Introducing a general polygonal primitive for feature mapping-based topology optimization

Yakov Zelickman[1*] and James K. Guest[1]

[1]Civil and Systems Engineering, Johns Hopkins University, 3400 North Charles St., Baltimore, Maryland, United States.

*Corresponding author(s). E-mail(s): yzelick1@jhu.edu;
Contributing authors: jkguest@jhu.edu;

Abstract

In topology optimization, feature mapping approaches allow for maintaining the simplicity of density-based methods while incorporating explicit geometrical parametrization. Existing methods often rely on geometric primitives that have analytical signed distance functions (SDF), which may offer limited design freedom or require costly numerical methods to approximate the SDF. This paper introduces a new type of general polygonal primitive that can be convex or non-convex, with an arbitrary number of vertices, the coordinates of which are assigned with design variables. As a result, the proposed parametrization is geometrically rich and explicit. Specifically, we present a new, differentiable, and efficient way to approximate the signed distance function of arbitrary polygons and develop a scheme that prevents self-intersection of polygons. The optimized designs with the proposed polygonal primitive are similar to classical results obtained with density-based methods, albeit with some minor sacrifice in performance due to the polygonal boundaries. The guaranteed straight lines of the optimized designs, however, are also beneficial in many cases, such as in reinforced concrete structures where curved boundaries are difficult to manufacture. Moreover, the explicit parametrization and the direct shape control facilitate the convenient imposition of a wide range of geometrical constraints that are not trivial with existing primitives.

Keywords: Feature Mapping, Geometrical Projection, Polygons, Topology Optimization

1 Introduction

Feature mapping methods are class of methods for shape and topology optimization where the geometry is explicitly parameterized and projected on a fixed finite elements (FE) mesh for analysis [1]. Feature mapping methods thus offer a natural representation for imposing geometrical constraints and exporting optimized designs, while keeping the numerical cost relatively low as re-meshing is not needed in most cases and the number of design variables is low [1]. Here we present a relatively brief literature review of feature mapping approaches, focusing on the different parametrization methods. For a more extensive survey and background we refer the reader to a recent review by Wein et al. [1].

A key component in feature mapping methods is an indicator function that indicates whether a point is within the boundaries of a given feature. The indicator functions are based on distances to the features, where best results a obtained when using the singed distance function (SDF) that has unit gradient value everywhere [2, 3]. Most existing studies consider simple geometrical features that have closed form and differentiable indicator functions, thereby allowing use of efficient gradient-based optimization algorithms.

One class of features that is often used in the literature can be referred to as point-centered features and includes circles, ellipses, superelipses, and approximate rectangles, which have explicit analytical indicator functions. For example, in an early paper ellipses were projected onto a fixed background mesh and a polygonization technique was used to represent the resulting implicit boundary [4]. Mei et al. [5] presented a general framework for feature mapping with simple primitives and Boolean operations using R-functions, and provided examples with ellipse, rectangle and triangle features. A hybrid feature mapping method, where rectangular features were projected using super Gaussian functions along with density-based optimization design, was proposed by Qian and Ananthasuresh [6]. A hybrid feature mapping method combining level-set optimization with parametrized circular and rectangular features has also been proposed [7]. Therein, the rectangular features were projected by Boolean smooth union operation with R-functions, which were also used to create the overall hybrid design. A similar formulation using density-based optimization as the free-form optimization approach, rather than level sets, was presented in Xia et al. [8]. In a recent study, the present authors optimized the thickness distribution of concrete slabs using rectangle features that are parameterized with double supergaussian functions [9].

Another class of features are skeleton features. These are parametrized by a skeleton function and a thickness distribution along the stem. The simplest example is a rectangular bar feature that is parametrized by its location, length, orientation and thickness, which was suggested in Bell et al. [10]. The authors then extended their work in Norato et al. [11] and proposed a bar feature with rounded ends to simplify the feature projection and sensitivity analysis, and improve the numerical performance of the optimization method. Similar bar features with slightly different projection function have also been proposed [12], and extended to allow for varying thickness [13], curved skeletons [14], and multi-material optimization [15]. Maximum length scale control for rounded bar elements has also been proposed where the basic idea was to limit the void fraction within a predefined mask area of each bar [16], following a similar idea as in density-based topology optimization [17]. A richer skeleton feature was presented in a recent study where the authors construct a 'spaghetti' feature with the aim obtaining continuous fiber reinforced structures [18]. The proposed feature has a piecewise linear skeleton with rounded corners and constant thickness, which allow for explicit signed distanced functions and their derivatives.

In recent studies Bézier and B-spline curves were used as skeleton features, resulting in rich and smooth parametrizations [19–21]. Further enrichment of B-spline skeleton features was suggested by Zhu et al. [22], Zhou et al. [23], where the features with varying features are introduced. However, general Bézier and B-spline curves do not have a closed form singed distance function [18], and a minimum distance problem should be solved for every point of interest, which may be very expensive numerically.

From the discussion above it is apparent that most existing feature mapping methods use either point-centered features or skeleton features, for which closed-form SDF can be obtained, but that offer limited geometrical freedom. In cases where more elaborate features are used, that are based on Bézier curves, expensive solution of optimization sub-problems is needed. Thus, the existing features offer limited geometric freedom, or are expensive computationally.

In this study we present a general polygonal primitive that can be both convex and non-convex and therefore results in rich geometrical representation and designs with straight line edges. Furthermore, we present a new and efficient method to compute an approximate SDF for arbitrary polygonal shapes.

The remainder of this paper is arranged as follows. In the next section we present the proposed polygonal primitive and its approximate SDF. Thereafter in Section 3 we present the feature mapping optimization approach, including the sensitivity analysis. In Section 4 we present three numerical examples, followed by some concluding remarks in Section 5.

2 Polygon Primitive

In this section we present a general polygonal primitive (PP), that can be then projected onto a fixed finite element (FE) mesh. The PP can have any number of vertices and can be both convex and non-convex, offering a fairly rich design representation. It is noted, however, that the PP cannot intersect itself and thus will require some regulation as will discussed in Section 2.5.

2.1 Polygon geometry

The proposed polygonal primitive is a 2D closed polygon with m vertices. Assuming without loss of generality that the polygon is on the $x-y$ plane, the corresponding coordinates vector is

$$\mathbf{p} = (p_1, \ldots, p_i, \ldots, p_m)^T \quad \text{with} \quad p_i = \left(x_i,\ y_i\right),$$

where the vertices p_i are arranged in a clockwise order. Since the polygon is closed it has also m edges, where the i^{th} edge connects the i^{th} and the $(i+1)^{\text{th}}$ vertices, and can be represented by a vector

$$\mathbf{e}_i = \begin{pmatrix} x_{i+1} \\ y_{i+1} \\ 0 \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \\ 0 \end{pmatrix} \quad \text{with} \quad i = 1 \dots m. \tag{1}$$

We note that all i indices in this manuscript are cyclic indices with respect to m. Thus, if $i := m$, then $i + 1 := 1$, and if $i := 1$ then $i - 1 := m$. As mentioned, a PP can be both convex and non-convex, hence we distinguish between so-called ear and mouth vertices, which have interior angle that is smaller and grater than 180^0, respectively. Mathematically, we can sort the vertices to mouth and ear vertices by the cross product of the connected edge vectors to each vertex

$$s_i = \frac{\mathbf{e}_i \times \mathbf{e}_{i-1}}{|\mathbf{e}_i \times \mathbf{e}_{i-1}|} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

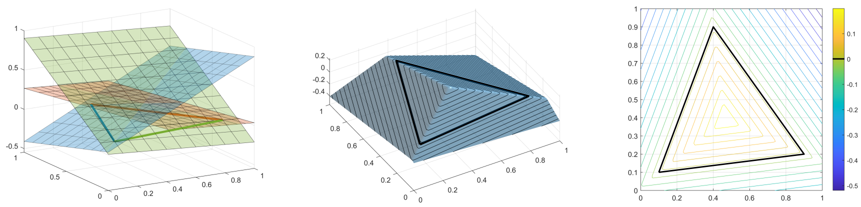
where $s_i = 1$ indicates that the i^{th} vertex is an ear vertex, and $s_i = -1$ indicates that it is a mouth vertex.

2.2 Signed distance function

The signed distance function (SDF) of an object returns the shortest distance from a point to the object, where points at the interior of the object and at the exterior have distances with opposite signs. The SDF are extensively used in many fields such as: computational graphics, [24], robotics [25], and navigation [26], as well as in level-set optimization methods [27].

