



Topology optimization with supershapes

Julián A. Norato¹

Received: 28 November 2017 / Revised: 7 May 2018 / Accepted: 14 June 2018 / Published online: 7 July 2018
© The Author(s) 2018

Abstract

This work presents a method for the continuum-based topology optimization of structures whereby the structure is represented by the union of supershapes. Supershapes are an extension of superellipses that can exhibit variable symmetry as well as asymmetry and that can describe through a single equation, the so-called superformula, a wide variety of shapes, including geometric primitives. As demonstrated by the author and his collaborators and by others in previous work, the availability of a feature-based description of the geometry opens the possibility to impose geometric constraints that are otherwise difficult to impose in density-based or level set-based approaches. Moreover, such description lends itself to direct translation to computer aided design systems. This work is an extension of the author's group previous work, where it was desired for the discrete geometric elements that describe the structure to have a fixed shape (but variable dimensions) in order to design structures made of stock material, such as bars and plates. The use of supershapes provides a more general geometry description that, using a single formulation, can render a structure made exclusively of the union of geometric primitives. It is also desirable to retain hallmark advantages of existing methods, namely the ability to employ a fixed grid for the analysis to circumvent re-meshing and the availability of sensitivities to use robust and efficient gradient-based optimization methods. The conduit between the geometric representation of the supershapes and the fixed analysis discretization is, as in previous work, a differentiable geometry projection that maps the supershapes parameters onto a density field. The proposed approach is demonstrated on classical problems of 2-dimensional compliance-based topology optimization.

Keywords Topology optimization · Geometry projection · Supershapes · Superformula

1 Introduction

Density-based and level set topology optimization methods have been employed with wide success to produce organic, free-form designs across multiple application realms. Despite this success, a withstanding challenge in topology optimization is the ability to impose geometric considerations to render designs that are amenable to production. As most industry practitioners of topology optimization can attest to, a design obtained via topology optimization is rarely produced as-is. Consequently, following the topology optimization, design engineers create designs

that try to capture as much as possible the material distribution of the optimal topology, but incorporate necessary aspects of the production process. This 'interpreted' design often incurs in a significant detriment of the structural performance. As a result, the engineer must spend a significant amount of time performing design changes by trial and error, which is not only a resource-intensive task, but one that leads to a suboptimal design. This is arguably the primary reason why topology optimization has not yet earned a permanent place in engineering workflows, despite the wide availability of commercial software.

One aspect that has been investigated in topology optimization (and is a focus of ongoing work) with regards to the physical realization of topology optimization designs is the incorporation of various 'manufacturing constraints.' Their purpose is to render designs that conform to certain aspects of manufacturing processes. For example, one such widely used constraint is the so-called 'casting draw' constraint, which renders designs that do not exhibit undercuts and thus require no cores for mold casting. The reader is referred to the survey in Liu and Ma (2016)

Responsible Editor: Hyunsun Alicia Kim

This work was supported by the National Science Foundation, CMMI-1634563

✉ Julián A. Norato
julian.norato@uconn.edu

¹ The University of Connecticut, 191 Auditorium Road, U-3139, Storrs, CT 06269, USA

for a review of manufacturing considerations in topology optimization.

In addition to ease of manufacturing, there are other production considerations that necessitate consideration of geometric requirements. These include, for example, the availability of high-level geometric features that facilitate positioning in fixtures, assembly and inspection. In many existing workflows for design of mechanical components (that do not use topology optimization), a widely used geometric representation that accommodates these requirements is the combination of geometric primitives via Boolean operations. Primitives have convenient datum entities (faces, edges, and vertices) that facilitate positioning in manufacturing fixtures and assembly. They also make it easier to specify dimensional tolerances, which is not only useful to perform tolerance stackup analysis but also for dimensional quality control and inspection. Thus, even if an organic design obtained via topology optimization can be manufactured by a given process, it still may be more convenient to replace it with a design made of primitives that accommodates the other aforementioned requirements. It should be noted that, in fact, most computer aided design systems for solid modeling work primarily by representing designs via primitives.

Despite the effectiveness of the aforementioned manufacturing constraints to enforce certain geometric requirements, it remains difficult to introduce general geometric constraints in the topology optimization. At the root of this difficulty lies the way in which density-based and level set-based topology optimization techniques represent the structure, namely by means of a density field or the level set of a function, respectively. These representations naturally and easily accommodate topological and shape changes, leading to organic designs. However, they make it difficult to impose geometric considerations such as, for example, obtaining a design with convenient datum entities for ease of production. Therefore, it is desirable to have topology optimization techniques that are endowed with a feature-based geometric representation of the structure that allows to impose these kinds of considerations, while retaining the advantage of existing techniques of employing a fixed grid for the analysis and optimization to circumvent remeshing.

Methods to incorporate geometric features with analytical geometry descriptions in topology optimization generally fall within two categories. First, there is a group of methods that, through various means, combine free-form topology optimization with embedded components or holes shaped as geometric primitives (cf. Chen et al. 2007; Qian and Ananthasuresh 2004; Xia et al. 2012; Zhang et al. 2011, 2015; Zhou and Wang 2013; Zhu et al. 2008). The second group, to which this work belongs, includes methods that represent the structure solely through a feature-based representation. The precursor of these methods is, in fact,

one of the first topology optimization techniques: the bubble method (Eschenauer et al. 1994), where splines are used to represent the structural boundaries. This method requires re-meshing upon design changes, which limits its use. More recently, novel topology optimization techniques have been advanced that, as their density-based and level set-based counterparts, employ a fixed grid for the primal and sensitivity analyses, thereby circumventing the need for remeshing. Table 1 provides a comparison of these methods. The table includes only methods where the geometry is entirely composed of geometric primitives (to describe either the solid or the void regions). The geometric shapes listed for each method are the ones reported in the cited works; this does not imply these methods cannot potentially accommodate other geometric representations. Moreover, the analysis technique reported in the table is the one demonstrated in the references cited, however it can be argued that sharp interface methods (such as the extended finite element method) and ersatz material methods can be used interchangeably.

Among the methods listed in Table 1, two methods represent the structure as the union of primitive-shaped geometric components with fixed shape but variable dimensions and position: the moving morphable components method (Guo et al. 2014a, 2016; Zhang et al. 2016b; 2017b), and the geometry projection method (Bell et al. 2012; Norato et al. 2015; Zhang et al. 2016a). The former method performs the topology optimization of a) 2d-structures, by employing as geometric components rectangles, approximated via superellipses, and quadrilateral shapes with two opposite sides described by polynomial and trigonometric curves; and b) 3d-structures, by using cuboids. The latter method uses rectangles with straight or semicircular ends for the design of 2d-structures, and plates with semicylindrical edges for the design of 3d-structures. The use of fixed-shape components facilitates the design of structures made of stock material.

This work extends the geometry projection method to incorporate supershapes as a more flexible geometric representation to accommodate via a single formulation both designs of similar complexity to those obtained with free-form topology optimization, as well as designs made exclusively of primitives. While the designs presented in this work are still far from manufacturable as they do not incorporate many important geometric considerations, ease of production is the driving motivation, and our method aims to be a step in that direction by advancing the ability to represent primitive-only designs with a unified formulation.

The remainder of the paper is organized as follows. In Section 2, we introduce the supershapes. Section 3 describes the projection of the supershapes onto a density field for the analysis. The computation of the signed distance to

Table 1 Existing methods for topology optimization with feature-based geometries. Abbreviations: topological derivative (TD), finite element analysis (FEA), extended finite element method (XFEM), topology optimization (TO), radial basis function (RBF) and non-uniform rational B-splines (NURBS)

Method, Year	Description of Structure	Design variables	Combination of shapes	Connection geometry-analysis	Topological changes	Analysis
Bubble method, 1994 (Eschenauer et al. 1994)	All boundaries are splines	Control points	None (splines do not intersect)	Remeshing	Hole insertion via indicator function	Standard FEA on conforming mesh
Feature-based struct. opt., 2006 (Cheng et al. 2006), 2008 (Chen et al. 2008)	Solid envelope with implicitly defined circular, rectangular and triangular holes	Scale, position and orientation of holes	Composition via R-functions	Modulus is function of smooth Heaviside of implicit function	Hole merging via R-composition, hole insertion via TD	Standard FEA on fixed grid
Isogeometric TO with trimmed splines, 2010 (Seo et al. 2010)	All boundaries are splines	Control points	Uses 'trimmed surface analysis' to merge curves	Through isogeometric analysis	Hole merging via spline trimming, hole insertion via TD	Isogeometric analysis
Material mask overlay, 2011 (Saxena 2011)	Solid envelope with circular, elliptical and rectangular holes	Location, orientation and size of holes	Through cell membership test	Cells removed from analysis if inside of holes	Holes randomly added and removed via zero-order opt.	Hexagonal elements + mesh smoothing
Material mask overlay with smooth projection, 2012 (Wang et al. 2012)	Solid envelope with circular holes	Locations and radii of holes	Product of smooth Heavisides	Elemental density as a function of Heavisides (ersatz material)	Holes merge through Heavisides product; no hole insertion	Standard FEA on fixed grid
Geometry projection method, 2012 (Bell et al. 2012), 2015 (Norato et al. 2015), 2016 (Zhang et al. 2016a)	Union of rectangular bars with straight (Bell et al. 2012) or semicircular (Norato et al. 2015) ends, 3-d plates 2016 (Zhang et al. 2016a)	Position, orientation, and penalized size variable for each geometric element	Union through p -norm	Projection to density field	Elements merge through aggregation function, and can be removed through size variable; no insertion	Standard FEA on fixed grid
Level set TO with geom. constraints, 2014 (Liu et al. 2014)	Solid envelope with primitive-shaped holes approximated via RBFs	Coefficients of RBFs	Composition via R-functions	XFEM to follow level set	Hole merging via composition; no hole insertion	XFEM
Moving morphable components (MMC), 2014 (Guo et al. 2014b), 2016a (Zhang et al. 2016b), 2016b (Guo et al. 2016), 2017 (Zhang et al. 2017b)	Union of rectangles approximated as superellipses (Guo et al. 2014b), or curved quadrilaterals with polynomial or trigonometric sides (Zhang et al. 2016b; Guo et al. 2016)	Positions and orientation of discrete elements, dimensions (Guo et al. 2014b) and curve parameters (Zhang et al. 2016b; Guo et al. 2016)	Maximum of Heaviside function of individual geometric components	XFEM to follow level set (Guo et al. 2014b; Zhang et al. 2016b), ersatz material from average of nodal smooth Heaviside (Guo et al. 2016)	Through union of Heavisides (bars merge, separate and are engulfed in other bars); no component insertion	XFEM (Guo et al. 2014b; Zhang et al. 2016b) and standard FEA on fixed grid (Guo et al. 2016)
ISOCOMP, 2015 (Lin et al. 2015)	Solid structure with holes described by NURBS	Control points	Holes do not merge	Ersatz material from NURBS interpolation	Hole insertion via TD; update via simulated annealing	Isogeometric analysis
MMC with thickness control, 2017 (Hoang and Jang 2016)	Union of rectangular bars with semicircular ends	Positions of endpoints of bars medial axes	Product of regularized Heavisides	Geometry projection (ersatz material)	Bars merge through product of Heavisides; no bar insertion	Standard FEA on fixed grid

the boundary of a supershape, which is required for the geometry projection, is described in Section 2. Sections 5 and 6 describe the optimization problem and details of the computer implementation respectively. Several examples that demonstrate the proposed method are presented in Section 7. Finally, Section 8 draws conclusions of this work.

2 Supershapes

Supershapes (also called Gielis curves, Gielis 2017) are a generalization of superellipses that can exhibit variable symmetry as well as asymmetry and can describe through a single equation, the so-called superformula, a wide variety of shapes, including geometric primitives (Gielis 2003). The appeal of supershapes is their ability to express through a single equation a wide range of primitives and organic shapes. The superformula is given by

$$r(\theta) := \left[\left(\left| \frac{1}{a} \cos \left(\frac{m\theta}{4} \right) \right| \right)^{n_2} + \left(\left| \frac{1}{b} \sin \left(\frac{m\theta}{4} \right) \right| \right)^{n_3} \right]^{-\frac{1}{n_1}} \tag{1}$$

Figure 1 shows some examples of supershapes. When $m = 4$ and $n_1 = n_2 = n_3 = n$, this expression reduces to the equation for a superellipse (also known as a Lamé oval, Grigeman 1970). Unlike superellipses, supershapes need not be symmetric; the parameter m controls the rotational symmetry. The values of a and b control the size, and the exponents n_1, n_2 and n_3 control the curvature of the sides. The superformula can produce a wide range of shapes, including many shapes found in nature; the reader is referred to Gielis (2003; 2017) for more examples.

