (see [9] for a detailed discussion on these discretization errors) in the $Q8$ results.

Both sets of results in Figure 18 exhibit the well-known mesh dependency associated with the refinement of discretization. By applying the projection scheme, one can enforce a minimum member size independent of the grid scale. The results for $r_{\min} = 0.12$ are shown in Figure 19. The designs with the projection method appear to be convergent under mesh refinement. However, the final compliance is increased significantly compared with the results without the minimum member size.

Finally, we consider the Michell cantilever problem with circular support, shown in Figure 20. For this problem, we compare the CVT meshes with triangular discretizations, which are widely used

DOI: 10.1002/nme

Table III. Summary of the computational cost associated with finite element analysis.

Element (type of integration)	$Q8$ (reduced)	$Q8$ (full)	Polygonal (1 pt/triangle)	Polygonal (3 pts/triangle)
# elements	4800	4800	4800	4800
# integration points	19 200	43 200	28 501	85 503
# unknowns	29 360	29 360	19 163	19 163
Time for assembling stiffness matrix (s)	0.1049	0.1657	0.1605	0.3797
Time for solving the linear system (s)	1.1847	1.1883	0.7398	0.7372
Total time for FE analysis (s)	1.2896	1.3540	0.9003	1.1169

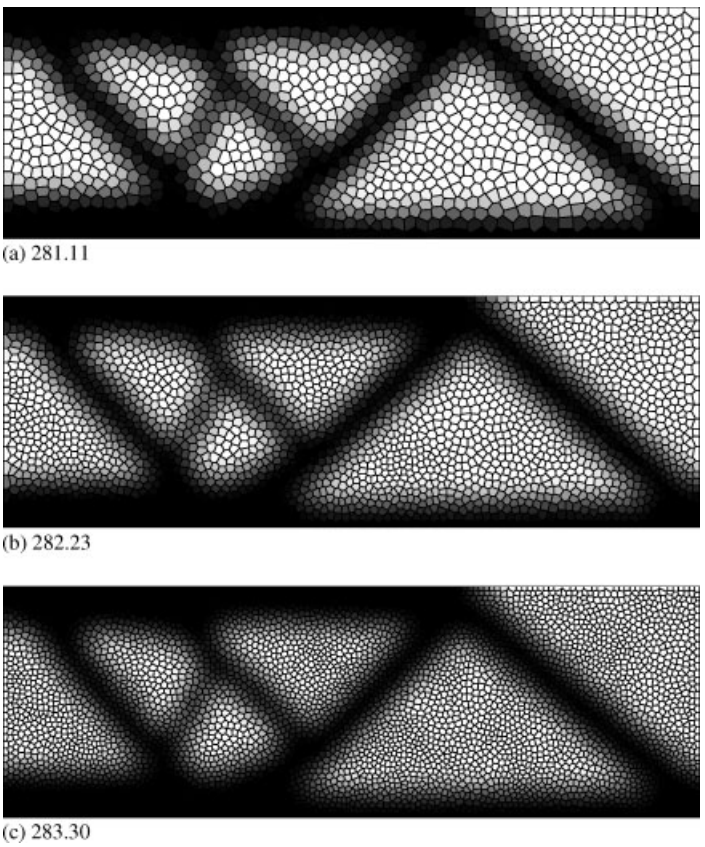
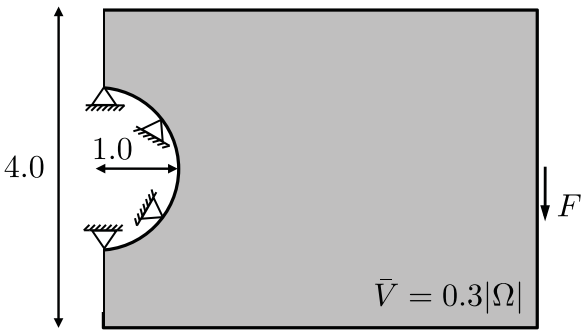


Figure 19. MBB solution obtained with CVT meshes using the projection scheme with $r_{\min}=0.12$: (a) 1200 n-gons; (b) 2700 n-gons; and (c) 4800 n-gons. The final compliance value is shown below each result.