Some shapes, such as circles and ellipses, have analytical SDFs, while other shapes have analytical approximated SDFs, such as rectangles and rounded bars. However, for general shapes there is no analytical SDF and numerical methods are mostly used, where the object is usually implicitly defined, e.g. with pixel information. Traditional methods for computing the signed distance fields include for example the Fast Marching Method [28] or flow methods [2, 29], where the Eikonal PDE is solved using different upwind finite differences schemes. More recently, artificial intelligence models were suggested to obtain a signed distance fields for a given body [30, 31].

In this study we present an explicit SDF of arbitrary explicitly defined polygons. The proposed method is founded upon two key realizations. The first is that the SDF of a line, or for brevity the edge SDF, is simply an inclined plane with angle of 45^0 with respect to working plane, on which we define the distance. The second is that we can approximate the SDF of an arbitrary polygon using a set of Boolean operations on the SDF of its edge lines. In this regard we note that the exact SDF of an arbitrary polygon can be obtained by indentifing the loci of point that are closest to the polygon's vertices and computing the SDF there accordingly. However, as will be discussed in Section 2.3, in the context of feature mapping we use the SDF to distinguish between the interior and the exterior of the polygon. Thus, we mostly care



(a) Edge SDFs **(b)** Min. envelope **(c)** Approximated SDF

Fig. 1. An SDF of a convex polygon is approximated by the min. envelope of its edge SDFs

about the sign information of the SDF, and the edge SDFs are sufficient. Moreover, identification of the vertex loci is not trivial for general polygons.

In this study we adopt a sign convention where interior points have positive distance whereas exterior points have negative distance. Specifically, the SDF of any convex polygon can be approximated by the minimum of its edge SDFs

$$z = \min_{i \in I} \bar{z}_i$$

where z is the SDF of the polygon, $\bar{z}_i$ is the SDF of the i^{th} edge, and I is a set of all edge indices that belong to the polygon. In Fig 1 we show an example of such a convex polygon, specifically a triangle, with its edge SDFs and the resultant SDF of the polygon shown.

Mathematically, to obtain the SDF for an edge $\mathbf{e}_i$, that is defined by the vertices p_i and p_{i+1}, we find an auxiliary point o such that it belongs to the SDF plane of $\mathbf{e}_i$ but not to the line defined by it. We may then construct the plane function that passes through all three points, i.e. p_i, p_{i+1}, and o. The distance of points p_i and p_{i+1} to the edge is obviously zero, whereas the distance of point o is equal to the perpendicular distance to edge $\mathbf{e}_i$

$$d_{oi} = \frac{|o_x + o_y + B|}{\sqrt{A^2 + 1}}, \quad \text{with} \quad A = -\frac{y_i - y_{i+1}}{x_i - x_{i+1}}, \, B = -Ax_{i+1} - y_{i+1}. \tag{2}$$

We choose the point o to be p_{i-1}, and therefore if vertex p_i is an ear vertex then the signed distance of p_{i-1} is positive. If the vertex is a mouth vertex then the signed distance of p_{i-1} is negative. Thus, the sign of the translation in the out of plane direction of point o is conveniently defined by the normal direction of vertex p_i, i.e. s_i

$$o_z = s_i d_{io}$$

and the SDF of edge $\mathbf{e}_i$ is

$$\bar{z}_i(x,y) = \left(-\frac{a}{c}x - \frac{b}{c}y + \frac{d}{c}\right) \quad \text{with} \quad \begin{cases} a = (y_{i+1} - y_i)\, o_z \\ b = (x_{i+1} - x_i)\, o_z \\ c = (x_{i-1} - x_i)(y_{i-1} - y_{i+1}) - (y_{i-1} - y_i)(x_{i-1} - x_{i+1}) \\ d = x_{i-1}a + y_{i-1}b + o_z c. \end{cases} \tag{3}$$

In non-convex polygons, such as the polygon in Fig. 2a, using the minimum envelope of the edge SDFs is not sufficient because it convexifies the polygon by trimming it. Therefore, we first identify all concave segments in the non-convex polygon, which are consecutive sequences of edges that are connected with mouth vertices. In Fig. 2a the polygon has two concave segments: the first includes edges 3,4, and 5, and the second includes the edges 7 and 8. Next, for each concave segment we construct a separate SDF by taking the maximum of all SDFs of edges, as presented in Fig. 2b. For a polygon with N_c concave segments, the SDF of the k^{th} concave segment is then

$$\check{z}_k = \max_{i \in I_k} \bar{z}_i \quad \text{with} \quad k \in \{1 \ldots N_c\}\,,$$

where I_k is an index set that includes all edges that belong to the k^{th} concave segment. In Fig. 2a the concave segments' index sets are $I_1 = \{2, 3, 4\}$, and $I_2 = \{7, 8\}$. The SDF of the non-convex polygon is then obtained by the minimum of the concave segments' SDF and all the edge SDFs that are not associated with any of the concave segments, which we will refer to as convex edges

$$z = \min\{\bar{z}_i,\, \check{z}_k\} \quad \text{with} \quad i \in I_0 = I \backslash \cup I_k, \quad \text{and} \quad k \in \{1 \ldots N_c\}\,,$$

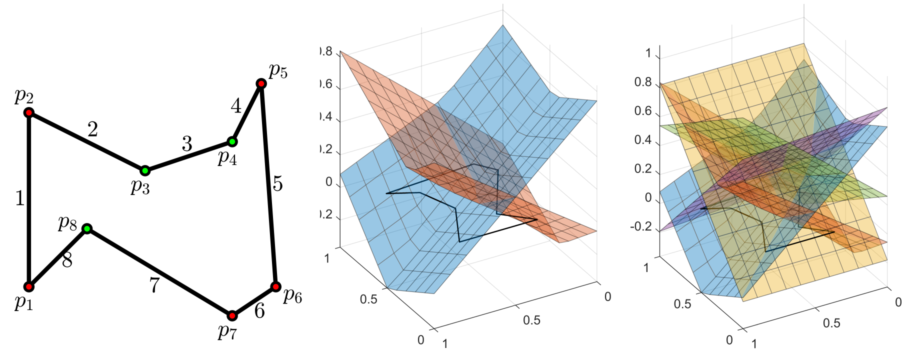
where I_0 is the set of convex edges, e.g. $I_0 = \{1, 5, 6\}$ in Fig. 2a.

In practice, an optimization problem likely will have more than one polygon, in which case we will obtain the signed distance function by another max. operation, between the individual polygons. Thus the SDF of n polygons would be

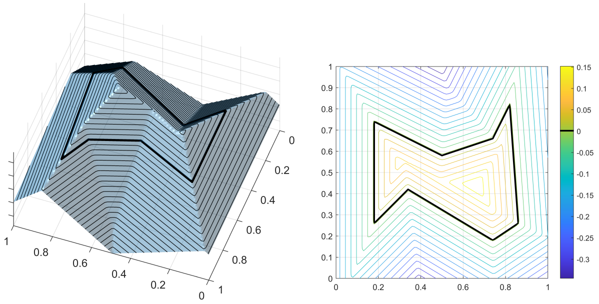
$$z = \max_j z_j \quad \text{with} \quad j \in \{1 \ldots n\}\,,$$

where z_j is the SDF of the j^{th} PP. Finally, because we want the SDF to be differentiable we approximate the min. and max. operations with smooth p-norm functions. As the SDF values are both negative and positive, we shift the SDF such that it has only positive values for the p-norm approximation and then shift it back to obtain the real SDF values. Thus, the SDF of n polygons, each with m vertices, is approximated by

$$z\,(x,y) = \left(\sum_{j=1}^{n} (z_j + |\alpha|)^q\right)^{1/q} - |\alpha|\,, \quad \text{with} \quad \alpha = \min_j\left(\min\left(z_j\right)\right) \tag{4}$$



(a) Non-convex polygon **(b)** Concave segments' SDF **(c)** All sub-SDFs



(d) min. envelope **(e)** Approximated SDF

Fig. 2. An SDF of a non-convex polygon is approximated by the min. envelope of its free edge SDFs and the Concave segments SDFs.

and

$$z_j = \left(\sum_{i \in I_{0j}} (\bar{z}_i + |\alpha_j|)^{-p} + \sum_{k=1}^{N_{cj}} \check{z}_k^{-p} \right)^{-1/p} - |\alpha_j| \,,$$

with

$$\alpha_j = \min_{i \in I_j} (\min(\bar{z}_i)) \,, \quad \check{z}_k = \left(\sum_{i \in I_{kj}} (\bar{z}_i + |\alpha_j|)^p \right)^{1/p} \quad \text{and} \quad I_{0j} = I_j \setminus \cup_k I_{kj}.$$

In the expression above, I_j, I_{0j},and I_{kj} are index sets of all edges, convex edges, and those belonging to the k^{th} concave segment of the j^{th} PP respectively; N_{cj} is the number of concave segments at the j^{th} PP.