For the purpose of this work, two modifications are made to the superformula of (1). First, the absolute value function is not differentiable at zero, which would preclude the use of efficient gradient-based optimization methods. Therefore,

we replace it with the approximation

$$|x| \approx \sqrt{\epsilon + x^2}, \quad 0 < \epsilon \ll 1 \tag{2}$$

Second, use of the power reduction formulas $2 \cos^2 x = 1 + \cos 2x$ and $2 \sin^2 x = 1 - \cos 2x$ provides a simplification both for computation and for the sensitivity analysis of the square trigonometric functions that result after replacing the absolute value with the above approximation. The modified superformula is given by

$$\tilde{r}(\theta) := \left[\tilde{x}(\theta)^{\frac{n_2}{2}} + \tilde{y}(\theta)^{\frac{n_3}{2}} \right]^{-\frac{1}{n_1}} \tag{3}$$

where

$$\tilde{x}(\theta) := \left(\epsilon + \frac{1}{2a} \left[1 + \cos \left(\frac{m\theta}{2} \right) \right] \right) \tag{4}$$

$$\tilde{y}(\theta) := \left(\epsilon + \frac{1}{2b} \left[1 - \cos \left(\frac{m\theta}{2} \right) \right] \right) \tag{5}$$

A Mathematica script to interactively plot a supershape with the foregoing expressions while modifying its parameters is presented in Appendix A. An important note about (4) and (5) is that the angle θ must be in the range $[-\pi, \pi]$, as otherwise the supershape will not be closed. Therefore, if an angle is passed to these formulas, it must be appropriately wrapped around this range.

In addition to the foregoing parameters that define the size and shape of the supershape, we also wish to control its scale, location and orientation in space. Therefore, we introduce a scale factor s , the position \mathbf{x}_c of the origin of the coordinate system $\mathbf{e}'_1 - \mathbf{e}'_2$ on which (3) is applied, and the rotation $\phi = \mathbf{e}'_1 \cdot \mathbf{e}_1$ of this coordinate system with respect to a global coordinate system $\mathbf{e}_1 - \mathbf{e}_2$ as additional design parameters, as shown in Fig. 2. We note that θ in (3) is

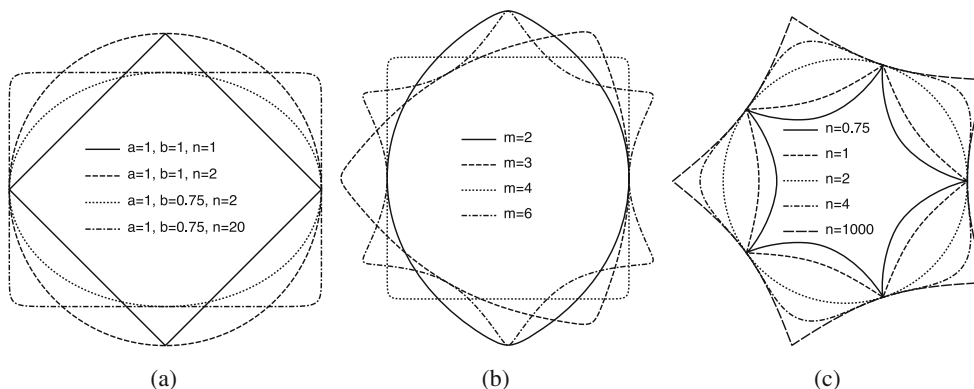


Fig. 1 Examples of supershapes. **a** From top to bottom: subellipse, circle, ellipse and superellipse (all shapes with $m = 4, n_1 = n_2 = n_3 = n$). **b** Shapes with different rotational symmetry (all shapes with

$a = b = 1, n_1 = n_2 = n_3 = 30$). **c** Shapes with different side curvature (all shapes with $m = 5, a = b = 1, n_1 = n_2 = n_3 = n$)

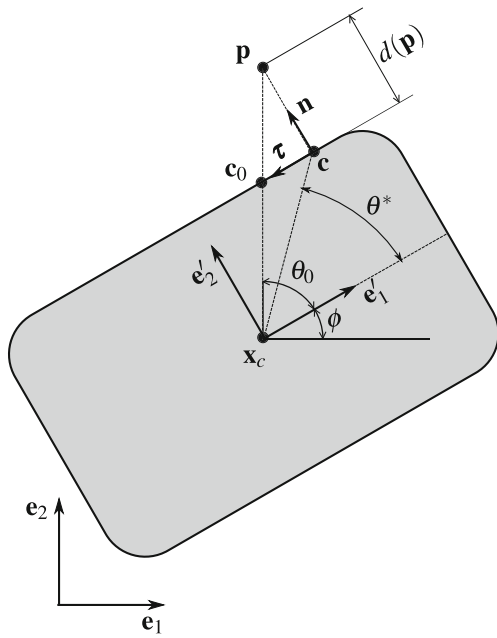


Fig. 2 Location and orientation of supershape

defined with respect to \mathbf{e}'_1 . With this notation, a point on the boundary of the shape is given by

$$\mathbf{x}(\theta, \phi, \mathbf{x}_c, m, a, b, n_1, n_2, n_3) = \mathbf{x}_c + s\tilde{r}(\theta)\mathbf{R}_\phi \begin{Bmatrix} \cos \theta \\ \sin \theta \end{Bmatrix} \quad (6)$$

$$= \mathbf{x}_c + s\tilde{r}(\theta)\mathbf{R}_{\phi+\theta}\mathbf{e}_1 \quad (7)$$

where

$$\mathbf{R}_t := \begin{bmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{bmatrix} \quad (8)$$

3 Geometry projection

To perform the analysis on a fixed grid, we use an ersatz material, whereby the elasticity tensor at a point is modified as a function of a density variable to reflect a fully or partially solid material, or void. This strategy is the same used by all density-based and many level set-based topology optimization methods. The design variables in our method, however, are the supershape parameters described in the preceding section. Therefore, we need a mapping between the supershape parameters and the aforementioned density. This mapping must be differentiable so that we can obtain design sensitivities with respect to the supershape parameters and employ efficient gradient-based optimization methods. To this end, we use the geometry projection method (Norato et al. 2004; Bell et al. 2012; Norato et al. 2015; Zhang et al. 2016a). The idea of the geometry projection is simple: the density at a point \mathbf{p} in

space is the fraction of the volume of the ball \mathbf{B}_p^r of radius r centered at \mathbf{p} that intersects the solid geometry, i.e.:

$$\rho(\mathbf{p}, r) := \frac{|\mathbf{B}_p^r \cap \omega|}{|\mathbf{B}_p^r|}, \quad (9)$$

where ω denotes the region of space occupied by the structure and $|\cdot|$ is a measure of the volume (or area). If we make the assumption that r is small enough that the portion of the solid boundary that intersects the ball, $\partial\omega \cap \mathbf{B}_p^r$, can be approximated by a straight line, then the density of (9) can be readily computed as a function of the signed distance d between \mathbf{p} and $\partial\omega$. As depicted in Fig. 3, in 2-d the density under this simplifying assumption corresponds to the fraction of the solid circular segment area to the area of the ball \mathbf{B}_p^r , giving

$$\rho(d(\mathbf{p}, \mathbf{z}), r) := \begin{cases} 0 & \text{if } d > r \\ \frac{1}{\pi r^2} \left[r^2 \arccos\left(\frac{d}{r}\right) - d\sqrt{r^2 - d^2} \right] & \text{if } -r \leq d \leq r \\ 1 & \text{if } d < -r, \end{cases} \quad (10)$$

where the arguments of ϕ have been omitted on the right-hand side of the equation for conciseness. Since r is fixed in the proposed method, it will be omitted as an argument hereafter. The vector of design parameters \mathbf{z} for the entire structure is the aggregation of the vectors \mathbf{z}_i of design parameters for each of the supershapes, i.e. $\mathbf{z} = [\mathbf{z}_1^T \ \mathbf{z}_2^T \ \dots \ \mathbf{z}_{N_s}^T]^T$, where N_s is the number of supershapes. The signed distance d is positive if \mathbf{p} is outside the shape, zero if \mathbf{p} lies on its boundary, and negative if it lies inside. The computation of the signed distance is discussed in Section 4.

In addition to the design parameters that dictate the shape, scale, position and orientation of a supershape, we also ascribe a size variable α to each supershape. A unique feature of the method presented in Bell et al. (2012), Norato et al. (2015), and Zhang et al. (2016a) is that in the ersatz material this variable is penalized as in density-based

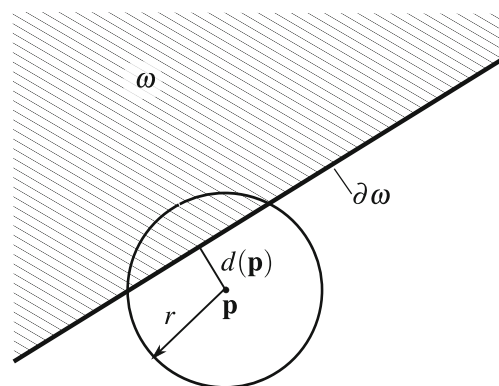


Fig. 3 Geometry projection

topology optimization. Therefore, the elasticity tensor at a point \mathbf{p} is given by

$$\mathbb{C}(\mathbf{p}, \mathbf{z}) := \hat{\rho}(\mathbf{p}, \mathbf{z})\mathbb{C}_0 \tag{11}$$

$$\hat{\rho}(\mathbf{p}, \mathbf{z}) := \alpha^q \rho(d(\mathbf{p}, \mathbf{z})) \tag{12}$$

where \mathbb{C} is the ersatz elasticity tensor, \mathbb{C}_0 is the elasticity tensor of the fully solid material, $\hat{\rho}$ is an *effective density* and q is a penalization power. The size variable α_i of supershape i is part of its vector of design variables, i.e.,

$$\mathbf{z}_i := [m_i, a_i, b_i, n_{1i}, n_{2i}, n_{3i}, \mathbf{x}_c^T, \phi_i, s_i, \alpha_i]^T \tag{13}$$

This penalized size variable is an important ingredient of our formulation, as it allows the optimizer to entirely remove a geometric component from the design when the size variable becomes zero.

Equation (10) is used for the calculation of the density with respect to a single geometric component. More generally, we consider a structure made by the union of multiple shapes. Here, this union is made through the maximum function, which corresponds to the Boolean union of implicit functions (Shapiro 2007). Specifically, at a point \mathbf{p} , a *composite density* is defined as

$$\tilde{\rho}(\mathbf{p}, \mathbf{z}) := \max_i \hat{\rho}_i(\mathbf{p}, \mathbf{z}_i) \tag{14}$$

where $\hat{\rho}_i$ and \mathbf{z}_i denote the effective density and design parameters for supershape i respectively.

Since the maximum function is not differentiable, we replace it with a lower-bound Kreisselmeier-Steinhauser (KS) approximation (as in Zhang et al. 2017a):

$$\max_i \hat{\rho}_i(\mathbf{p}, \mathbf{z}_i) \approx KS_{\max}(\hat{\rho}_i(\mathbf{p}, \mathbf{z}_i)) := \frac{1}{k} \ln \left(\frac{1}{N_s} \sum_i^{N_s} e^{k\hat{\rho}_i(\mathbf{p}, \mathbf{z}_i)} \right) \tag{15}$$

where k is a specified parameter and N_s is the total number of shapes. Besides other benefits (described in Zhang et al. 2017a), this approximation has the advantage that it approximates the true maximum from below. This helps the method produce designs with less gaps between shapes in the optimal designs. These gaps occur because points in the gap regions obtain some stiffness from the nearby shapes through the geometry projection and thus have a stiffness that is appreciably larger than that of void regions away from the shapes. This phenomenon is exacerbated by the use of other aggregation functions (such as the p -norm) that approximate the maximum from above, as they consequently increase the effective stiffness of these gap regions. In our numerical experiments, the lower-bound KS function does a much better job of rendering designs without gaps.

We note that if every $\hat{\rho}_i$ for every supershape i is zero at a point \mathbf{p} , the composite density will be consequently zero and the analysis will be ill-posed. Therefore, as is customary

in ersatz material methods, a small lower bound ρ_{\min} must be imposed on the composite density. We achieve this via

$$\tilde{\rho}(\mathbf{p}, \mathbf{z}) := \rho_{\min} + (1 - \rho_{\min})KS_{\max}(\hat{\rho}_i(\mathbf{p}, \mathbf{z}_i)) \tag{16}$$

where $0 < \rho_{\min} \ll 1$.