for meshing in finite element community. We use quadratic $T6$ elements to avoid checkerboard-type instabilities. In both cases, the meshes are constructed to be symmetric about the horizontal axis at mid-depth, and solutions are forced to be symmetric via a simple mapping scheme. Also a tighter convergence criterion consisting of $\Delta p=0.25$, tolerance of 0.001 and maximum of 100

DOI: 10.1002/nme



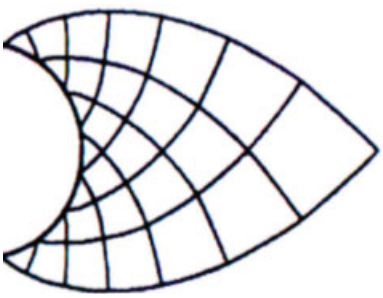


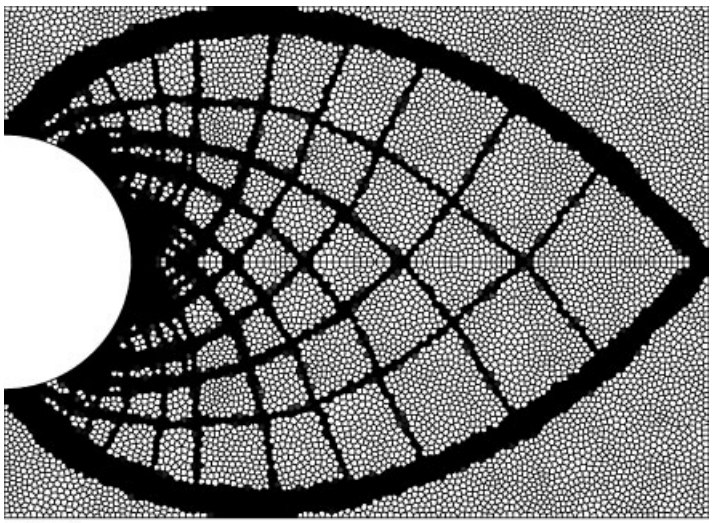
Figure 20. Cantilever beam with circular support: (a) the extended design domain and loading and support conditions and (b) a typical Michell solution taken from Reference [52].

iterations per value of p is used for the continuation. Similar to the MBB results, the CVT design has more members and thus exhibits higher resolution. More importantly, the general layout of members is in better agreement with the Michell solution. The members intersect at roughly 90° angles and are spaced more evenly. Also, the principal stresses for the optimal design are aligned with the members according to the Michell layout theory. This fact is illustrated for a smaller CVT mesh for better visualization in Figure 22. These results indicate that the CVT meshes have the flexibility to represent the optimal layout for this problem. The $T6$ mesh, on the other hand, suffers from the limitation of its geometry. Members that line up with the mesh must strictly conform to it, while others are poorly approximated (see the members marked on the Figure 21).