2.3 Projection

In feature mapping methods the design is explicitly parameterized through geometrical primitives, which are projected onto a background, fixed, finite element mesh for analysis. Thus, elements that are inside and outside the boundaries of the PP will generally have different projected material properties.

For a given design with a SDF z, we can conveniently define the interior and exterior domains

$$\Omega_{in} = \{(x, y) \,|\, z(x, y) \geq 0\}, \quad \Omega_{out} = \{(x, y) \,|\, z(x, y) < 0\},$$

where $\Omega_{in} \cap \Omega_{out} = \emptyset$, $\Omega_{in} \cup \Omega_{out} = \Omega$, and Ω is the entire design domain. We define an indicator function that equals one for points that belong to Ω_{in}, and zero otherwise. To make the projection operation differentiable we approximate the indicator function with a smooth Heaviside function [32, 33], with threshold $\eta = 0$, leading to the following expression

$$\zeta = h_\beta(z) = \frac{1}{2}\left(1 + \frac{\tanh(\beta z)}{\tanh(\beta)}\right), \tag{5}$$

where β controls the sharpness of the approximation.

For the projection we adopt the pseudo density approach where the projected elemental properties are proportional to the pseudo density, which equals the elemental area fraction that is within the PP domains. Thus, the elemental pseudo densities can be obtained by integration of the indicator function over each element, calculated numerically using a Gauss quadrature rule

$$\rho_\ell = \int_{A_\ell} \zeta dA \approx \sum_{g=1}^{N_{g\ell}} w_g \zeta_g, \tag{6}$$

where A_ℓ is the elemental area, $N_{g\ell}$ is the number of Gauss integration points in element ℓ, and ζ_g and w_g are the indicator function at the g^{th} Gauss point and the corresponding weight, respectively.

It has been shown that the numerical performance is strongly related to the integration accuracy, and usually rich quadrature rules are required for successful feature mapping optimization [1]. However, for elements that are far enough from the boundaries of the design, ζ converges to zero or one, depending on the sign of z, and coarse integration may be sufficient. Moreover, even for elements that are close the boundaries and require fine integration, the variation in ζ is caused only by PP that are close to those elements. In other words, for each individual PP, fine sampling of the SDF is necessary only at the vicinity of the boundaries of the PP - the transition zone. Beyond the transition zones the Heaviside projection reaches a plateau and its derivative approaches zero. We use this property to identify the elements for which coarse sampling of the SDFs is sufficient with the aim to reduce the computational cost of the projection.

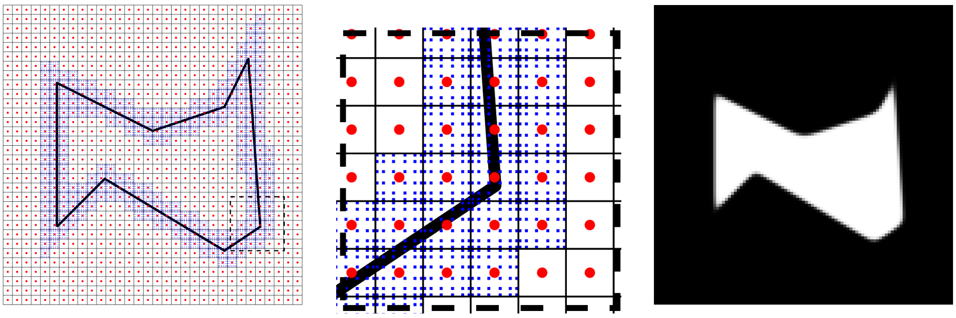


Fig. 3. Hierarchical numerical integration with richer Gauss quadrature at the vicinity of the PP boundaries.

For each PP we perform a two stage process, where we first evaluate at the centroids of all elements the SDF, a local indicator function $\zeta_j = h_\beta(z_j)$ and its derivatives with respect to the SDF, where the subscript j emphasized that the SDF and the indicator function are of a specific PP. Next, for elements with $\frac{\partial \zeta_j}{\partial z_j} \geq 1 \times 10^{-4}$ (ζ_j is monotonically increasing function) we evaluate the SDF at all Gauss points. Otherwise, we assign to all Gauss points the SDF that was evaluated at the centroid of the element. Here we use a 6×6 quadrature rule, as illustrated in Fig. 3 for sampling points of a non-convex PP. As will be discussed in Section 3.3, we implement a continuation on β, eventually resulting in fairly sharp projection with narrow transition zones, where most elements would have coarse integration, potentially impairing the numerical performance. Therefore, we define a minimum width of 0.1 times the minimal domain size where fine sampling is ensured.

Once the pseudo densities of the elements are known, the elemental properties can be projected. Herein we choose to project the void phase. Thus, the modulus of elasticity of the ℓ^{th} element is

$$E_\ell = E_{\max} - (E_{\max} - E_{\min})\, \rho_\ell^s. \quad \text{with} \quad s = 3, \tag{7}$$

where the reason for adding the power s will be discussed in Section 3.3. Once E_ℓ is known, the elemental stiffness matrices, $\mathbf{k}_\ell$ are computed and the global stiffness matrix, $\mathbf{K}$, that correspond to current layout of the PP, is assembled.

2.4 Numerical complexity

In previous section we stated that accurate integration is necessary for successful convergence. Therefore, at every optimization iteration the indicator function and the SDF must be calculated for an order of 10^6 to 10^7 Gauss integration points, making the numerical complexity very important. From a computational perspective, the proposed method is composed of two main parts: computation of the edge SDF in Eq. (3), and the Boolean operations in Eq. (4). Both parts require evaluation of explicit functions and therefore linearly depend on the number of the Gauss integration points , N. Hence, the complexity of the proposed projection method is $\mathcal{O}(N)$.

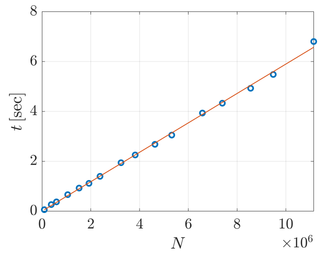


Fig. 4. Numerical complexity.

In Figure 4 we plot the computational time spent on computing the SDF (and its derivatives as will presented in the next section) for different values of N, confirming the linear complexity. We note that for very large N, memory management becomes dominant and the numerical complexity becomes superlinear.

Additional improvement in the computation time can be achieved by using a more sophisticated sampling strategy with more than two integration densities. Finally, we note that all operations are spatially local and therefore parallelization is straightforward, likely allowing for additional savings in compute time.

2.5 Design parametrization and regularization

In previous subsections we presented the SDF of an arbitrary polygon that is explicit both in x and y, as well as in the coordinates of the polygon. It is therefore natural to choose the coordinates of the PP as design variables. Considering a design space with n polygon primitives, each with m vertices, the vector of the design variables is

$$\mathbf{X} = (\mathbf{p}_1 \ldots \mathbf{p}_j \ldots \mathbf{p}_n) \quad \text{with} \quad \mathbf{p}_j = \{p_i \,|i \in J_j\} \,,$$

where J_j is an index set of all vertices that belong to the j^{th} PP. This explicit parametrization is very rich since the vertices move independently and a PP with m sides span the entire m-gon family. For example, a four sided PP can equally be a rectangle, trapezoid, rhombus, triangle or even an arrow-head. However, the movement of the vertices is not entirely free since each polygon must not intersect itself, requiring some restriction on the design space.

Self intersection of a polygon means that the distance of at least one vertex to one of the edges is zero. Thus, at each iteration we limit the allowed movement range of the vertices such that the distance between all vertices and edges will remain positive. Looking at a vertex i in a polygon with m edges, the vertex can intersect any of the edges that do not share the vertex i. Additionally, it is possible that the edges that do share the vertex will 'hit' one of the mouth vertices in the polygon, if they exist. Thus, for a vertex i in a polygon with m edges, and with a set of mouth vertices J_M, the distances that potentially limit the movement of vertex i are:

$$d_i = \left\{d_{i1}, \ldots, d_{i(i-2)}, d_{k(i-1)}, d_{ki}, d_{i(i+1)}, \ldots, d_{im}\right\} \quad \text{with} \quad k \in J_M \backslash \left\{i-1, i, i+1\right\},$$

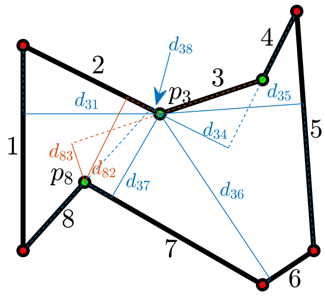


Fig. 5. Shape regulation.

where d_{ie} is the perpendicular distance between vertex i to edge e, and is given in Eq. (2). In figure 5 we present again the non-convex polygon, with the limiting distances of vertex p_3, where the blue color marks distances to non-adjacent edges and the orange color indicates distances between the adjacent edges to mouth vertices (in this case only to vertex p_8).