Finally, the ersatz material is modeled using the composite density as

$$\mathbb{C}(\mathbf{p}, \mathbf{z}) := \tilde{\rho}(\mathbf{p}, \mathbf{z})\mathbb{C}_0 \tag{17}$$

In our method, we use the finite element method to solve the analysis problem. A composite density is computed at each finite element, with \mathbf{p} above corresponding to the element centroid.

4 Signed distance calculation

Even though supershapes are given in polar coordinates, the computation of the signed distance is not as straightforward as in the case of the geometric representations used in our previous works, whereby the discrete geometric components are determined by a medial axis (in the case of approximately cylindrical bars, Norato et al. 2015) or a medial surface (in the case of plates, Zhang et al. 2016a). In these geometric representations, which in effect are implicit, distance-constrained offset surfaces (Bloomenthal 1990), the signed distance can be readily computed in closed form as the distance to the medial axis (or surface) minus the offset (e.g., the half-width of bars or half-thickness of plates). Due to the form of (1), however, it is not possible to obtain a closed-form expression for the signed distance from any point to the supershape boundary, and therefore it must be obtained numerically. It is worth noting that (6) constitutes an *explicit* representation, since this expression constitutes a rule to generate points on the boundary of the supershape, as opposed to implicit representations, whereby a rule determines whether a point is outside or inside the solid (Shapiro 2002).

Given a supershape with parameters \mathbf{z}_i , the distance from a point \mathbf{p} to its boundary is given by

$$|d(\mathbf{p}, \mathbf{z}_i)| := \|\mathbf{p} - \mathbf{c}(\mathbf{z}_i)\| \tag{18}$$

where \mathbf{c} , the point on the boundary of the supershape closest to \mathbf{p} (cf. Fig. 2), is given from (6) by

$$\mathbf{c}(\mathbf{z}_i) := \mathbf{x}(\theta^*(\mathbf{z}_i), \mathbf{z}_i) \tag{19}$$

where θ^* is the solution to the problem

$$\theta^*(\mathbf{z}_i) = \arg \min_{\theta} f(\theta) := \|\mathbf{p} - \mathbf{x}(\theta, \mathbf{z}_i)\|_2^2 \tag{20}$$

The foregoing problem can be solved numerically using root-finding methods to solve the first-order optimality

condition $f'(\theta) = 0$. We employ a damped Newton method to solve this problem, with θ at iteration I given by:

$$\theta^{(I)} \leftarrow \theta^{(I-1)} - \beta^{(I)} \frac{f'(\theta)}{f''(\theta)} \tag{21}$$

where the damping parameter is given by

$$\beta^{(I)} = \begin{cases} \beta^{(0)} & \text{If } I = 1 \\ \min(1, 2\beta^{(I-1)}) & \text{If } I > 1 \text{ and } f^{(I)} < f^{(I-1)} \\ \max(\beta_{min}, \frac{1}{2}\beta^{(I-1)}) & \text{Otherwise} \end{cases} \tag{22}$$

The Newton update is repeated until $|f'|$ falls under a specified tolerance $restol$. The derivative f' can be analytically computed from 3) and (6) (cf. (48)). To compute f'' , we use forward finite differences. The Newton update can fail if $f''(\theta) = 0$, i.e. at an inflection point of the squared distance. However, we did not observe this situation in our numerical experiments. While a more sophisticated, robust and efficient solution strategy could be used to solve the shortest distance calculation (for example, a trust-region method), the above iteration is easy to vectorize for multithreading and works well in our examples.

A good initial guess θ_0 for the Newton iteration corresponds to the point \mathbf{c}_0 where the line between \mathbf{p} and \mathbf{x}_c intersects the boundary (shown in Fig. 2), which can be readily computed by

$$\theta_0 = \arctan\left(\frac{p_2 - x_{c2}}{p_1 - x_{c1}}\right) - \phi \tag{23}$$

A four-quadrant version of the arctan function must be employed and, as discussed in Section 2, the resulting angle must be wrapped to the interval $[-\pi, \pi]$.

The sign of the signed distance can be computed, for instance, as

$$\text{sgn}(d(\mathbf{p}, \mathbf{z}_i)) = \text{sgn}((\mathbf{p} - \mathbf{c}) \cdot (\mathbf{c} - \mathbf{x}_c)) \tag{24}$$

Since we compute one composite density per element, a signed distance calculation to the element centroid is required for every element in the mesh. This calculation is evidently more expensive than the one used in our previous works, where a closed-form expression in terms of the design parameters is available. This cost is alleviated by the fact that these calculations are embarrassingly parallel, hence we can use parallel computing to speed up the computation. The computational cost of one signed distance calculation for the entire mesh is $O(N_{el} \times N_s)$, where N_{el} is the number of elements in the mesh. We note that this cost is linear in N_{el} , whereas the cost of the finite element analysis is $O(N_{eq}^q)$, where $N_{eq} > N_{el}$ is the number of equations and $q > 1$. Therefore, as the number of elements grows, the finite element analysis dominates the cost of the optimization as usual. Finally, it is possible (but not done in this work) to use efficient spatial partitioning techniques

such as kD-trees to further reduce the cost of the signed distance calculation.

5 Optimization problem

In this work, we consider the classical topology optimization problem of minimizing compliance subject to a volume fraction constraint:

$$\begin{aligned} \min_{\mathbf{z}} C(\mathbf{u}(\mathbf{z})) &:= \int_{\Gamma_t} \mathbf{u}(\mathbf{z}) \cdot \mathbf{t} ds \\ \text{subject to} & \\ \frac{V(\mathbf{z})}{|\Omega|} - v_f^* &\leq 0 \\ \mathbf{a}(\mathbf{u}(\mathbf{z}), \mathbf{v}) &= \mathbf{l}(\mathbf{v}), \quad \forall \mathbf{v} \in \mathcal{U}_\Omega; \mathbf{u} \in \mathcal{U}_\Omega \\ z_{low} \leq z_j &\leq z_{upp} \quad j = 1, \dots, 10N_s \end{aligned} \tag{25}$$

In the above problem, C denotes the structural compliance. While we consider compliance minimization for simplicity in this work, we note that stress considerations can be incorporated in geometry projection methods (Zhang et al. 2017a). The set Ω denotes the design envelope, i.e. the region of space where material can be distributed. We assume the tractions \mathbf{t} are design-independent, and they are applied on a fixed portion of $\partial\Omega$, which we denote by Γ_t . The volumes of the structure and the design envelope are denoted by V and $|\Omega|$ respectively, and v_f^* is the constraint limit on the volume fraction. The volume of the structure is computed via

$$V(\mathbf{z}) = \int_{\Omega} \bar{\rho}(\mathbf{p}, \mathbf{z}) dv \tag{26}$$

Equilibrium of the structure is enforced for any design produced by the optimizer via the second constraint in (25), where \mathbf{a} and \mathbf{l} are the energy bilinear and load linear forms respectively, given by

$$\mathbf{a}(\mathbf{u}, \mathbf{v}) := \int_{\Omega} \nabla \mathbf{v} \cdot \mathbb{C}(\mathbf{p}, \mathbf{z}) \mathbf{u} dv \tag{27}$$

$$\mathbf{l}(\mathbf{v}) := \int_{\Gamma_t} \mathbf{v} \cdot \mathbf{t} da \tag{28}$$

and $\mathcal{U}_\Omega := \{\mathbf{u} \in H^1(\Omega) : \mathbf{u} = \mathbf{0} \text{ on } \Gamma_u\}$ is the set of admissible displacements. The portion of $\partial\Omega$ on which homogeneous displacement boundary conditions are applied is denoted by Γ_u , and we assume this boundary to also be design-independent. The effective elasticity tensor \mathbb{C} in (27) is computed from (17). Finally, in the above we have also assumed that there are no body loads applied on the structure. An important note is due here: for the SIMP penalization on the supershape size variables to be effective, the elasticity tensor \mathbb{C} in (27) must be computed using a value of q in (12) larger than the value used to compute the volume in (26). Here, we use $q = 3$ for the former and $q = 1$ for the latter.

The last line in the problem of (25) corresponds to bounds on the design variables. The purpose of these bounds is discussed in detail in the following section. Since the problem is highly nonlinear, we impose move limits on the design variables so that only limited design steps are taken by the optimizer at each iteration, thus avoiding early overshooting to a poor design from which the optimization cannot recover. Since different variables represent different quantities and have different bound ranges, it is easier to impose move limits if we first scale all of the design variables to the range $[0, 1]$, for example via $\hat{z}_i := (z_i - z_{low}) / (z_{upp} - z_{low})$, and then impose the same move limit on all of the variables. For a move limit $0 < M \leq 1$, we then replace the last line in (25) at iteration I with

$$\max(0, \hat{z}_i^{(I-1)} - M) \leq \hat{z}_i^{(I)} \leq \min(1, \hat{z}_i^{(I-1)} + M) \quad (29)$$

The sensitivities are accordingly scaled by $\partial \hat{z}_i / \partial z_i = 1 / (z_{upp} - z_{low})$. Appendix B summarizes the sensitivities of the functions in (25).

6 Computer implementation

We start the description of our computer implementation by providing an algorithm for our method, shown in Fig. 4. Our method is implemented in MATLAB. The finite element analysis is performed on a regular grid of N_{el} square bilinear elements of size h . Each of the outer *repeat* iterations corresponds to a design update in the optimization, which is executed until the norm of the Karush-Kuhn-Tucker (KKT) condition or the relative change in the compliance between consecutive iterations fall below specified tolerances.

6.1 Signed distance and projected density calculation

We compute one projected density (10) per element, for which we compute the signed distance from the centroid of the element to each of the supershapes. This calculation corresponds to the first double *for* loop inside the *repeat* loop. This distance is computed by using the damped Newton iteration described in Section 4. The step size of the finite difference approximation of f'' (20) is $1\text{E-}6$. The initial damping parameter is $\beta^{(0)} = 1\text{E-}3$, and $\beta_{min} = 1\text{E-}6$. The stopping criterion is $restol = 1\text{E-}8$.

By using N_{el} -vector representations of the angles in (20) and (23) and of the distance in (18) and (20), we take advantage of MATLAB's automatic multithreading of vector operations to parallelize the inner *for* loop. While this incurs in some redundant computation for elements for which the iteration converges faster than others, this strategy is still much more efficient than the alternative sequential solution. The computation of the sign of (24) is similarly

Algorithm Topology Optimization with Supershapes

```

k ← 0                                ▷ Iteration counter
ẑ(0) ← ẑ0                            ▷ Initial design
Cold = -objtol
repeat
  for i = 1, ..., Ns do
    for j = 1, ..., Nel do
      Compute signed distance dji to shape i          ▷ §4
      Compute projected density ρji                    ▷ Eq. (10)
    end for
  end for
  for j = 1, ..., Nel do
    Compute element stiffness matrix Kj using
      C(z) of Eq. (17)
    Assemble Kj into global stiffness matrix K
  end for
  Solve K(z)u(z) = f for u(z)
  Compute C(u(z)), ∇C(u(z))                            ▷ Eq. (25) and App. B
  Compute V(z), ∇V(z)                                    ▷ Eq. (26) and App. B
  Impose move limits and update ẑlow and ẑupp          ▷ Eq. (29)
  ẑ(k+1) ← opt(ẑ(k), C, ∇C, V, ∇V, ẑlow, ẑupp)        ▷ Update design
  Compute norm of KKT condition ||kkt||
  k ← k + 1
  Compute relative change in objective Δc = |(C - Cold) / Cold|
  Cold ← C
until ||kkt|| ≤ kkttol or Δc ≤ objtol

```

Fig. 4 Algorithm for topology optimization with supershapes

vectorized. We note that while in our implementation we use multithreading, the signed distance calculation is embarrassingly parallel and can be readily implemented in distributed memory programming.

For the calculation of the projected density, we employ a sample window radius of $r = 2.5h$ in (10). In previous implementations of the geometry projection method we have used $r = \sqrt{2}h/2$, i.e., the sampling window circumscribes the element. However, in the numerical experiments performed in this work we found that a larger radius promotes faster convergence, particularly in early iterations. The calculation of the projected density of (10) is also done using efficient vector operations. An array of projected densities per element and per supershape is stored for subsequent calculations.

6.2 Finite element primal and sensitivity analyses

With the projected density for each element/supershape available, we proceed to solve the finite element analysis (corresponding to the second constraint in (25)). To compute the stiffness matrix \mathbf{K}_j of element j , we compute the ersatz elasticity tensor of (17), which requires the calculation of the composite density $\tilde{\rho}$ of (16). As mentioned before, we use $q = 3$ in the calculation of the effective density of (12) to penalize intermediate values of the size variable α_i of supershape i .