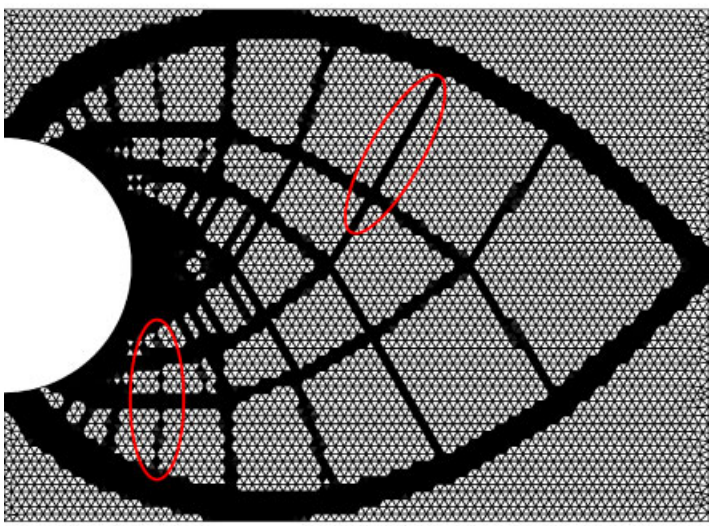
7. CONCLUSIONS

Solutions of discrete topology optimization problems with fixed mesh representation include a form of mesh dependency that stems from the basic geometric features of the spatial discretization associated with standard triangles/tetrahedra or quads/bricks usually employed in the finite element method. The constrained geometry of such discretization can cause bias in the orientation of members (cf. Figure 21(b)), leading to non-objective, mesh-dependent, sub-optimal designs. To circumvent this problem, we employ fully unstructured meshes to reduce the influence of the

DOI: 10.1002/nme



(a) 62.56



(b) 62.15

Figure 21. Solutions for the cantilever beam with circular support based on (a) 10 000 Voronoi elements and (b) 10 220 $T6$ elements. The final compliance value is shown below each result.

simplex geometry on topology optimization solutions. This is accomplished by means of polygonal meshes based on Voronoi tessellations, which in addition to possessing higher degree of geometric isotropy allow for greater flexibility in discretization without introducing numerical instabilities/pathologies.

Most two-dimensional investigations in topology optimization are based on triangular elements of three nodes ($T3$) and bilinear quadrilateral ($Q4$) [48]. For $n=3$ and $n=4$ the resulting polygonal

DOI: 10.1002/nme

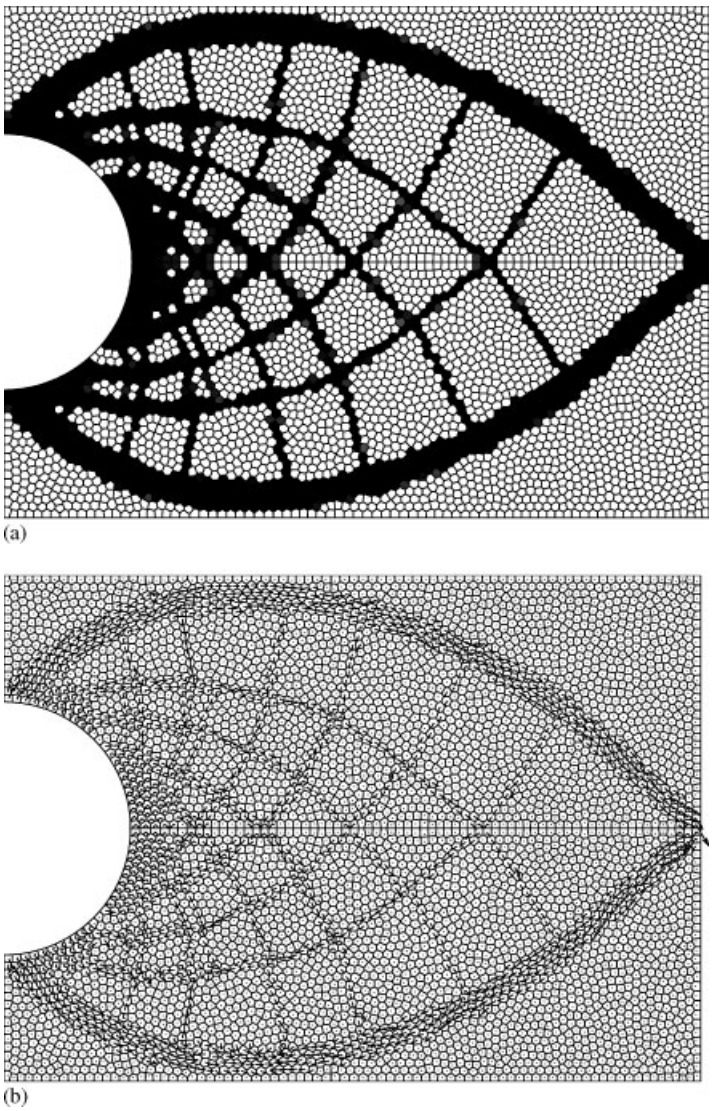


Figure 22. (a) Solution for the cantilever beam with circular support using CVT mesh with 5000 elements and (b) the trajectory of the principal stresses for the optimal design.

element coincides with the $T3$ and $Q4$ finite elements, respectively. Thus, the present polygonal finite element framework ($n = 3, 4, 5, 6, 7, \ldots$) provides a unifying paradigm for topology optimization, including developments beyond the scope of the present work. For instance, further investigations associated with compliant mechanism design, solution of multiscale problems involving multiple governing equations (e.g. thermo-electro-mechanical) and fluid–structure, can benefit from the present approach.