A natural approach to define the move limits of the coordinates would be to limit the maximal change in the coordinate of vertex i based on the minimum limiting distance d_i. However, such an approach might lead vertices to become trapped in their location when any of the limiting distances approach zero. For example, the distance between vertex p_3 to edge e_8, i.e., d_{38} in Fig. 5, is very small and vertex p_3 would have very restrictive move limits. However, d_{38} really limits the movement of vertex p_3 only to the left and to the top, whereas p_3 can move more freely to the right or to the bottom. Therefore, we sort the limiting distances based on the limit that they impose. Mathematically, we look at the derivatives of the limiting distances with respect to movement of vertex i and group the limiting distance based on the sign of the derivatives. For example, $\frac{\partial d_{ie}}{\partial x_i} > 0$ means that edge e limits the movement of vertex i from left, and similarly for $\frac{\partial d_{ie}}{\partial x_i} < 0$ and for derivatives with respect to y_i. Thus, we structure the move limits of the vertices as follows:

$$-\min(d_{ix-}) \leq \Delta_{xi} \leq \min(d_{ix+}) \quad \text{and} -\min(d_{iy-}) \leq \Delta_{yi} \leq \min(d_{iy+})$$

with

$$d_{ix+} = \left\{ d_i \,\middle|\, \frac{\partial d_i}{\partial x_i} < 0 \right\}, \quad d_{ix-} = \left\{ d_i \,\middle|\, \frac{\partial d_i}{\partial x_i} > 0 \right\},$$
$$d_{iy+} = \left\{ d_i \,\middle|\, \frac{\partial d_i}{\partial y_i} < 0 \right\}, \quad d_{iy-} = \left\{ d_i \,\middle|\, \frac{\partial d_i}{\partial y_i} > 0 \right\},$$

2.6 Limitations

The proposed formulation is theoretically general and can cope with polygons with arbitrary number of vertices and most of the possible layouts. However, despite the

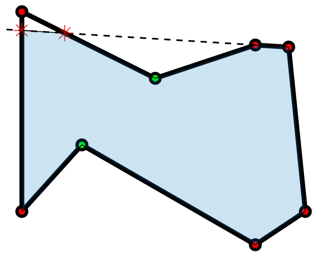


Fig. 6. Trimming effect.

restriction schemes that are imposed, the proposed method has several limitations that need to be underscored.

The main limitation is related to a case where a line defined by the immediate edge next to a concave segment intersects the polygon. In this case the polygon will be trimmed along this line, effectively convexifying the polygon, as can be seen in Figure 6. For this to happen at least four vertices should be approximately aligned, the likelihood of which increases with the number of vertices. Therefore, trimming can not happen in a polygon with 4 edges (or less) and, in our experience, has not happened in polygons with 6 edges or less. However, for polygons with 7 edges and greater, trimming may occur, and might lead to inferior results, as will be demonstrated in Section 4. It should be noted that the projection and its sensitivity analysis remain consistent even when trimming occurs. The challenge of the trimming affect will need to be addressed in future research.

3 Optimization with Polygon primitives

A key property of the PP is that it is an explicit and smooth function of the coordinates of its vertices. This enables to use efficient gradient based optimization algorithms, herein chosen as the Method of Moving Asymptotes (MMA) [34]. In this section we present the adopted problem formulation and the sensitivity analysis (SA) necessary to drive MMA.

3.1 Problem formulation

We focus on introducing the Polygonal Primitive in the context of the well-known minimum compliance problem, expressed as

$$\begin{aligned} \min_{\mathbf{x}} \quad & c \\ \text{s.t.} \quad & < v^{*} \\ \text{with} \quad & \mathbf{K}\mathbf{u} = \mathbf{f}, \end{aligned} \tag{8}$$

where $c = \mathbf{f}^T\mathbf{u}$ is the structural compliance, $v = \sum_\ell \rho_\ell$ is the volume fraction, asterisk * indicates a target value, and $\mathbf{u}$ and $\mathbf{f}$ are the nodal displacement and force vectors, respectively.

We also aim to demonstrate the ease with which geometrical constraints can be added, thanks to the explicit parametrization of PP. Therefore, in Section 4.3 we present and discuss three examples of such constraints.

3.2 Sensitivity analysis

The SA of the compliance objective function is an implicit function of the design variables through the state equations. Following the adjoint approach [35], the derivative of the compliance with respect to the x coordinate (or the y coordinate) of the i^{th} vertex is

$$\frac{\partial c}{\partial x_i} = \mathbf{u}^T \frac{\partial \mathbf{K}}{\partial x_i}\mathbf{u} \quad \text{with} \quad \frac{\partial \mathbf{K}}{\partial x_i} = \Sigma_\ell \frac{\partial \mathbf{k}_\ell}{\partial x_i}.$$

where Σ_ℓ is the elemental assembly operation. The derivative of the elemental stiffness matrices with respect to design variables can be expressed as

$$\frac{\partial \mathbf{k}_\ell}{\partial x_i} = \frac{\partial \mathbf{k}_\ell}{\partial E_\ell}\frac{\partial E_\ell}{\partial x_i} \quad \text{with} \quad \frac{\partial E_\ell}{\partial x_i} = -\left(E_{max} - E_{min}\right) s\rho_\ell^{s-1}\frac{\partial \rho_\ell}{\partial x_i}.$$

The derivative of the pseudo density with respect to a vertex coordinate is obtained by differentiating Eq. (5) and substituting into Eq. (6)

$$\frac{\partial \rho_\ell}{\partial x_i} = \sum_{g=1}^{N_{g\ell}} w_g \frac{1 - \left(\tanh\left(\beta z_g\right)\right)^2 \beta}{\tanh\left(\beta\right)} \frac{\partial z_g}{\partial x_i}, \tag{9}$$

where z_g and $\frac{\partial z_g}{\partial x_i}$ are the SDF and its derivative with respect to x_i, evaluated at Gauss point g. To obtain $\frac{\partial z}{\partial x_i}$ we differentiate Eq.(4),

$$\frac{\partial z}{\partial x_i} = \left(\sum_{j=1}^{n} z_j^q\right)^{\frac{1}{q}-1} \sum_{j=1}^{n} z_j^{q-1}\frac{\partial z_j}{\partial x_i},$$

where $\frac{\partial z_j}{\partial x_i} = 0$ if $p_i \notin J_j$, otherwise we keep in mind that change in x_i affects only the SDFs of the adjacent edges, i.e., $\bar{z}_{i-1}$ and $\bar{z}_i$, and the derivative is therefore

$$\frac{\partial z_j}{\partial x_i} = \left(\sum_{t\in I_{oj}} \bar{z}_t^{-p} + \sum_{k=1}^{N_{cj}} \check{z}_k^{-p}\right)^{-\frac{1}{p}-1} \left(C_{i-1}\frac{\partial \bar{z}_{i-1}}{\partial x_i} + C_i \frac{\partial \bar{z}_i}{\partial x_i}\right),$$

with

$$C_\theta = \begin{cases} \bar{z}_\theta^{-(p+1)} & \theta \in I_{0j} \\ \left(\sum_{t \in I_{jk}} \bar{z}_t^p \right)^{-2} \bar{z}_\theta^{p-1} & \theta \in I_{jk}, \end{cases}$$

where both t and θ are utilitarian indices for this equation. Finally, the derivative of the edge SDFs with respect to the coordinates of the PP is calculated by differentiating Eq. (3), where one should keep in mind that for $\bar{z}_i$ we chose the auxiliary point o to be p_{i-1}.

The sensitivity analysis of the volume constraint now becomes straightforward with

$$\frac{\partial v}{\partial x_i} = \sum_\ell \frac{\partial \rho_\ell}{\partial x_i},$$

where $\frac{\partial \rho_\ell}{\partial x_i}$ is given in Eq. (9). In fact, once we obtain the derivatives of the pseudo densities with respect to the PP coordinates, many other functionals can be added to formulation in a regular fashion.

3.3 Implementation

The proposed polygon projection method includes four main parameters that govern the projection. In this subsection we discuss their effect on the optimization and suggest appropriate values.

The first two parameters, q and p, are related to the approximation of the SDF using p-norm functions. A characteristic feature of p-norm functions is that low power values soften the max. and min. approximations. Thus, as p and q decrease, the approximated max. and min. values are increasingly overestimated and underestimated, respectively. In the context of the SDF approximations this means that reducing p and q smooths the corners of the polygon, shrinks the individual PP up to a point that they vanish, and merges adjacent PP, as can be seen in Figure 7 that shows the contour lines for different power values. Therefore, p and q should be chosen to be large enough to reasonably preserve the features of the projected PP, but not so large that p and q result in numerical instabilities associated with round-off errors. In our experience $p = 70$, and $q = 200$ results in good representation of the projected PP and does not lead to any numerical issues for the problems considered herein.