The lower bound ρ_{min} in (16) is set to $1\text{E-}4$ in all presented results. Lower values increase the ill-conditioning of the stiffness matrix, and thus decrease the accuracy of the primal

solution and, consequently, of the sensitivities. As ρ_{\min} decreases, the disagreement between the analytical sensitivities of the compliance and those approximated via finite differences increases. Inaccurate sensitivities hinder the convergence and efficacy of the gradient-based optimization. On the other hand, too high a ρ_{\min} leads to unrealistically stiff void regions. The foregoing value seems to consistently work well in our experiments, and is used throughout this work.

Finally, we use a value of $k = 32$ for the calculation of the lower KS function of (16).

The solution \mathbf{u} to the primal analysis is used to compute the compliance C of (25). The computation of the volume of (26), on the other hand, is performed with $q = 1$ in (12). Finally, sensitivities of the compliance and volume fraction are computed following the expressions in Appendix B.

6.3 Optimization

Before taking the design step, we impose move limits on the design by modifying the lower and upper bounds on the scaled design variables as per (29). To perform the design update, we employ the method of moving asymptotes (MMA) (Svanberg 1987). Specifically, we use the MATLAB implementation corresponding to the version of MMA presented in Svanberg (2007). All MMA parameters are set to the default values suggested in Svanberg (2007) for the standard problem formulation, namely $a_0 = 1$, and $a_l = 0$, $c_l = 1,000$ and $d_l = 1$ for every constraint l in the optimization. The asymptote increase and decrease factors are set to the default values of 1.2 and 0.7 respectively. The reader is referred to Svanberg (2007) for a detailed explanation of these terms, and here we provide them solely for the purpose of reproducibility of our results.

6.4 Design variables bounds

An important aspect of our method that we discuss in this section is the need for appropriate bounds on the various design parameters. This need arises from considerations on 1) accuracy of the geometry projection, 2) robustness of the sensitivities calculation, and 3) physical realization.

With regards to accuracy of the geometry projection, we recall that the definition of the projected density of (10) rests on the assumption that the portion of the supershape boundary intersecting the sample window, $\partial\omega \cap \mathbf{B}_p^r$, can be approximated as a single straight line. Therefore, we want to avoid shape boundaries that are too curvy in relation to the element size, as well as thin slivers that would lead to disjoint intersections. Loosely speaking, we want the element size h to be small in relation to the supershape. This is much easier to control in the offset surfaces reported in our previous work, as they accommodate (by design) very little shape variation.

To control the supershape proportions in relation to the element size h , we impose various bounds on its parameters. We set lower bounds on a and b to avoid shapes that resemble thin slivers. We also impose upper bounds on these parameters, since slivers can also develop for extreme a/b ratios if, for example, the exponents n_1 , n_2 and n_3 are such that convex sides form. With the same motivation, we impose lower bounds on these exponents to avoid highly convex sides, and, in some examples, we also set $n_1 = n_2 = n_3$. In general, we aim to attain an adequate range for the aspect ratio of the shape. We note that it is difficult to strictly enforce the requirement that the intersection between the boundary of a shape and an element is approximately straight, since supershapes can develop corners. Nevertheless, this did not seem to be a problem in our numerical experiments given the applied bounds.

The second consideration as to why parameter bounds are needed pertains the calculation of sensitivities. As discussed in our previous work (cf., Norato et al. 2015; Zhang et al. 2016a), when the closest point on the boundary of the supershape is not unique, the sensitivities are not defined. In the case of offset surfaces, this occurs when the signed distance is evaluated at a point lying on the medial axis (in 2-d) or medial surface (in 3-d), and the sample window size is equal or greater to the bar width or plate thickness. In that case, it is easy to prevent this situation by requiring that the sample window size (and hence the element size) are sufficiently small with regards to these dimensions. In the case of supershapes, on the other hand, it is more difficult to prevent this situation, given the wide variation of shapes that can be accommodated. As before, controlling the aspect ratio of the supershape via bounds on the aforementioned parameters seems to effectively avoid this situation in practice.

Non-uniqueness of the signed distance can also arise from sharp corners in the supershapes. However, the substitution of the absolute value in the supershape formula with the approximation of (2) rounds the corners. We use $\epsilon = 1\text{E-}3$ in (2) for all of our examples. We also place upper bounds on the exponents n_1 , n_2 and n_3 to avoid increasing the non-linearity of the signed distance in the supershape parameters, and consequently of the optimization problem functions.

The third and final consideration to put bounds on the supershape parameters is simply physical realization. In particular, we want to avoid shapes so small that they could not be fabricated. The most natural parameters to control the shape size are a and b along with the scaling factor s . In general, however, enforcement of a strict minimum size in the structure is not straightforward.¹ Convex sides can render a supershape size smaller than a and b . Moreover,

¹The minimum size of a supershape can be defined as its minimum width. The width of a closed planar curve is the distance between two parallel supporting lines bounding the curve (Struik 2012).

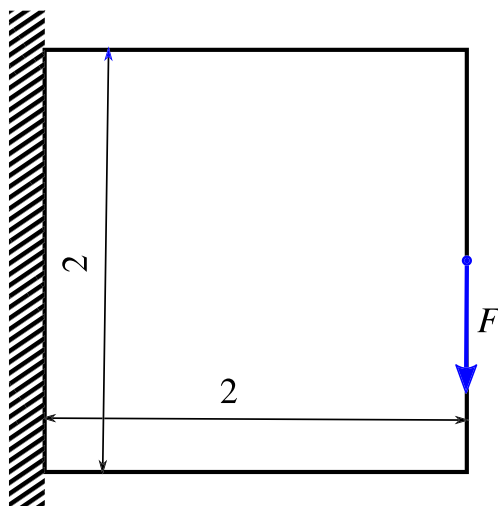


Fig. 5 Short cantilever beam problem

the intersection of supershapes can be smaller than the minimum size of the intersecting supershapes. A strategy to enforce a minimum size in geometry projection methods is described in Hoang and Jang (2016). We defer investigation of this issue to future work.

In summary, an important aspect of using supershapes, given the wide freedom of shapes they can represent, is the imposition of bounds on the parameters to ensure robust behavior of the proposed method. The values of the bounds presented in our examples seemed to consistently work well. However, a more in-depth understanding of the various interplays between supershape parameters would render a more systematic strategy to control the resulting supershapes. Such a study is deferred to future work.

7 Results

In this section we present numerical examples to demonstrate the proposed method. In all examples, we consider an

isotropic elastic material with Young’s modulus $E = 1E5$ and Poisson’s ratio of $\nu = 0.3$ (since minimum-compliance designs depend only on the volume fraction and not on the magnitude of the applied load or the material’s modulus, we henceforth omit units for brevity). The magnitude of the applied load in all examples is $F = 10$. All supershapes have a thickness of $t = 1$, and in all of the initial designs we assign $\alpha_q = 0.5$ to all the shapes. Unless noted, in all examples, we use a move limit of $m = 0.05$.

7.1 Short cantilever beam

We first design a short, cantilever beam (cf. Fig. 5) with a volume fraction constraint of $v_f^* = 0.5$. A 40×40 grid is used for the analysis, i.e., $h = 0.05$. In this example, we use a single component, with the intention of showing the flexibility of supershapes. Therefore, this constitutes a shape optimization example. Geometry projection methods have been used for shape optimization in Norato et al. (2004; Wein and Stingl 2018).

The bounds for this problem are $a, b \in [0.1, 2]$, $m \in [2, 6]$, $n_1, n_2, n_3 \in [1, 10]$, $s \in [1, 2]$, $\mathbf{x}_c \in [0, 2]$, $i = 1, 2$, and $\phi \in [-\pi, \pi]$. The convergence tolerances are $kktol = 1E-3$ and $objtol = 1E-3$. The initial design, shown in Fig. 6a, is a circle placed at the center of the design region, and has $a = b = 0.5$, $m = 2$, $n_1 = n_2 = n_3 = 2$, $s = 1.0$, and $\phi = 0$.

Designs corresponding to several iterations in the optimization are shown in Fig. 6. The optimization converges in 122 iterations. The optimal design is $m = 3.566$, $a = 1.196$, $b = 0.443$, $n_1 = 1$, $n_2 = 1$, $n_3 = 1.115$, $\mathbf{x}_c = [0.128 \ 0.997]^T$, $\phi = -0.005$, $s = 1.835$ and $\alpha = 1$. Its compliance is $C = 0.025484$, and its volume fraction is $v_f = 0.499996$.

The supershape plot is clipped to the design region Ω , since this is the only portion of the supershape that has an effect in the compliance and the volume (correspondingly, the fabrication of the structure should cut material outside

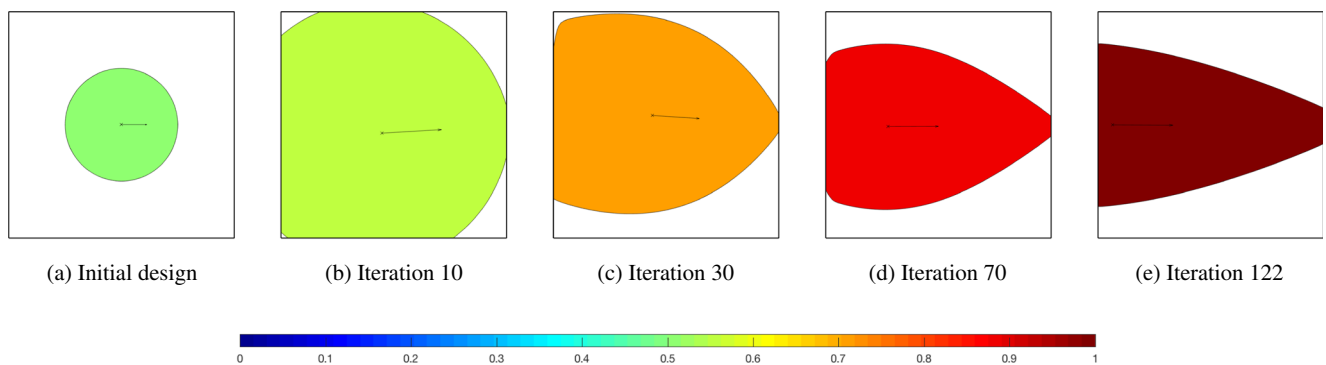


Fig. 6 Design iterates for short cantilever beam problem. The color denotes the size variable α . In this and subsequent figures, the arrow originates at the center \mathbf{x}_c , its size is $0.5 \max(a, b)$, and its orientation is that of the supershape. Only the portion of the supershape inside the design region Ω is shown

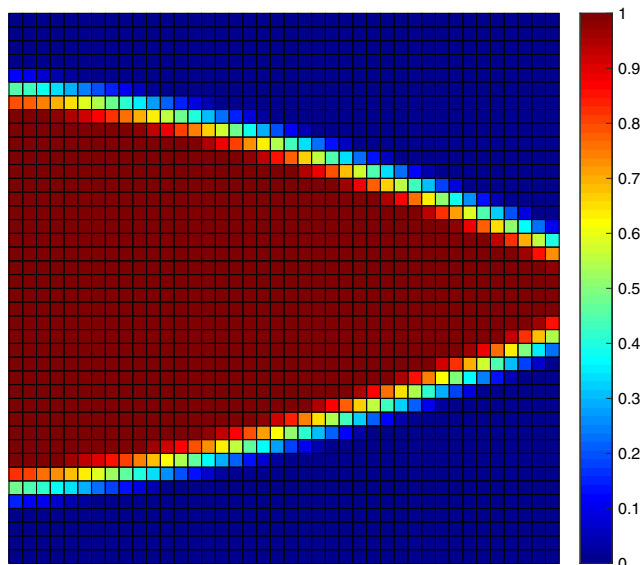


Fig. 7 Composite density $\bar{\rho}$ for optimal short cantilever beam

the design region). The composite density $\bar{\rho}$ for the optimal design is shown in Fig. 7; note that in this case the composite density is the same as the effective density $\hat{\rho}$, since there is only one shape in the design.