DOI: 10.1002/nme

NOMENCLATURE

$\boldsymbol{p}_i$	generating seed for the Voronoi diagram
V_i	Voronoi cell corresponding to seed $\boldsymbol{p}_i$
$\mathscr{V}(P)$	Voronoi diagram generated from the set of seeds P
$\delta(\cdot,\cdot)$	Euclidean distance in $\mathbb{R}^d$
$\mu(\boldsymbol{x})$	density function defining the centroid of Voronoi cells
$\overline{\boldsymbol{p}}_i$	centroid of cell V_i
$\mathscr{E}(P)$	energy functional for diagram $\mathscr{V}(P)$
P_{int}	set of interior seeds for the mesh generation procedure
P_{aux}	set of auxiliary points for the mesh generation procedure
$d(\boldsymbol{x})$	distance function at point $\boldsymbol{x}$
$\mathscr{I}(\boldsymbol{p})$	index set for the natural neighbors of point $\boldsymbol{p}$
$\phi_i(\boldsymbol{x})$	Laplace shape function
$w_i(\boldsymbol{x})$	weight function used to define the Laplace interpolants
$s_i(\boldsymbol{x})$	length of Voronoi edge common to V_i and $V_{\boldsymbol{p}}$
$h_i(\boldsymbol{x})$	distance between $\boldsymbol{p}$ and node $\boldsymbol{q}_i$
$\boldsymbol{C}_0$	stiffness tensor for the base material
$\boldsymbol{C}(\boldsymbol{x})$	stiffness tensor following the SIMP law
$\rho(\boldsymbol{x})$	material density characterizing the topology
$\overline{V}$	upper bound for the volume of design
Ω	extended design domain
$\mathscr{U}$	space of admissible displacement fields
$a(\cdot,\cdot)$	energy bilinear form
$\ell(\cdot)$	load linear form
$\boldsymbol{\varepsilon}(\cdot)$	linearized strain operator
$\boldsymbol{f}$	body force defined over Ω
$\boldsymbol{t}$	traction applied to $\partial\Omega_t$
$\boldsymbol{u}$	admissible displacement field satisfying equilibrium
$\rho^h(\boldsymbol{x})$	discretization of density field
$\chi_e(\boldsymbol{x})$	characteristic function associated with finite element Ω_e
$\{\rho_e\}_{e=1}^n$	set of element densities
$\{\rho^i\}_{i=1}^m$	set of nodal densities
w_e^i	weight function associated with element e and node i
r_e^i	distance from centroid of element e to node i
$r_{\min}$	prescribed radius of projection

ACKNOWLEDGEMENTS

The first two authors acknowledge the support by the Department of Energy Computational Science Graduate Fellowship Program of the Office of Science and National Nuclear Security Administration in the Department of Energy under contract DE-FG02-97ER25308. The last two authors acknowlege the financial support by Tecgraf (Group of Technology in Computer Graphics), PUC-Rio, Rio de Janeiro, Brazil. The authors are grateful to Prof. Krister Svanberg for providing his GCMMA code, which was used to generate the examples in this paper. Also, the authors thank Prof. John Bolander for providing his Voronoi meshing code; and Rodrigo Espinha and Prof. Waldemar Celes for their help with the implementation of polygonal finite elements in the TopS code.