The parameter β in Eq. (5) controls the sharpness of the projection, where clear designs require crisp projection with distinct phases of material and void. However, the sensitivity of the projection is important, where smoother projection allows the optimization to 'see' further and enables smoother design changes. Therefore, we implement a continuation scheme on β. In the early stages of the optimization we facilitate design changes by having smooth projection with $\beta = \beta_i$ and β_i being small. As the optimization progresses, we gradually increase the sharpness of the projection based on the convergence of the optimization to a final value of β_f. A similar strategy has been used by many density-based projection methods, for example in [36–38]. Specifically herein, each time that the change in the objective function is smaller than

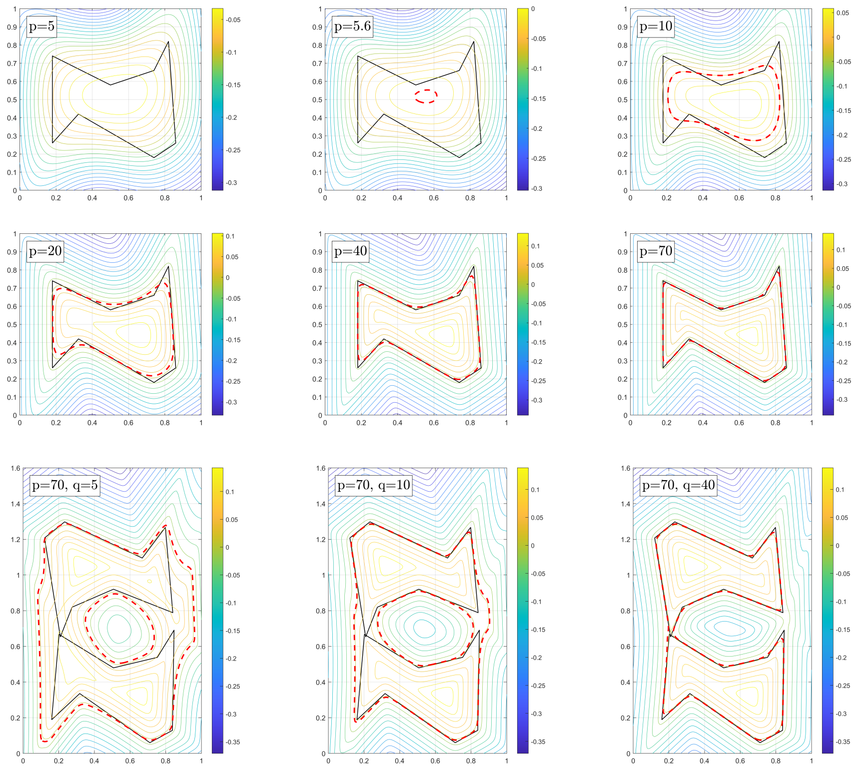


Fig. 7. Contour lines with different values of p and q, where the dashed red line indicates the zero contour line and hence the contour of the projected polygon. Low values of p shrink the individual PP, while small values of q merge adjacent PP

1×10^{-4} for five consecutive iterations we update β as follows

$$\beta_{new} = \min\left(1.5\beta_{old},\, \beta_f\right).$$

The initial and final values of β are $\beta_i = 10$, and $\beta_f = 1000$.

The smooth projection in the early optimization stages promotes designs with smoothly varying densities throughout the entire design domain, similar to a density distribution that is obtained with classical density-based topology optimization without penalization. At this early stage, the PP tend to cluster and partially overlap. As the projection sharpness increases, the PP struggle to separate, which eventually results in convergence to poor optimized designs. To resolve this we add the power s in Eq. (7) to penalize intermediate densities from the very beginning of the optimization, which steers the PP to improved locations and leads to better results.

(a) **(b)** $c_{ref} = 0.599$ **(c)** $c_{opt} = 0.607$

Fig. 8. (a) Problem setup for the Cantilever structure. (b) Density based optimized design. (c) Optimized design with polygonal features

4 Numerical examples

In this section we present three numerical examples that illustrate the richness of the proposed PP for feature mapping-based topology optimization. We will draw comparisons to reference designs that are obtained using a density-based method and assess the structural performance "cost" of having designs with straight lines. In all examples the plane finite elements have side length of $\frac{1}{128}$, and the formulation includes only the volume constraint, unless it is stated otherwise.

4.1 Cantilever structure

The first example is a 1×2 cantilever that is supported along its left edge and is loaded with a vertical unit force that is distributed over 0.2 length units along its right edge, as can be seen in Figure 8a. We set the elastic modulus and the Poisson ratio to $E = 100$ and $\nu = 0.2$, and the target volume fraction to $v^* = 0.5$. The design space for the optimization is defined with eight PP ($n = 8$) and three sided polygons ($m = 3$), thus we expect to have simple optimized designs with no more than eight holes. The initial design is automatically set such that: the proportions of the PP grid are as close as possible to the proportions of the design domain (e.g. a 2 by 4 grid for $n = 8$); the volume constraint is satisfied; and all PP are regular polygons with the first vertex laying to the right of their centroid on the horizontal axis passing through it.

Before optimizing with the proposed polygon projection method, we first establish a reference design found using the classical SIMP method [39, 40] with a filter radius of 0.11, which promotes designs with thick features and small amount of holes [36]. The physical density is obtained using a regularized Heaviside function with $\eta = 0.5$ and a continuation of the sharpness [33]. The density based optimization produces the reference design presented in Figure 8b, and has a compliance value of $c_{ref} = 0.599$.

Next, we optimize using the proposed polygon projection method, resulting in optimized design that greatly resembles the reference design but is entirely made of straight lines, as can be seen in Figure 8c. The compliance value of the optimized design is $c_{opt} = 0.607$, which is slightly worse than the compliance of the reference design and reflects that the design space has been restricted to contain only straight lines. The optimization converges after 212 iterations and Figure 9 shows how the design evolves during the optimization. The top left figure depicts the initial design and the PP numbering, where the PP are sequentially numbered from bottom up and

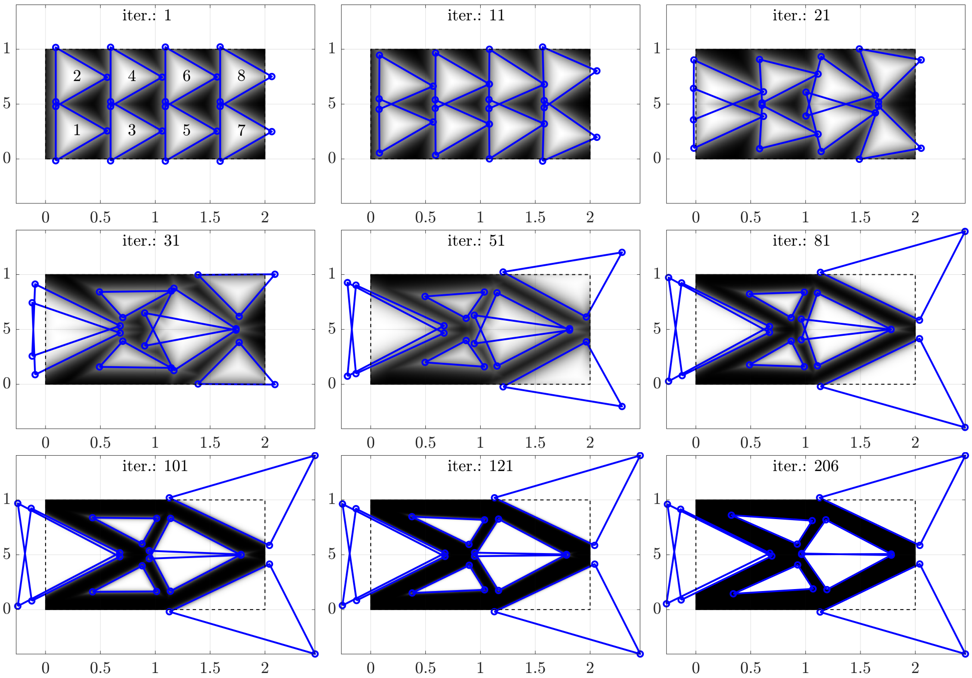


Fig. 9. Design convergence of the cantilever optimization problem. The continuation of β is evident with the increasing sharpness as the optimization progresses

then from left to right. Additionally, it can be seen in Figure 9 that the vertices of the polygons exit the FE mesh, illustrating the characteristic separation between the analysis and the design spaces.

The design space can be enriched by increasing the number of vertices per polygon and/or by increasing the number of polygons. In both cases we expect the optimized design to approach the reference design as we enrich the design space.