The optimal design is also shown in Fig. 8 without clipping to Ω . In the same figure, we compare the optimal supershape design to the shape corresponding to uniform energy density along the boundary, and with volume fraction of 0.5. For an elastic structure with a single design-independent loading, and as long as the design region allows it, the minimum compliance optimal shape has uniform strain energy density along the boundary (Pedersen 2000). Moreover, under the same conditions, the stiffest design is also the strongest design. Thus, the uniform-energy shape is the same as the uniform-strength shape, which Galileo found to be parabolic for a cantilever beam (Timoshenko 1953). Using Euler beam theory, it can be readily shown that

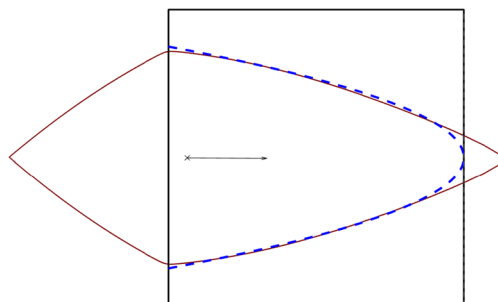


Fig. 8 Optimal supershape for short cantilever beam design (continuous line), without clipping to design region Ω . Dashed line corresponds to the uniform-energy boundary for 0.5 volume fraction

the uniform-energy shape for this design region and for a volume fraction of 0.5 is given by

$$y = \pm \frac{3}{4} \sqrt{1 - \frac{x}{2}} \tag{30}$$

The above expression is based on an origin $(x, y) = (0, 0)$ at the midpoint of the left side of the design region. As seen in Fig. 8, there is good agreement with the uniform-energy shape.

7.2 MBB beam

Our next example corresponds to the Messerschmitt-Bölkow-Blohm (MBB) beam, extensively studied in topology optimization. The problem setup is shown in Fig. 9. The volume fraction constraint for this problem $v_f^* = 0.4$. A 160×40 grid is used for the analysis, i.e., $h = 0.125$. Taking advantage of the symmetry of the problem, we perform the analysis and design on half of the design region.

The bounds for this problem are $a, b \in [0.4, 5]$, $m \in [2, 6]$, $s \in [1, 2]$, $\mathbf{x}_{c_x} \in [0, 20]$, $\mathbf{x}_{c_y} \in [0, 5]$ and $\phi \in [-\pi, \pi]$. In these examples, we assign $n_1 = n_2 = n_3 = 10$ and fix these parameters in the optimization to avoid unfavorable, high aspect ratio and/or highly concave shapes that could cause the geometry projection to fail. The convergence tolerances are $kktol = 1E-3$ and $objtol = 1E-5$. We solve the topology optimization using different initial designs with supershapes arranged in an $N_x \times N_y$ grid, with positions \mathbf{x}_{c_x} and \mathbf{x}_{c_y} evenly spaced by $20/(N_x + 1)$ and $5/(N_y + 1)$ respectively. The remaining parameters for the shapes in the initial design are $a = b = 0.5$, $m = 4$, $s = 1.0$, and $\phi = 0$.

Figure 10 shows the initial and optimal designs for beams with 5×2 , 10×2 and 10×4 supershapes. The optimization converges in 148, 131 and 152 iterations, and the compliance values of the optimal designs are 0.5938, 0.5439, and 0.5006 respectively. The volume fraction of the optimal design in all cases is less than $1E-3$ from the imposed limit of 0.4. We first observe that the optimal designs are reminiscent of known density-based optimal topologies for the MBB beam. Although there are differences in the designs obtained, the bounds on the supershape parameters clearly impose a minimum size on the design because we do not see thinner members in

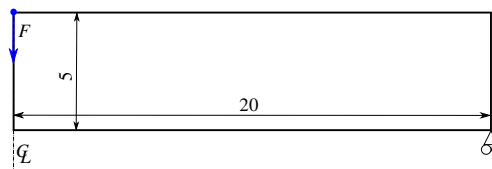


Fig. 9 MBB beam problem

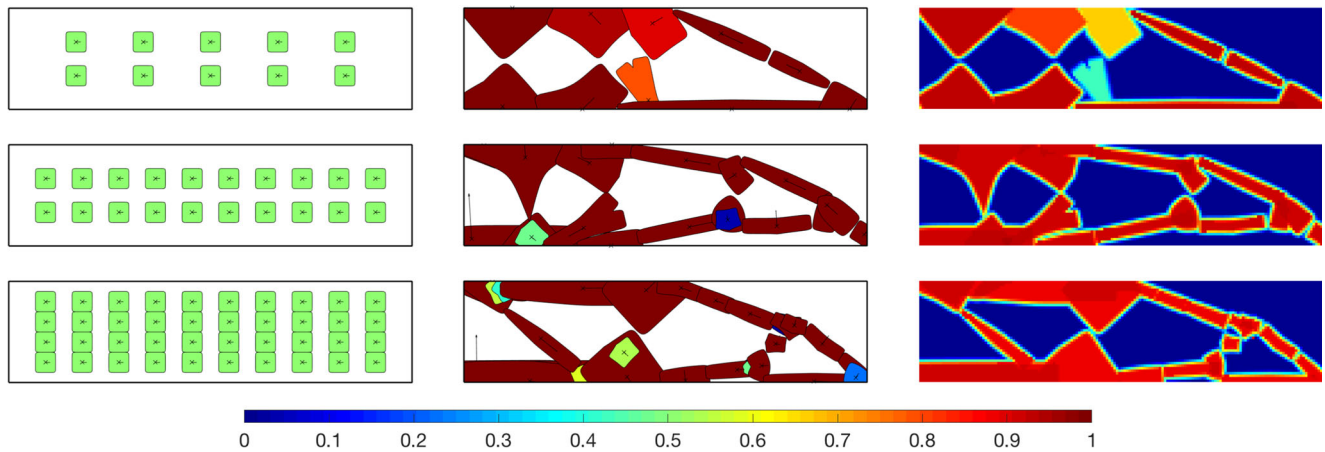


Fig. 10 Initial (left) and optimal (center) designs, and composite density (left) for MBB problems. The color of the shapes denotes their size variable α . The color scale is the same for the shapes size variables and the composite density. Supershapes are clipped to the design region Ω

one design than in the others. We note that the optimizer removes some shapes in the 10×2 and 10×4 designs.

This example illustrates one aspect of our formulation, and that is that in some cases there are gaps between the shapes. As discussed in Section 3, this owes primarily to the fact that these gaps are much stiffer than void regions that are away from the shapes. Since the composite density in these gaps is in general closer to unity than it is to zero as a result of the geometry projection with respect to the nearby shapes, and since this composite density is reflected in the analysis via (17), it is not unreasonable to close these gaps

in the actual design. This can be easily done by modifying the primitives around the gap.

We also use this example to discuss the effect of mesh size on the results. As discussed in Section 6.4, we require the element size to be smaller than the minimum size of the shape to ensure well-defined sensitivities. Beyond this requirement, the mesh can be subsequently refined. To illustrate the effect of mesh refinement, we perform the optimization for the MBB beam with 5×2 shapes (with initial design shown in Fig. 10, top/left) and with finite element meshes of 80×20 , 160×40 and 200×50

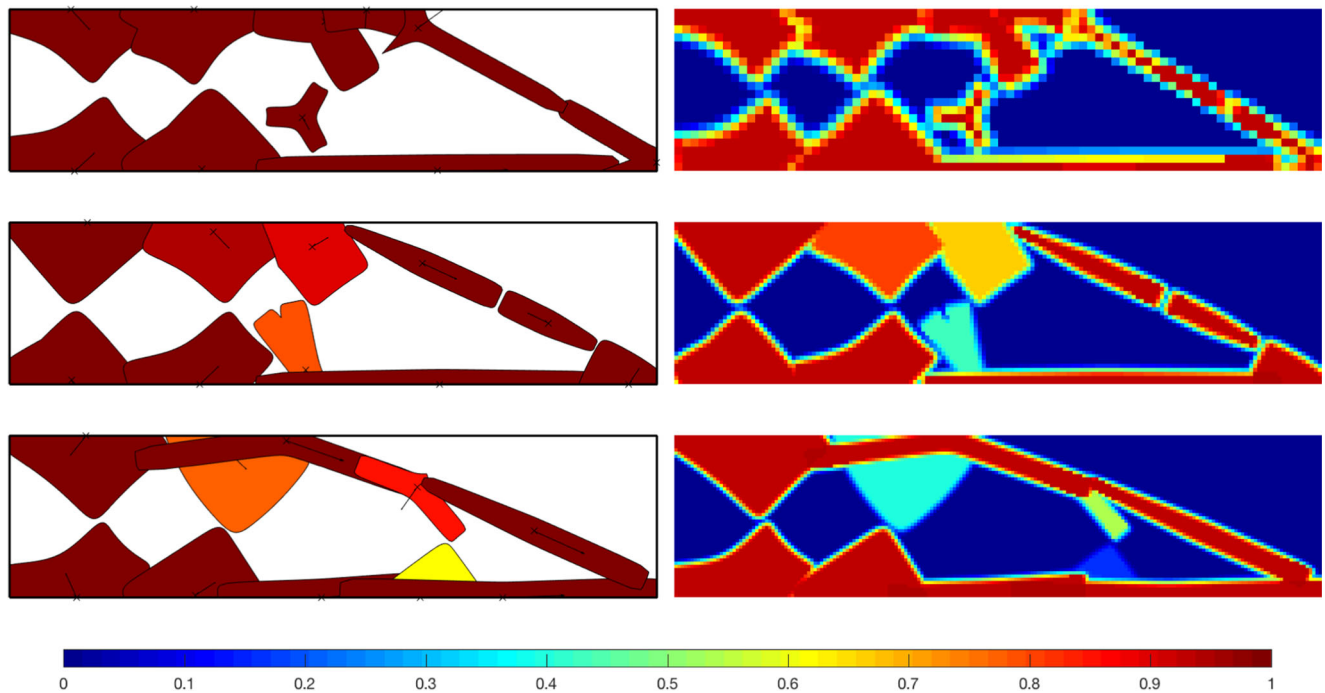


Fig. 11 Designs (left) and composite density (right) of MBB problems for meshes of 80×20 (top), 160×40 (middle) and 200×50 (bottom) elements. The color of the shapes denotes their size variable α . The

color scale is the same for the shapes size variables and the composite density. Supershapes are clipped to the design region Ω

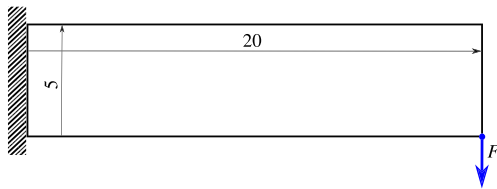


Fig. 12 Cantilever beam problem

50 elements (the second result is the same one shown in Fig. 10, top). The results are shown in Fig. 11. The optimization converges in 250, 148 and 201 iterations, and the compliance values of the optimal designs are 0.5283, 0.5439, and 0.5714 respectively.

Two observations are worth making. First, a minimum length scale is in general satisfied, and this is dictated by the fixed number of shapes, and by the lower bounds on their dimensions. That is, regardless of the mesh size, we expect to obtain designs of similar complexity if we use the same design representation (number of shapes and bounds on shape parameters). Second, as we expect, a finer mesh provides a better resolution of the shapes upon projection, and therefore it provides better control over the shape boundaries than coarser meshes. The compliance values

actually increase when refining the mesh, but part (if not all) of that increase is due to the fact that the finer mesh is more compliant, since it has a larger number of analysis degrees of freedom.

7.3 Long cantilever beam

The designs obtained in the preceding section are reminiscent of MBB beam designs obtained with free-form topology optimization methods. This is an indication that the proposed method obtains expected results. However, these designs can be more easily and more efficiently obtained using, for example, density-based topology optimization. Furthermore, one could post-process the optimal free-form design and 'fit' a set of supershapes to it, as it has been done in the past using, e.g., splines (cf. Bendsoe and Sigmund 2003 and the extensive references therein).

Where supershapes provide interesting possibilities is in producing designs made of geometric primitives such as, for example, rectangles, ellipses and triangles. As explained in Section 1, the availability of datum geometric entities helps designs with primitives facilitate certain aspects of the production of mechanical components that are not available in free-form designs.