DOI: 10.1002/nme

REFERENCES

1. Sigmund O, Petersson J. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization* 1998; **16**(1):68–75.
2. Ambrosio L, Buttazzo G. An optimal design problem with perimeter penalization. *Calculus of Variations and Partial Differential Equations* 1993; **1**(1):55–69.
3. Haber RB, Jog CS, Bendsoe MP. A new approach to variable-topology shape design using a constraint on perimeter. *Structural Optimization* 1996; **11**(1):1–12.
4. Petersson J. Some convergence results in perimeter-controlled topology optimization. *Computer Methods in Applied Mechanics and Engineering* 1999; **171**(1–2):123–140.
5. Pedersen CBW, Buhl T, Sigmund O. Topology synthesis of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering* 2001; **50**(12):2683–2705.
6. Park K, Paulino GH, Celes W, Espinha R. Adaptive dynamic cohesive fracture simulation using edge-swap and nodal pertubation operators. *International Journal for Numerical Methods in Engineering*, submitted.
7. Papoulia KD, Vavasis SA, Ganguly P. Spatial convergence of crack nucleation using a cohesive finite-element model on a pinwheel-based mesh. *International Journal for Numerical Methods in Engineering* 2006; **67**(1):1–16.
8. Bolander JE, Saito S. Fracture analyses using spring networks with random geometry. *Engineering Fracture Mechanics* 1998; **61**:569–591.
9. Rozvany GIN, Querin OM, Gaspar Z, Pomezanski V. Weight-increasing effect of topology simplification. *Structural and Multidisciplinary Optimization* 2003; **25**(5–6):459–465.
10. Talischi C, Paulino GH, Le CH. Honeycomb Wachspress finite elements for structural topology optimization. *Structural and Multidisciplinary Optimization* 2009; **37**(6):569–583.
11. Saxena A. A material-mask overlay strategy for continuum topology optimization of compliant mechanisms using honeycomb discretization. *Journal of Mechanical Design* 2008; **130**(8):082304.
12. Aurenhammer F. Voronoi diagrams—a survey of a fundamental geometric data structure. *Computing Surveys* 1991; **23**(3):345–405.
13. Yip M, Mohle J, Bolander JE. Automated modeling of three-dimensional structural components using irregular lattices. *Computer Aided Civil and Infrastructure Engineering* 2005; **20**(6):393–407.
14. Lloyd S. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 1982; **28**(2):129–137.
15. Preparata FP, Shamos MI. *Computational Geometry—An Introduction*. Springer: New York, 1985.
16. Du Q, Faber V, Gunzburger M. Centroidal Voronoi tessellations: applications and algorithms. *SIAM Review* 1999; **41**(4):637–676.
17. Newman D. The hexagon theorem. *IEEE Transactions on Information Theory* 1982; **28**:137–139.
18. Du Q, Emelianenko M, Ju L. Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SIAM Journal on Numerical Analysis* 2006; **44**:102–119.
19. Persson PO, Strang G. A simple mesh generator in MATLAB. *SIAM Review* 2004; **46**:329–345.
20. Cuthill E, McKee J. Reducing the bandwidth of sparse symmetric matrices. *Proceedings of the 24th National Conference*. ACM Press: New York, NY, U.S.A., 1969; 157–172.
21. Paulino GH, Menezes IFM, Gattass M, Mukherjee S. Node and element resequencing using the Laplacian of a finite element graph. Part I: general concepts and algorithm. *International Journal for Numerical Methods in Engineering* 1994; **37**(9):1511–1530.
22. Paulino GH, Menezes IFM, Gattass M, Mukherjee S. Node and element resequencing using the Laplacian of a finite element graph. Part II: implementation and numerical results. *International Journal for Numerical Methods in Engineering* 1994; **37**(9):1531–1555.
23. Dongarra JJ, Duff IS, Sorensen DC, van der Vorst HA. *Numerical Linear Algebra for High-Performance Computers*. SIAM: Philadelphia, PA, U.S.A., 1998.
24. Wang S, de Sturler E, Paulino GH. Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering* 2007; **69**(12):2441–2468.
25. Sukumar N, Tabarraei A. Conforming polygonal finite elements. *International Journal for Numerical Methods in Engineering* 2004; **61**(12):2045–2066.
26. Sibson R. A vector identity for the Dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society* 1980; **87**:151–155.
27. Hughes TJR. *The Finite Element Method*: *Linear Static and Dynamic Finite Element Analysis*. Dover: New York, 2000.
28. Sukumar N, Moran B, Semenov AY, Belikov VV. Natural neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering* 2001; **50**(1):1–27.