To demonstrate this, we first keep the number of polygons fixed, with $n = 8$, and vary the number of edges per polygon, with $m = \{3, 4, 5, 6, 7, 8\}$. The optimized designs are presented in Figure 10, where it is evident that the optimization converged to essentially identical solutions, with some minor non-convexity for $m = 7$ and $m = 8$. However, the compliance slightly improves in general as the number of edges increases, with the exception of $m = 4$, and $m = 8$ designs. In the case of $m = 4$, when we reoptimize with the initial orientation of the polygons rotated by 45^0 (i.e. PP edges aligned with the problem axes) the optimization converges to a similar design but with compliance value of $c = 0.6054$. This result fits well in the general trend but also illustrates the high non-convexity of the problem. In the case of $m = 8$, a possible explanation for the worse than expected compliance value is the trimming phenomena that was discussed in Section 2.6 and can be seen in the optimized design with $m = 8$, at the tip of the cantilever.

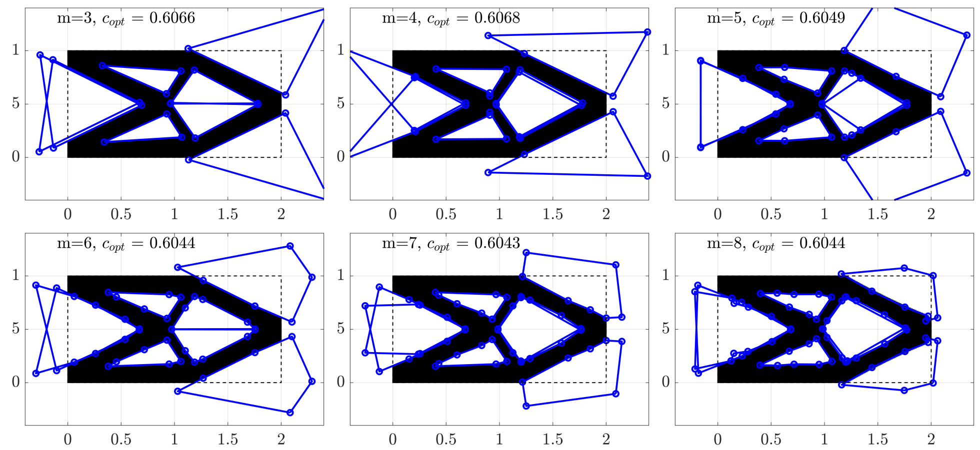


Fig. 10. Optimized designs found when using a different number of edges per polygon.

We now fix the number of vertices per polygon to $m = 4$, and consider different numbers of PP, with $n = \{8, 15, 24, 32, 60\}$. Some of the optimized results are plotted in the top row of Figure 11, where the optimization converges to similar designs in all cases. Interestingly, for $n \geq 24$ the objective function gets slightly worse as the number of PP increases. This is quite surprising, as one would expect to generate more detailed designs with more members and lower compliance magnitudes as number of PP increases.

This is likely caused by the low initial values of β. As discussed in Section 3.3, this facilitates exploration of the design space by smearing the projection but also encourages PP to merge. To demonstrate this, we present in the bottom row of Figure 11 optimized designs that were obtained also with $n = \{8, 15, 24, 32\}$ but with increased initial projection sharpness of $\beta_i = 100$. As expected, the PP have less tendency to merge, resulting in designs with more complex topologies. However, since the design changes are driven only by the sensitivity information from the immediate vicinity to the PP, the optimization straggles to find good local optima and effectively is limited to local design changes. This is especially predominant when a small number of PP need to undergo significant design changes, as can be seen when comparing the designs with $m = 8$ in Figure 11. In fact, as the number of PP in the designs space increases, significant design changes globally can be obtained with small local design changes and therefore higher value of β_i can be used, which also facilitate more complex topologies. It is also seen that some PP collapse to degenerate 1D shapes, where small voids might still be projected and impair solution performance. This could potentially be resolved by adding a topological design variable to each PP which will allow elimination of the PP without collapsing it, and will be included in a future research.

4.2 Beam and L-bracket structures

Here we present additional two benchmark examples of a simply-supported beam and an L-bracket. Figures 12a and 13a show the setup for both problems.

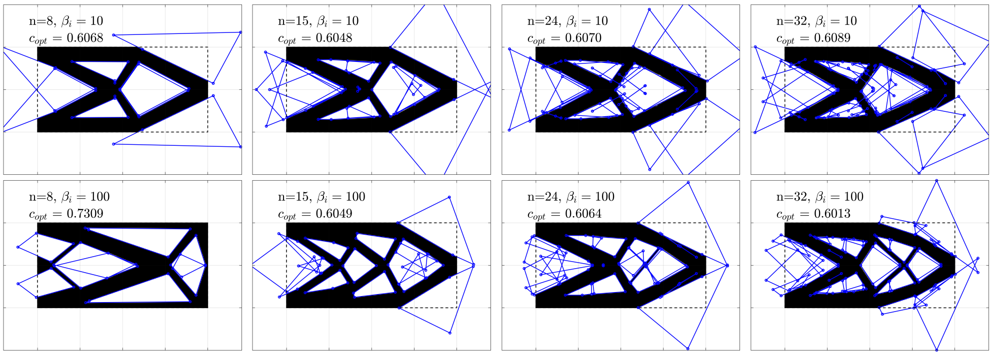
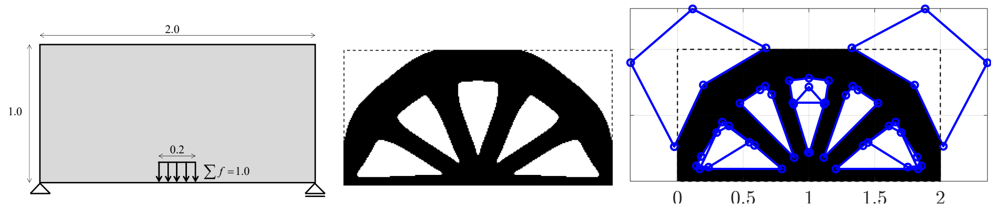


Fig. 11. Optimized design with different number of PP. Top row: optimized design with smooth initial projection. Bottom row: optimized designs with sharp initial projection.

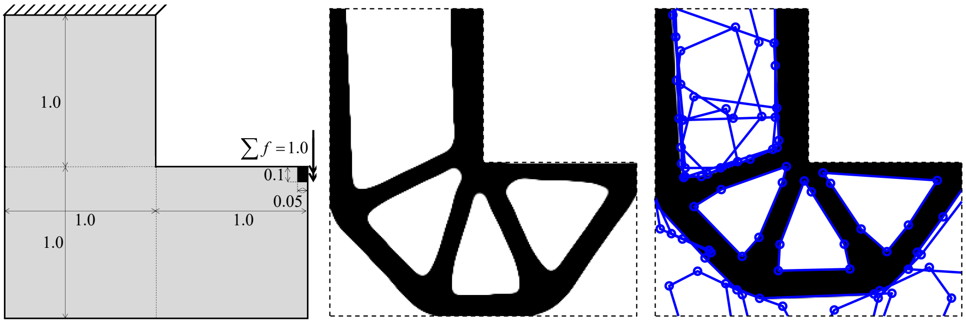


(a) **(b)** $c_{ref} = 0.0097$ **(c)** $c_{opt} = 0.118$

Fig. 12. (a) Problem setup for the beam structure. (b) Density based optimized design. (c) Optimized design with polygonal features

For the beam structure we establish a reference design with filter radius of $r_{min} = 0.11$, resulting in an optimized structure with six tension struts as presented in Figure 12b, and reference compliance magnitude of $c_{ref} = 0.097$. The optimized design found using 10 PP with 5 sides each is presented in Figure 12c and has similar layout with slightly worse compliance of $c_{ref} = 0.118$. Here the optimized design exhibits distinct non-convex shapes of the PP at the top corners of the design domain, demonstrating a successful transition of convex shapes to non-convex shapes and their projection.

The reference design for the L-bracket is obtained with filter radius of $r_{min} = 0.2$, which results in a design with three tension struts (Figure 13b) and a reference compliance magnitude of $c_{ref} = 0.849$. The design generated using 16 PP with 6 edges that are spread along the L-bracket in two 'layers' features the same topology as the reference design with slightly different shape. The structure does not quite reach the bottom boundary of the design domain and the struts are somewhat thicker. Surprisingly, the compliance of the polygonal structure is marginally smaller than the compliance of the reference design, with $c_{opt} = 0.844$. This can be at least partially attributed to the non-convexity of the minimum compliance optimization problem,



(a) **(b)** $c_{ref} = 0.849$ **(c)** $c_{opt} = 0.844$

Fig. 13. (a) Problem setup for the L-bracket. (b) Density based optimized design. (c) Optimized design with polygonal features

but is nevertheless surprising since the PP method restricts the design to have only straight edge features.