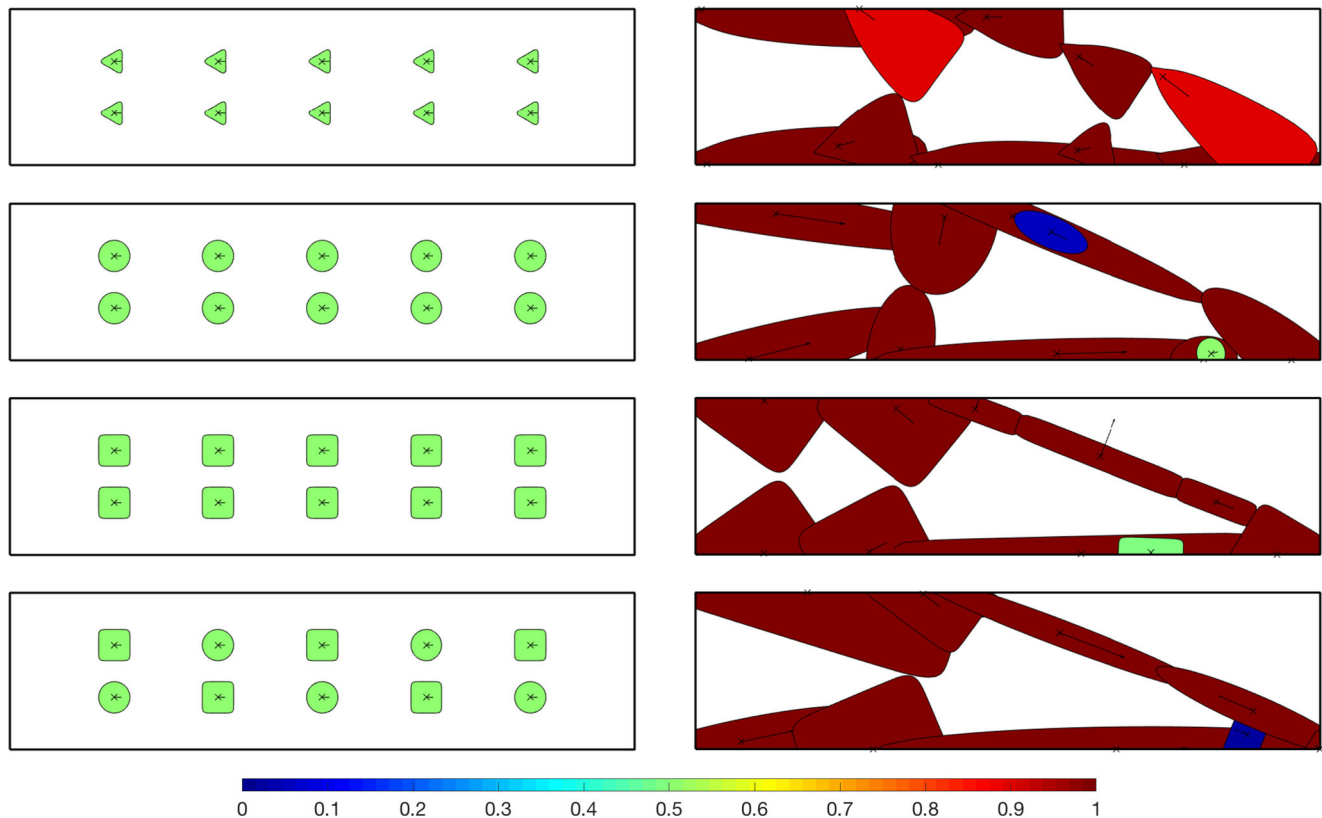


Fig. 13 Initial (left) and optimal (right) designs for cantilever beam problem using triangles, ellipses, rectangles, and a combination of ellipses and rectangles, from top to bottom respectively. The color of

the shapes (corresponding to the horizontal color bar) denotes their size variable α . Supershapes are clipped to the design region Ω

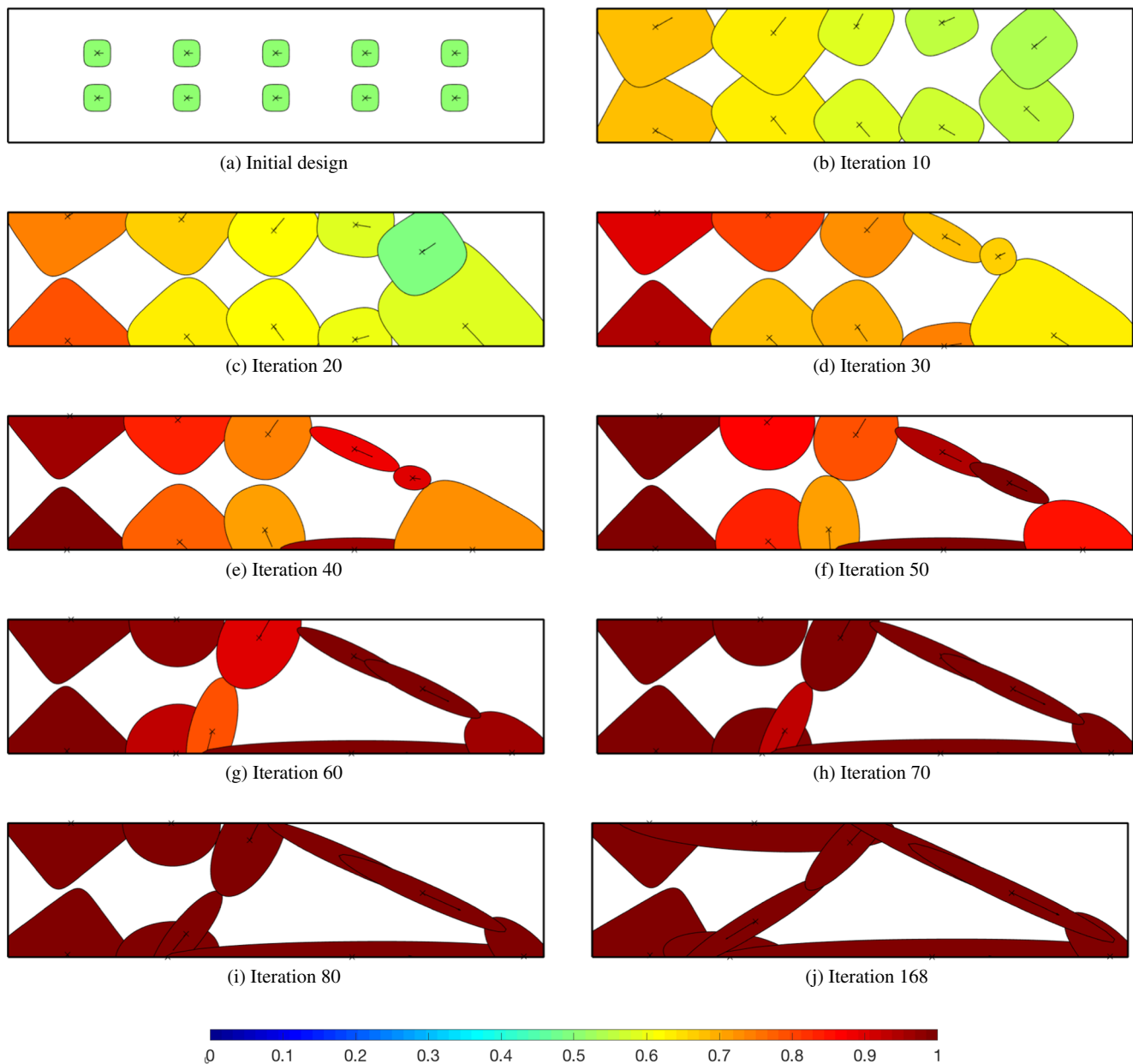


Fig. 14 Design iterates for cantilever beam problem using shapes that morph into ellipses or rectangles. The initial design has $n = n_1 = n_2 = n_3 = 4$. The color of the shapes (corresponding to the horizontal color bar) denotes their size variable α . Supershapes are clipped to the design region Ω

To demonstrate the design of structures made of primitives with our method, in this section we design a long cantilever beam with the same dimensions and analysis mesh as the beam of the preceding section, but with the load and boundary conditions shown in Fig. 12. The volume fraction constraint for this problem is again $v_f^* = 0.4$. All examples employ a 5×2 grid of supershapes in the initial design, and the convergence tolerances are $kktol = 1E-3$ and $objtol = 1E-4$. In all examples, $a = b = 0.5$, $\phi = 0$, $s = 1$ and $\alpha = 0.5$ for all supershapes in the initial design. Likewise, the bounds $\phi \in [-\pi, \pi]$, $s \in [1, 3]$, $\mathbf{x}_{c_x} \in [0, 20]$ and $\mathbf{x}_{c_y} \in [0, 5]$ are applied to all examples.

Figure 13 shows the initial and optimal designs for a beam whose initial design is made of triangles, ellipses, rectangles, and a combination of ellipses and rectangles. For triangles, we fix $m = 3$, $n_1 = 4$, and $n_2 = n_3 = 8$; for ellipses, we fix $m = 4$, $n_1 = n_2 = n_3 = 2$; and for rectangles, we fix $m = 4$, $n_1 = n_2 = n_3 = 8$ (note that rectangles are effectively modeled using superellipses). We note that in the case of combined ellipses and rectangles, the foregoing parameters for each supershape are fixed throughout the optimization, i.e. a supershape that is an ellipse or a rectangle in the initial design remains an ellipse or rectangle throughout the optimization. In all cases except

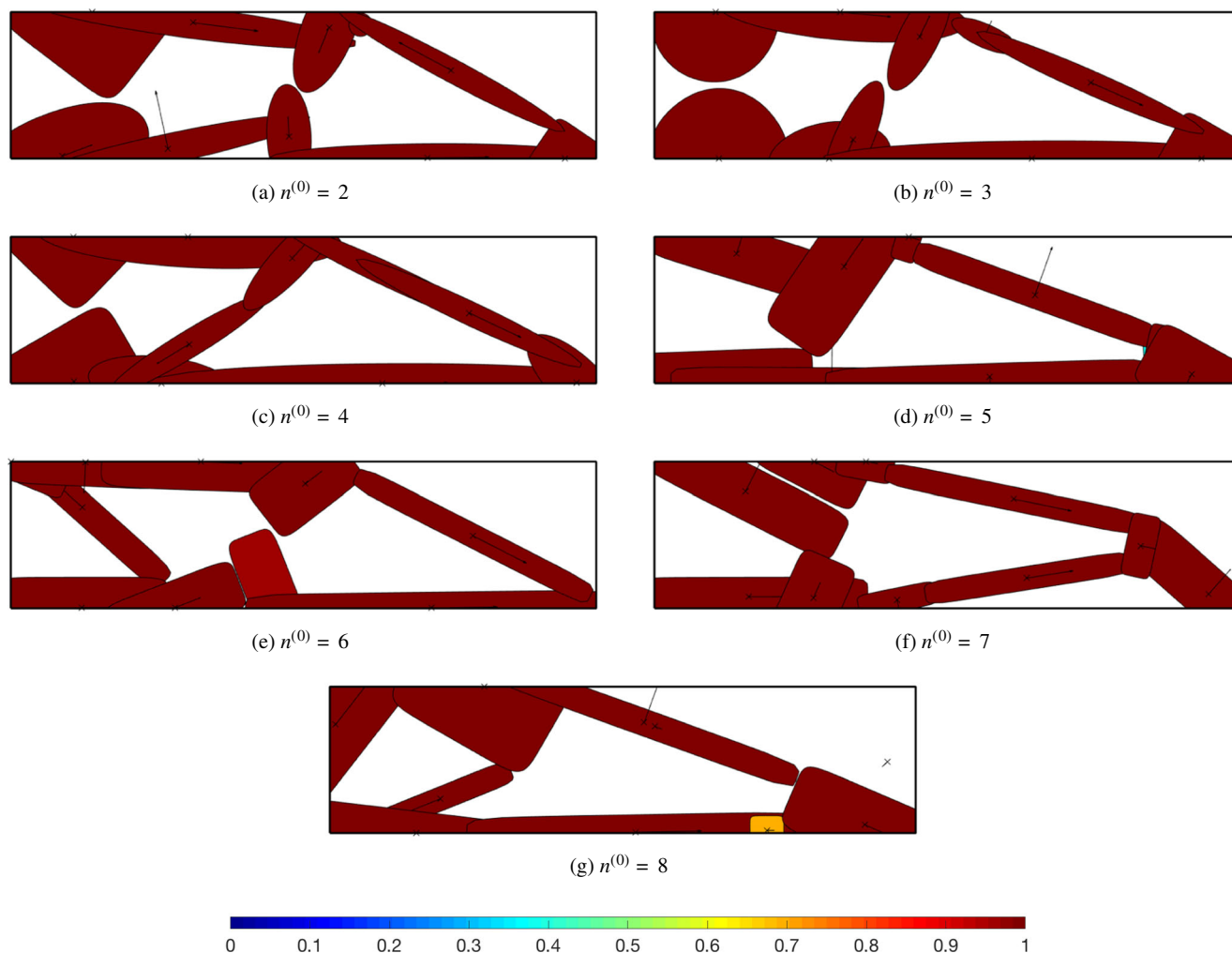


Fig. 15 Optimal designs for cantilever beam problem using shapes that morph into ellipses or rectangles with different initial values of the parameter $n = n_1 = n_2 = n_3$. The color of the shapes (corresponding

to the horizontal color bar) denotes their size variable α . Supershapes are clipped to the design region Ω

the triangles, we impose the bounds $a, b \in [0.4, 5]$. In the case of triangles, it is more difficult to capture a non-equilateral triangle with the supershapes, and larger a/b aspect ratios tend to render shapes that look more like water drops, and so we impose a smaller range of $a, b \in [0.4, 4]$. Also in the case of triangles, we fix the value of $b = 0.4$, because for $b < a$ one obtains 'bowtie' shapes. Note that we can still get triangles with size larger than 0.4 thanks to the scale factor s .