DOI: 10.1002/nme

29. Tabarraei A, Sukumar N. Application of polygonal finite elements in linear elasticity. *International Journal of Computational Methods* 2006; **3**(4):503–520.
30. Lyness JN, Monegato G. Quadrature rules for regions having regular hexagonal symmetry. *SIAM Journal on Numerical Analysis* 1977; **14**(2):283–295.
31. Natarajan S, Bordas S, Mahapatra DR. Numerical integration over arbitrary polygonal domains based on Schwarz–Christoffel conformal mapping. *International Journal for Numerical Methods in Engineering* 2009; **80**(1):103–134.
32. Mousavi SE, Xiao H, Sukumar N. Generalized Gaussian quadrature rules on arbitrary polygons. *International Journal for Numerical Methods in Engineering* 2009; DOI: 10.1002/nme.2759.
33. Cook RD, Malkus DS, Plesha ME. *Concepts and Applications of Finite Element Analysis* (4th edn). Wiley: New York, 2002.
34. Yuqui L, Yin X. Generalized conforming triangular membrane element with vertex rigid rotational freedoms. *Finite Elements in Analysis and Design* 1994; **17**:259–271.
35. Tabarraei A, Sukumar N. Adaptive computations using material forces and residual-based error estimators on quadtree meshes. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(25–28):2657–2680.
36. Huang Y, Qin H, Desheng W. Centroidal Voronoi tessellation based finite element superconvergence. *International Journal for Numerical Methods in Engineering* 2008; **76**:1819–1839.
37. Bendsoe MP. Optimal design as material distribution. *Structural Optimization* 1989; **1**:193–202.
38. Rahmatalla SF, Swan CC. A Q4/Q4 continuum structural topology optimization implementation. *Structural and Multidisciplinary Optimization* 2004; **27**(1–2):130–135.
39. Matsui K, Terada K. Continuous approximation of material distribution for topology optimization. *International Journal for Numerical Methods in Engineering* 2004; **59**(14):1925–1944.
40. Guest JK, Prevost JH, Belytschko T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering* 2004; **61**(2):238–254.
41. Almeida SRM, Paulino GH, Silva ECN. A simple and effective inverse projection scheme for void distribution in topology optimization. *Structural and Multidisciplinary Optimization* 2009; **39**(4):359–371.
42. Bourdin B. Filters in topology optimization. *International Journal for Numerical Methods in Engineering* 2001; **50**(9):2143–2158.
43. Celes W, Paulino GH, Espinha R. A compact adjacency-based topological data structure for finite element mesh representation. *International Journal for Numerical Methods in Engineering* 2005; **64**(11):1529–1556.
44. Celes W, Paulino GH, Espinha R. Efficient handling of implicit entities in reduced mesh representations. *Journal of Computing and Information Science in Engineering* 2005; **5**(4):348–359.
45. Paulino GH, Celes W, Espinha R, Zhang Z. A general topology-based framework for adaptive insertion of cohesive elements in finite element meshes. *Engineering with Computers* 2008; **24**(1):59–78.
46. Svanberg K. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering* 1987; **24**(2):359–373.
47. Svanberg K. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization* 2002; **12**(2):555–573.
48. Bendsoe MP, Sigmund O. *Topology Optimization—Theory, Methods and Applications*. Springer: New York, 2003.
49. Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik* 1959; **1**:269–271.
50. Olhoff N, Bendsoe MP, Rasmussen J. On CAD-integrated structural topology and design optimization. *Computer Methods in Applied Mechanics and Engineering* 1991; **89**(1–3):259–279.
51. Rozvany GIN. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization* 2009; **37**(3):217–237.
52. Suzuki K, Kikuchi N. A homogenization method for shape and topology optimization. *Computer Methods in Applied Mechanics and Engineering* 1991; **93**(3):291–318.