4.3 Geometrical constraints

In Section 2, we introduced the parametrization of the PP, assigning a distinct design variable to each coordinate of every vertex. This explicit shape parametrization facilitates convenient imposition of geometrical constraints, which may pertain to manufacturing, assembly, service states or other considerations. In this subsection, we illustrate the straightforward application of this idea by formulating three distinct constraints. Specifically, we will add explicit area and length constraints to the optimization formulation, and implicitly impose orientation constraints through manipulating the SA. We note that other techniques, such as dynamic move limits or the introduction of auxiliary design variables, may be also used to impose different geometrical constraints. Regardless of the approach employed, the explicit and comprehensive geometrical parametrization allows geometrical features to be naturally captured and, consequently, facilitates the application of geometrical constraints. Finally, since these constraints are expressed as explicit functions of the design variables, their sensitivity analysis is notably straightforward and omitted here for brevity.

4.3.1 Area constraint

The first constraint we add to the formulation in Eq. (9) is an area constraint. We begin by calculating the area of an arbitrary polygon as a function of the coordinates of its vertices, i.e. the design variables. First, we triangulate the polygon using the ear-clipping triangulation method [41, 42], then we compute the area of all triangle tiles, and finally sum these areas to get the area of the polygon

$$A_j = \frac{1}{2} \sum_{t \in T_j} |\mathbf{v}_{t1} \times \mathbf{v}_{t2}| \, .$$

In the expression above, A_j is the area of the jth polygon, T_j is the set of all triangles that tile the jth polygon, and $\mathbf{v}_{t1}$ and $\mathbf{v}_{t2}$ are vectors of two of the sides of the tth triangle. Once the area of the polygons are known, we can formulate a variety of constraints. In particular, we require that a subset J_A of the polygons will have a minimal area, for example to allow for enough light, ventilation, flow-rate, etc. Mathematically, instead of having an individual constraint for each polygon in J_A, we aggregate the individual constraints to a single constraint on the minimal area of the polygons in J_A, which is approximated using a p-norm function

$$\tilde{A} = \left(\sum_{j = \in J_A} A_j^{-r} \right)^{-1/r} \geq \tilde{A}^*,$$

where r is an integer power value, and $\tilde{A}^*$ is a dynamically corrected target area for the polygons in J_A. The dynamic update of the target area is essential to bridge over the p-norm approximation gap [43, 44]. Following the approach outlined in Zelickman and Amir [45], we update the threshold value every 5 iterations as follows:

$$\tilde{A}^*_{new} = \frac{\tilde{A}}{\min\limits_{j \in J_A} (A_j)} A^*,$$

where A^* is the actual target area value.

The problem setup is consistent with Figure 8a, and the design space is defined with eight quadrilateral PP initially aligned with the domain axes. The reference design for assessing the area-constrained optimization is the design in the top center plot of Figure 10, and the reference compliance value is $c_{ref} = 0.6068$. At this reference design, the area of polygons 3 and 4 is measured as 0.0756 units, and consequently we set the target area for those polygons to $A^* = 0.13$, anticipating the area constraint to be active.

The optimized design with the area constraint is presented in Fig. 14a, where the optimizer stretched the area-constrained polygons in x direction to reach the target area A^*, while the area of the remaining polygons is reduced to utilize all available material, such that the volume constraint is also active. As expected, the optimized compliance increased to $c_{opt} = 0.6173$, reflecting the performance sacrifice due to the more restricted design space.

4.3.2 Length constraint

Next, we replace the area constraint with an edge-length constraint. Specifically, we fix the length of all edges belong to a subset of length-constrained edges, I_L. The constraint is formulated in terms of the relative length difference, where we aggregate again the individual length differences to a single edge length measure, which approximates the distance of the length ratio from 1.0

$$\tilde{\lambda} = \left[\sum_{i \in I_L} (\lambda_i - 1)^r \right]^{1/r} + 1, \quad \text{with} \quad \lambda_i = \frac{\|\mathbf{e}_i\|}{L^*},$$

where L^* is the prescribed length, and r is an even number, to account both for shorter and longer length violations. The length constraint then becomes

$$\tilde{\lambda} \leq \tilde{\lambda}^*,$$

where the target value of the length measure is updated similarly to the target area

$$\tilde{\lambda}^*_{new} = \frac{\tilde{\lambda}}{\max_i (\lambda_i - 1) + 1}.$$

In the numerical example here, I_L consists of 2 edges in polygons 3 and 4, and 3 edges in polygons 5 and 6, as marked in Figure 14b. The length of those edges in the reference design vary in the range between 0.2546 and 0.7963, and therefore we set the target length to $L^* = 0.3$.

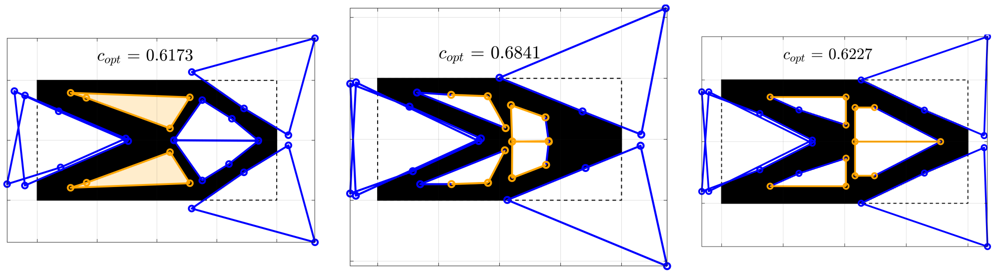
The optimized design is presented in Fig. 14b, where it is clear that design differs from the reference design and the optimized compliance value is increased by 13% and equals $c_{opt} = 0.6841$. The length constraints are mathematically satisfied with the lengths of all edges in I_L very close to L^*. However, some pairs of edges are almost aligned, resulting in effective length that is greater than L^*. This can be prevented for example by limiting the angles of the polygon, which are also an explicit function of the coordinates of the vertices [45].

4.3.3 Orientation constraint

The explicit parametrization of PP allows for geometrical constraints to also be imposed implicitly. For example, we can enforce edges to maintain their orientation along the optimization by properly averaging the values of the SA. Conveniently, this averaging is mathematically obtained by multiplying the SA with a fixed averaging matrix. For example, the orientation-constrained SA of compliance is

$$\left\{ \frac{\partial c}{\partial \mathbf{x}} \right\}_g = \frac{\partial c}{\partial \mathbf{x}} \mathbf{G},$$

where $\frac{\partial c}{\partial \mathbf{x}}$ is a row vector with derivatives of the compliance with respect to all design variables, $\mathbf{G}$ is the geometrical constraint matrix, and the subscript g indicates geometrically constrained derivatives. Figure 14c presents the optimized design when fixing the orientation of edges to be horizontal or vertical, for the same edges as in the length



(a) **(b)** **(c)**

Fig. 14. Optimized design with geometrical constraints. a) Minimal area constraint, $c_{opt} = 0.6173$. b) Fixed length constraint, $c_{opt} = 0.6841$. c) Fixed orientation constraint, $c_{opt} = 0.6227$.

constraint. Again, the 'geometrical constraints' are satisfied with some sacrifice in the performance relative to the reference design, as $c_{opt} = 0.623$.

5 Conclusions

In this study we introduced a new polygonal primitive for feature mapping-based topology optimization. The polygon can have any number of sides, and can be convex or non-convex. In the kernel of the proposed polygonal primitive lies a new method for computing an approximate signed distance function of arbitrary polygons as well as a regulation scheme that prevents self intersection of the polygons during the optimization. The signed distance function is obtained by performing Boolean operations on planes and it has a complexity of $\mathcal{O}(n)$ and is therefore very efficient numerically. Moreover, since most of the operations are pointwise, the computation of the signed distance function can be easily parallelized.

Over three benchmark problems we showed that the optimized designs with the proposed method generates designs that are very similar to optimized designs found using a classical density-based method, generally with some sacrifice in the performance due to the straight edges of structural features. In this regard, we note that because the designs are guaranteed to have only straight lines, the optimized designs can be easily implemented where such requirements exist, such as in reinforced concrete structures.

The idea of imposing geometrical constraints through the proposed polygonal primitive was also demonstrated. Specifically, we introduced area, length and orientation constraints that are straightforward and computationally inexpensive to compute, in contrast to traditional point-centered or skeleton primitives where such constraints are likely more challenging to enforce in a mathematically consistent manner. Thus, the explicit and rich shape parametrization of the polygon primitives allows us to naturally capture shape features and therefore facilitate imposition of a wide range of geometrical constraints.