From top to bottom in Fig. 13, the compliance values for the optimal designs are $C = 0.5759, 0.5263, 0.566$ and 0.5348 . These designs are attained in 150, 127, 129 and 141 iterations respectively. All designs satisfy the volume fraction constraint within $1E-3$. Clearly, the design that performs the worst is the one with triangles. We posit this is because the geometry of the better designs cannot easily be captured with ten triangles.

An even more interesting possibility is to let shapes morph into a selected set of primitives.² We illustrate this idea by repeating the cantilever beam design, while forcing shapes to morph into either rectangles or ellipses. Both shapes have the same symmetry ($m = 4$), but different exponents ($n = 2$ for ellipses, $n = 8$ for rectangles, $n_1 = n_2 = n_3 = n$). In the optimization, we fix m , but we let n vary within $[2, 8]$. To steer shapes towards becoming ellipses or rectangles, we add the following 'shape' constraint to the optimization problem of (25):

$$g_s(\mathbf{z}) := \frac{1}{81N_s} \sum_i^{N_s} (n_i - 2)^2 (n_i - 8)^2 - g_s^* \leq 0 \quad (31)$$

In the above expression, n_i is the value of n for shape i . Clearly, if $n_i = 2$ or $n_i = 8$, the contribution of shape

²A similar strategy to produce primitive-shaped holes that are interpolated between several primitives is discussed in Mei et al. (2008).

Table 2 Optimization results for long cantilever designs of Fig. 15

$n^{(0)}$	C	It.	$N_{rectangles}$	$N_{ellipses}$
2	0.556	137	2 ($n_i \geq 7.999$)	8 ($n_i \leq 2.039$)
3	0.569	92	1 ($n_i = 7.998$)	9 ($n_i \leq 2.117$)
4	0.509	168	2 ($n_i \geq 7.999$)	8 ($n_i \leq 2.124$)
5	0.575	121	10 ($n_i \geq 7.848$)	0
6	0.503	239	10 ($n_i \geq 7.863$)	0
7	0.590	119	10 ($n_i \geq 7.898$)	0
8	0.582	214	10 ($n_i \geq 7.854$)	0

i to the sum is zero. The number 81 in the denominator ensures that the value in the sum is at most 1.0 (which occurs when $n_i = 5$). The factor outside the sum helps ensure that $g_s \in [0, 1]$, hence the choice of the constraint limit g_s^* can be made independent of the number of shapes. In the following examples, we use $g_s^* = 1E-3$. To avoid the shapes from ‘locking’ prematurely into the desired primitives, we use a continuation strategy, whereby we set $g_s^* = 1$ in the first iteration, and then decrease it by $2E-3$ in each iteration until we reach the desired value of $g_s^* = 1E-3$ (that is, it takes at least 50 iterations to attain the desired constraint limit). In addition to this continuation strategy, for this example we employ a tighter move limit of $M = 0.025$. Figure 14 shows several design iterates using a starting design with $n = n_1 = n_2 = n_3 = 4$. The optimal design is made of two rectangles ($n \geq 7.99$) and six ellipses ($n \leq 2.124$). The figure shows how the shapes morph into one of the two primitives as the optimization progresses.

This example also serves to illustrate a characteristic of the proposed method, namely that it is more prone to converging to unfavorable local minima than free-form topology optimization methods. This is a trait of geometry projection methods we have observed before (Norato et al. 2015; Zhang et al. 2017a), and it is likely due to the more restrictive representation of possible geometries imposed by the discrete geometric components. To illustrate this dependency, we show in Fig. 15 the results of the same optimization problem with different initial values of n . Table 2 lists results for the corresponding optimal designs, including compliance values, number of iterations to convergence, and number of rectangles ($n \approx 8$) and ellipses ($n \approx 2$) with their respective ranges for n_i . All designs satisfy the volume fraction and shape constraints tightly.

8 Conclusions

This paper demonstrates a method to perform topology optimization using supershapes. The numerical examples show the method effectively produces designs similar to those obtained with free-form topology optimization

techniques. The appealing feature of the proposed method is the ability to produce designs made exclusively of various geometric primitives that are all modeled with a single representation, the superformula. The fact that different primitives can be represented via a single equation also allows to produce designs whereby shapes morph into specified types of primitives. We demonstrated this capability by introducing a shape constraint that steers the shapes into becoming one of two types of primitives (rectangles and ellipses).

To fulfill its final purpose, which is to produce designs made of primitives that are amenable to production, the proposed method requires further advancement. The extension to 3-dimensional problems is the most immediate one. The strict imposition of minimum size constraints, which in this work are loosely enforced by imposing bounds on the supershape parameters, is also an important need. A deeper understanding of the interactions between supershape parameters is needed in order to facilitate the imposition of manufacturing-driven geometric constraints. Finally, strategies to circumvent convergence to an undesired local minimum are necessary. These strategies are not only important from the point of view of how good the optimal design is, but also to free the designer from having to try different initial designs. These research directions are a matter of ongoing and future work.

Acknowledgements The author expresses his gratitude to the National Science Foundation, award CMMI-1634563, for support to conduct this work, and to Prof. Krister Svanberg for kindly providing his MMA MATLAB optimizer to perform the optimization. The author also gives special thanks to Prof. Horea Ilies for numerous discussions on the geometry aspects of this work.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix A: Interactive script for plotting supershapes

The following is a Mathematica script to generate a supershape plot that can be interactively manipulated. The equations correspond to the modified superformula of (3). The shape center position \mathbf{x}_c in (6) is not part of the equation, but the shape orientation ϕ and scaling factor s are. As noted in Section 2, the angle in the superformula has to be wrapped to the range $[-\pi, \pi]$, as otherwise the shape may not be closed. For this reason, we cannot use the `PolarPlot` function in Mathematica, because if the angle used for plotting is $\theta + \phi$, the argument for (3) will be outside of the foregoing range for a range of values

of ϕ . To circumvent this, one would have to make the plotting range a function of ϕ . An easier solution is to use the `ParametricPlot` function. The entire script is as follows, which produces an interactive plot like the one shown in Fig. 16.

```
r[t_, m_, a_, b_, n1_, n2_, n3_, s_, eps_] :=
s ((eps+(1/(2 a^2)) (1+Cos[m t/2]))^(n2/2)+
(eps+(1/(2 b^2)) (1-Cos[m t/2]))^(n3/2))^
(-1/n1)
```

```
x[t_, m_, a_, b_, n1_, n2_, n3_, eps_, phi_, s_] :=
r[t, m, a, b, n1, n2, n3, s, eps] Cos[t+phi]
```

```
y[t_, m_, a_, b_, n1_, n2_, n3_, eps_, phi_, s_] :=
r[t, m, a, b, n1, n2, n3, s, eps] Sin[t+phi]
```

Manipulate

```
[ ParametricPlot[{x[t, m, a, b, n1, n2, n3, e,
phi, s], y[t, m, a, b, n1, n2, n3, e, phi, s]},
{t, -Pi, Pi}, PlotStyle -> Color], {m, 2, 6},
{a, 0.1, 2}, {b, 0.1, 2}, {n1, 1, 10}, {n2, 1, 10},
{n3, 1, 10}, {e, 0.001, 0.1}, {phi, -Pi, Pi},
{s, 1, 2}, {Color, Blue}]
```

Fig. 16 Interactive Mathematica plot of supershape

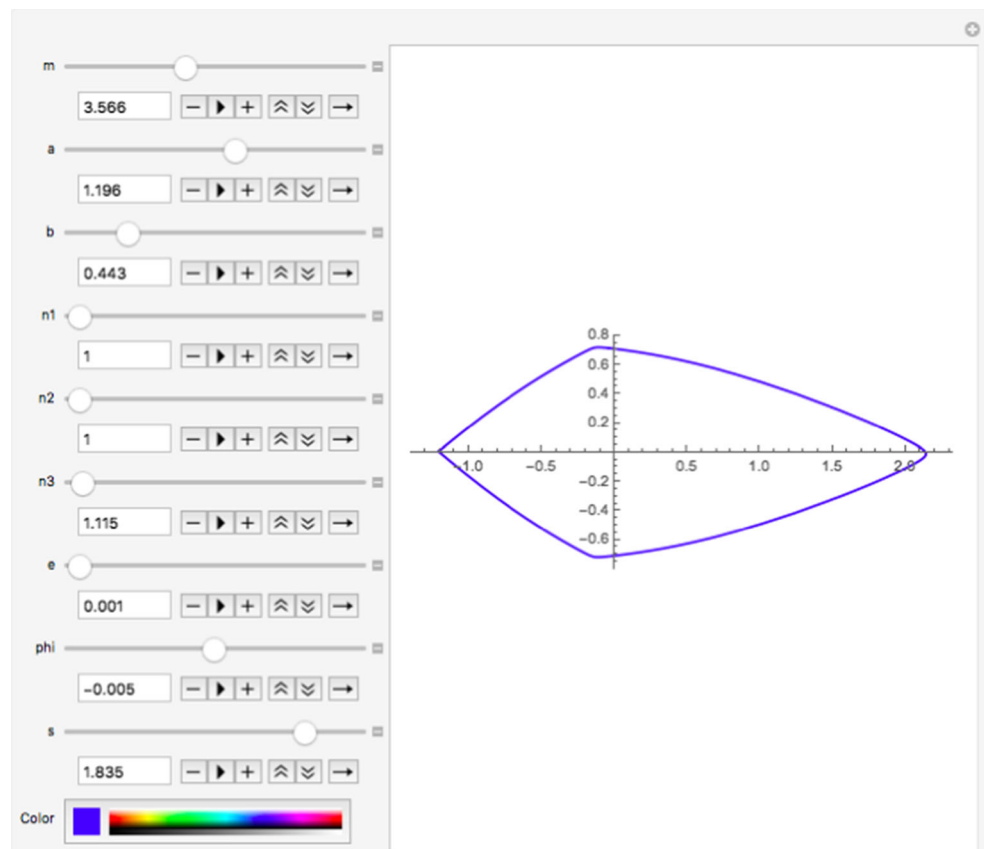


Figure 16 shows the parameter values for the optimal design of the example in Section 7.1. The plotting range can be adjusted for each one of the parameters in the last statement of the script. If, say, $n = n_1, n_2, n_3$, then the last statement should change to

```
Manipulate[
ParametricPlot[{x[t, m, a, b, n, n, n, e, phi, s],
y[t, m, a, b, n1, n2, n3, e, phi, s]},
{t, -Pi, Pi}, PlotStyle -> Color], {m, 2, 6},
{a, 0.1, 2}, {b, 0.1, 2}, {n, 1,
10}, {e, 0.001, 0.1},
{phi, -Pi, Pi}, {s, 1, 2}, {Color, Blue}]
```

Appendix B: Sensitivity analysis

In this Appendix, we list the expressions necessary to obtain design sensitivities necessary for the gradient-based optimization. For brevity, we do not derive these expressions; however, they can be readily obtained. We start by stating the sensitivities for the geometry projection and the signed distance, and then state sensitivities for the optimization functions.

B.1 Composite density

The derivative of the composite density of (16) with respect to a design variable z is given by

$$D_z \tilde{\rho}(\mathbf{p}, \mathbf{z}) = \frac{1 - \rho_{\min}}{N_s} \exp\left(k \frac{\rho_{\min} - \tilde{\rho}}{1 - \rho_{\min}}\right) \sum_{i=1}^{N_s} e^{k \hat{\rho}_i} D_z \hat{\rho}_i \quad (32)$$

where we recall that $\hat{\rho}_i$ is the effective density for supershape i .