The main purpose of the current study is to introduce the polygonal primitive and its advantages, whereas there are some open question that need to be addressed in future research. First, when using polygons with 7 or more sides, we observed that trimming of the shape may occur, which reduces the control over the projected shape and may lead to inferior designs. Using a different sequence of Boolean operations may resolve this issue, and is worth exploring. Additionally, adding topological design variables to the formulation will allow the optimization to eliminate primitives without collapsing them to 1D elements, which will likely improve the numerical performance. Finally, length scale control is not guaranteed with the proposed formulation, which in the authors opinion should be included in future studies.

Acknowledgments. This work was supported by the National Aeronautics and Space Administration (NASA) Space Technology Research Institute (STRI) for Ultra-Strong Composites by Computational Design (US-COMP) (Grant No. NNX17AJ32G). Any opinions, findings, and conclusions or recommendations expressed in this article are those of the author(s) and do not necessarily reflect the views of NASA.

Conflict of interests

The authors declare that they have no conflict of interest.

Replication of results

All equations and data needed for replication of the results is in the manuscript. Upon request, further information will be provided for academic use.

References

[1] Wein F, Dunning PD, Norato JA. A review on feature-mapping methods for structural optimization. Structural and multidisciplinary optimization. 2020;62:1597–1638.

[2] Cheng LT, Tsai YH. Redistancing by flow of time dependent eikonal equation. Journal of Computational Physics. 2008;227(8):4002–4017.

[3] Zhou Y, Zhang W, Zhu J, Xu Z. Feature-driven topology optimization method with signed distance function. Computer Methods in Applied Mechanics and Engineering. 2016;310:1–32.

[4] Norato J, Haber R, Tortorelli D, Bendsøe MP. A geometry projection method for shape optimization. International Journal for Numerical Methods in Engineering. 2004;60(14):2289–2312.

[5] Mei Y, Wang X, Cheng G. A feature-based topological optimization for structure design. Advances in Engineering Software. 2008;39(2):71–87.

[6] Qian Z, Ananthasuresh G. Optimal embedding of rigid objects in the topology design of structures. Mechanics Based Design of Structures and Machines. 2004;32(2):165–193.

[7] Chen J, Shapiro V, Suresh K, Tsukanov I. Shape optimization with topological changes and parametric control. International journal for numerical methods in engineering. 2007;71(3):313–346.

[8] Xia L, Zhu J, Zhang W, Breitkopf P. An implicit model for the integrated optimization of component layout and structure topology. Computer Methods in Applied Mechanics and Engineering. 2013;257:87–102.

[9] Zelickman Y, Guest JK. Construction aware optimization of concrete plate thicknesses. Engineering Structures. 2023;296:116889.

[10] Bell B, Norato J, Tortorelli D. A geometry projection method for continuum-based topology optimization of structures. In: 12th AIAA Aviation Technology, integration, and operations (ATIO) conference and 14th AIAA/ISSMO multidisciplinary analysis and optimization conference; 2012. p. 5485.

[11] Norato J, Bell B, Tortorelli DA. A geometry projection method for continuum-based topology optimization with discrete elements. Computer Methods in Applied Mechanics and Engineering. 2015;293:306–327.

[12] Guo X, Zhang W, Zhong W. Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. Journal of Applied Mechanics. 2014;81(8):081009.

[13] Zhang W, Yuan J, Zhang J, Guo X. A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model. Structural and Multidisciplinary Optimization. 2016;53:1243–1260.

[14] Guo X, Zhang W, Zhang J, Yuan J. Explicit structural topology optimization based on moving morphable components (MMC) with curved skeletons. Computer methods in applied mechanics and engineering. 2016;310:711–748.

[15] Zhang W, Song J, Zhou J, Du Z, Zhu Y, Sun Z, et al. Topology optimization with multiple materials via moving morphable component (MMC) method. International Journal for Numerical Methods in Engineering. 2018;113(11):1653–1675.

[16] Hoang VN, Jang GW. Topology optimization using moving morphable bars for versatile thickness control. Computer Methods in Applied Mechanics and Engineering. 2017;317:153–173.

[17] Guest JK. Imposing maximum length scale in topology optimization. Structural and Multidisciplinary Optimization. 2009;37:463–473.

[18] Greifenstein J, Letournel E, Stingl M, Wein F. Efficient spline design via feature-mapping for continuous fiber-reinforced structures. Structural and Multidisciplinary Optimization. 2023;66(5):99.

[19] Zhu B, Wang R, Zhang H, Li H, Liang J, Zhang X, et al. An approach for geometrically nonlinear topology optimization using moving wide-Bezier components with constrained ends. Journal of Mechanical Design. 2022;144(1):011704.

[20] Zhu B, Wang R, Wang N, Li H, Zhang X, Nishiwaki S. Explicit structural topology optimization using moving wide Bezier components with constrained ends. Structural and Multidisciplinary Optimization. 2021;64:53–70.

[21] Deng H, Hinnebusch S, To AC. Topology optimization design of stretchable metamaterials with Bézier skeleton explicit density (BSED) representation algorithm. Computer Methods in Applied Mechanics and Engineering. 2020;366:113093.

[22] Zhu B, Wang R, Liang J, Lai J, Zhang H, Li H, et al. Design of compliant mechanisms: An explicit topology optimization method using end-constrained spline curves with variable width. Mechanism and Machine Theory. 2022;171:104713.

[23] Zhou Y, Jiu L, Gao T, Zhu J, Zhang W. Worm-like highly deformable Multi-Section B-Spline Offset Feature for topology optimization. Computer Methods in Applied Mechanics and Engineering. 2023;412:116108.

[24] Bærentzen JA. Robust generation of signed distance fields from triangle meshes. In: Fourth International Workshop on Volume Graphics, 2005. IEEE; 2005. p. 167–239.

[25] Koch P, May S, Schmidpeter M, Kühn M, Pfitzner C, Merkl C, et al. Multi-robot localization and mapping based on signed distance functions. Journal of Intelligent & Robotic Systems. 2016;83:409–428.

[26] Oleynikova H, Millane A, Taylor Z, Galceran E, Nieto J, Siegwart R. Signed distance fields: A natural representation for both mapping and planning. In: RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics. University of Michigan; 2016. .

[27] Van Dijk NP, Maute K, Langelaar M, Van Keulen F. Level-set methods for structural topology optimization: a review. Structural and Multidisciplinary Optimization. 2013;48:437–472.

[28] Sethian JA. A fast marching level set method for monotonically advancing fronts. proceedings of the National Academy of Sciences. 1996;93(4):1591–1595.

[29] Peng D, Merriman B, Osher S, Zhao H, Kang M. A PDE-based fast local level set method. Journal of computational physics. 1999;155(2):410–438.

[30] Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S. Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019. p. 165–174.

[31] Sitzmann V, Chan E, Tucker R, Snavely N, Wetzstein G. Metasdf: Meta-learning signed distance functions. Advances in Neural Information Processing Systems. 2020;33:10136–10147.

[32] Xu S, Cai Y, Cheng G. Volume preserving nonlinear density filter based on heaviside functions. Structural and Multidisciplinary Optimization. 2010;41:495–505.

[33] Wang F, Lazarov BS, Sigmund O. On projection methods, convergence and robust formulations in topology optimization. Structural and multidisciplinary optimization. 2011;43:767–784.

[34] 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.

[35] Design sensitivity analysis: overview and review. Inverse problems in Engineering. 1994;1(1):71–105.

[36] Guest JK, Prévost 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.

[37] Guest JK. Topology optimization with multiple phase projection. Computer Methods in Applied Mechanics and Engineering. 2009;199(1-4):123–135.

[38] Li L, Khandelwal K. Volume preserving projection filters and continuation methods in topology optimization. Engineering Structures. 2015;85:144–161.

[39] Bendsøe MP, Kikuchi N. Generating optimal topologies in structural design using a homogenization method. Computer methods in applied mechanics and engineering. 1988;71(2):197–224.

[40] Bendsøe MP. Optimal shape design as a material distribution problem. Structural optimization. 1989;1:193–202.

[41] Mei G, Tipper JC, Xu N. Ear-clipping based algorithms of generating high-quality polygon triangulation. In: Proceedings of the 2012 International Conference on Information Technology and Software Engineering: Software Engineering & Digital Media Technology. Springer; 2013. p. 979–988.

[42] ElGindy H, Everett H, Toussaint G. Slicing an ear using prune-and-search. Pattern Recognition Letters. 1993;14(9):719–722.

[43] Le C, Norato J, Bruns T, Ha C, Tortorelli D. Stress-based topology optimization for continua. Structural and Multidisciplinary Optimization. 2010;41:605–620.

[44] Yang D, Liu H, Zhang W, Li S. Stress-constrained topology optimization based on maximum stress measures. Computers & Structures. 2018;198:23–39.

[45] Zelickman Y, Amir O. Optimization of post-tensioned concrete slabs for minimum cost. Engineering Structures. 2022;259:114132.