Here and henceforth the arguments are removed from the right-hand side of expressions for brevity. From (12), the design sensitivity of the effective density is

$$D_z \hat{\rho}_i(\mathbf{p}, \mathbf{z}) = \begin{cases} 0 & \text{if } z \in \mathbf{z}_j, j \neq i \\ q \alpha_i^{q-1} \rho_i & \text{if } z \equiv \alpha_i \\ D_z \rho_i & \text{otherwise} \end{cases} \quad (33)$$

where we recall from (13) that \mathbf{z}_j is the vector of design parameters for supershape j . From (10), the sensitivity of the projected density ρ_i is in turn given by

$$D_z \rho_i(d(\mathbf{p}, \mathbf{z})) = \begin{cases} 0 & \text{if } d > r \text{ or } d < -r \\ -\frac{2}{\pi r^2} \sqrt{r^2 - d^2} D_z d & \text{otherwise} \end{cases} \quad (34)$$

In the expression above and in the following section, it is understood that $z \in \mathbf{z}_i$, since the projected density for supershape i depends only on its parameters, and hence the sensitivity with respect to $z \in \mathbf{z}_j, j \neq i$ is zero.

B.2 Signed distance

Following (18) and (20), we rewrite the squared distance as

$$d^2(\mathbf{p}, \mathbf{z}_i) = (\mathbf{p} - \mathbf{c})^T (\mathbf{p} - \mathbf{c}) \quad (35)$$

with $\mathbf{c} = \mathbf{x}(\theta^*(\mathbf{z}_i), \mathbf{z}_i)$. Differentiating both sides with respect to a design parameter $z \in \mathbf{z}_i$, and rearranging, we find

$$D_z d(\mathbf{p}, \mathbf{z}_i) = -\frac{1}{d} (\mathbf{p} - \mathbf{c})^T D_z \mathbf{x} \quad (36)$$

We note that when $\mathbf{p} = \mathbf{c}$, this derivative is undefined since $\mathbf{p} - \mathbf{x} = \mathbf{0}$ and $d = 0$. Although possible, we did not observe this situation in our numerical examples. Since we ignore the case when d is exactly zero, we note the design derivatives of the sign of (24) is zero, hence we make d in the denominator $D_z d$ of (36) the signed distance (as opposed to the positive root of d^2). Using the chain rule, we get

$$D_z \mathbf{x}(\theta^*(\mathbf{z}_i), \mathbf{z}_i) = \partial_\theta \mathbf{x} D_z \theta^* + \partial_z \mathbf{x} \quad (37)$$

The first-order optimality condition of the problem of (20) dictates that $D_z \theta^*$ must be zero, therefore

$$D_z \mathbf{x}(\theta^*(\mathbf{z}_i), \mathbf{z}_i) = \partial_z \mathbf{x} \quad (38)$$

The sensitivities $\partial_z \mathbf{x}$ with respect to each of the supershape’s parameters can be readily derived from (3–6), and are given by:

$$\partial_a \mathbf{x} = \left[\frac{s \tilde{r}^{n_1+1} n_2 \tilde{x}^{\frac{n_2}{2}-1} \left(1 + \cos\left(\frac{m\theta}{2}\right)\right)}{2a^3 n_1} \right] \mathbf{R}_{\theta+\phi} \mathbf{e}_1 \quad (39)$$

$$\partial_b \mathbf{x} = \left[\frac{s \tilde{r}^{n_1+1} n_3 \tilde{y}^{\frac{n_3}{2}-1} \left(1 - \cos\left(\frac{m\theta}{2}\right)\right)}{2b^3 n_1} \right] \mathbf{R}_{\theta+\phi} \mathbf{e}_1 \quad (40)$$

$$\partial_m \mathbf{x} = -\frac{s \tilde{r}^{n_1+1} \theta}{8n_1} \sin\left(\frac{m\theta}{2}\right) \left[\frac{n_2}{a^2} \tilde{x}^{\frac{n_2}{2}-1} - \frac{n_3}{b^2} \tilde{y}^{\frac{n_3}{2}-1} \right] \mathbf{R}_{\theta+\phi} \mathbf{e}_1 \quad (41)$$

$$\partial_{n_1} \mathbf{x} = \frac{s \tilde{r}}{n_1^2} \ln(\tilde{r}^{-n_1}) \mathbf{R}_{\theta+\phi} \mathbf{e}_1 \quad (42)$$

$$\partial_{n_2} \mathbf{x} = -\frac{s \tilde{r}^{n_1+1}}{2n_1} \tilde{x}^{\frac{n_2}{2}} \ln(\tilde{x}) \mathbf{R}_{\theta+\phi} \mathbf{e}_1 \quad (43)$$

$$\partial_{n_3} \mathbf{x} = -\frac{s \tilde{r}^{n_1+1}}{2n_1} \tilde{y}^{\frac{n_3}{2}} \ln(\tilde{y}) \mathbf{R}_{\theta+\phi} \mathbf{e}_1 \quad (44)$$

$$\partial_{\mathbf{x}_c} \mathbf{x} = \mathbf{I} \quad (45)$$

$$\partial_\phi \mathbf{x} = s \tilde{r} \mathbf{R}_{\theta+\phi+\pi/2} \mathbf{e}_1 \quad (46)$$

$$\partial_s \mathbf{x} = \tilde{r} \mathbf{R}_{\theta+\phi} \mathbf{e}_1 \quad (47)$$

The derivative $\partial_\theta d^2$ corresponding to $f'(\theta)$ in (20) and required to solve the minimum squared distance problem is given by

$$\partial_\theta d^2 = -2s(\mathbf{p} - \mathbf{c})^T \left[\frac{m \tilde{r}^{n_1+1}}{8n_1} \sin\left(\frac{m\theta}{2}\right) \left(\frac{n_2}{a^2} \tilde{x}^{\frac{n_2}{2}-1} - \frac{n_3}{b^2} \tilde{y}^{\frac{n_3}{2}-1} \right) \mathbf{R}_{\theta+} \right. \\ \left. \tilde{r} \mathbf{R}_{\theta+\pi/2} \right] \mathbf{R}_\phi \mathbf{e}_1 \quad (48)$$

The sensitivity expressions in this section were verified using Mathematica.

B.3 Optimization functions

The sensitivity of the compliance of (25) can be readily obtained using adjoint analysis (see, for example, Bendsøe and Sigmund 2003) as:

$$D_z C(\mathbf{z}) = - \int_\Omega \nabla \mathbf{u} \cdot D_z \mathbb{C}(\mathbf{p}, \mathbf{z}) \nabla \mathbf{u} \, dv \quad (49)$$

where $D_z \mathbb{C}$ can be obtained by differentiating (17) as

$$D_z \mathbb{C}(\mathbf{p}, \mathbf{z}) = D_z \tilde{\rho} \mathbb{C}_0 \quad (50)$$

The sensitivity of the composite density can be computed using the expressions in the previous two sections. Using the finite element discretization, and since we use a uniform composite density per element, the compliance sensitivity can be computed as

$$D_z C = - \sum_{j=1}^{N_{el}} D_z \tilde{\rho}_j \mathbf{U}_j^T \mathbf{K}_{j_0} \mathbf{U}_j \quad (51)$$

where $D_z \tilde{\rho}_j$, \mathbf{U}_j and \mathbf{K}_{j_0} are the sensitivity of the composite density, the vector of nodal displacements, and the fully solid stiffness matrix of element j respectively.

The sensitivity of the volume of (26) is similarly given by

$$D_z V(\mathbf{z}) = \int_{\Omega} D_z \tilde{\rho}(\mathbf{p}, \mathbf{z}) dv \quad (52)$$

and can be simply computed in the finite element discretization as $D_z \tilde{\rho}_j v_j$, where v_j is the volume of element j .

Finally, the sensitivity of the shape constraint of (31) is simply

$$D_z g_s(\mathbf{z}) = \begin{cases} \frac{4}{81N_s} (n_i - 8)(n_i - 2)(n_i - 5) & \text{if } z \equiv n_i \\ 0 & \text{otherwise} \end{cases} \quad (53)$$

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

- Bell B, Norato J, Tortorelli D (2012) 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
- Bendsøe MP, Sigmund O (2003) Topology optimization: theory, methods and applications. Springer
- Bloomenthal J (1990) Techniques for implicit modeling. Tech. rep., Xerox PARC Technical Report, P89-00106, also in SIG-GRAPH'90 Course Notes on Modeling and Animating
- Chen J, Shapiro V, Suresh K, Tsukanov I (2007) Shape optimization with topological changes and parametric control. *Int J Numer Methods Eng* 71(3):313–346
- Chen S, Wang MY, Liu AQ (2008) Shape feature control in structural topology optimization. *Comput Aided Des* 40(9):951–962
- Cheng G, Mei Y, Wang X (2006) A feature-based structural topology optimization method. In: IUTAM Symposium on topological design optimization of structures, machines and materials. Springer, pp 505–514
- Eschenauer HA, Kobelev VV, Schumacher A (1994) Bubble method for topology and shape optimization of structures. *Struct Optim* 8(1):42–51
- Gielis J (2003) A generic geometric transformation that unifies a wide range of natural and abstract shapes. *Amer J Botany* 90(3):333–338
- Gielis J (2017) The geometrical beauty of plants. Atlantis Press
- Griggeman NT (1970) Lamé ovals. *The Mathematical Gazette*, pp 31–37
- Guo X, Zhang W, Zhong W (2014a) Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. *J Appl Mech* 81(8):081,009
- Guo X, Zhang W, Zhong W (2014b) Topology optimization based on moving deformable components: a new computational framework. arXiv:14044820
- Guo X, Zhang W, Zhang J, Yuan J (2016) Explicit structural topology optimization based on moving morphable components (mmc) with curved skeletons. *Comput Methods Appl Mech Eng* 310:711–748
- Hoang VN, Jang GW (2016) Topology optimization using moving morphable bars for versatile thickness control. *Computer Methods in Applied Mechanics and Engineering*
- Lin HY, Rayasam M, Subbarayan G (2015) Isocomp: unified geometric and material composition for optimal topology design. *Struct Multidiscip Optim* 51(3):687–703
- Liu J, Ma Y (2016) A survey of manufacturing oriented topology optimization methods. *Adv Eng Softw* 100:161–175
- Liu T, Wang S, Li B, Gao L (2014) A level-set-based topology and shape optimization method for continuum structure under geometric constraints. *Struct Multidiscip Optim* 50(2):253–273
- Mei Y, Wang X, Cheng G (2008) A feature-based topological optimization for structure design. *Adv Eng Softw* 39(2):71–87
- Norato J, Haber R, Tortorelli D, Bendsøe MP (2004) A geometry projection method for shape optimization. *Int J Numer Methods Eng* 60(14):2289–2312
- Norato J, Bell B, Tortorelli D (2015) A geometry projection method for continuum-based topology optimization with discrete elements. *Comput Methods Appl Mech Eng* 293:306–327
- Pedersen P (2000) On optimal shapes in materials and structures. *Struct Multidiscip Optim* 19(3):169–182
- Qian Z, Ananthasuresh G (2004) Optimal embedding of rigid objects in the topology design of structures. *Mech Based Des Struct Mach* 32(2):165–193
- Saxena A (2011) Are circular shaped masks adequate in adaptive mask overlay topology synthesis method? *J Mech Des* 133(1):011,001
- Seo YD, Kim HJ, Youn SK (2010) Isogeometric topology optimization using trimmed spline surfaces. *Comput Methods Appl Mech Eng* 199(49):3270–3296
- Shapiro V (2002) Solid modeling. *Handbook Comput Aided Geom Des* 20:473–518
- Shapiro V (2007) Semi-analytic geometry with r-functions. *ACTA numerica* 16:239–303
- Struik DJ (2012) Lectures on classical differential geometry. Courier Corporation
- Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. *Int J Numer Methods Eng* 24(2):359–373
- Svanberg K (2007) Mma and gcmma, versions september 2007. *Optim Syst Theory*, 104
- Timoshenko S (1953) History of strength of materials: with a brief account of the history of theory of elasticity and theory of structures. Courier Corporation
- Wang F, Jensen JS, Sigmund O (2012) High-performance slow light photonic crystal waveguides with topology optimized or circular-hole based material layouts. *Photon Nanostruct-Fund Appl* 10(4):378–388
- Wein F, Stingl M (2018) A combined parametric shape optimization and ersatz material approach. *Struct Multidiscip Optim* 57(3):1297–1315
- Xia L, Zhu J, Zhang W (2012) Sensitivity analysis with the modified heaviside function for the optimal layout design of multi-component systems. *Comput Methods Appl Mech Eng* 241:142–154
- Zhang W, Xia L, Zhu J, Zhang Q (2011) Some recent advances in the integrated layout design of multicomponent systems. *J Mech Des* 133(10):104,503
- Zhang W, Zhong W, Guo X (2015) Explicit layout control in optimal design of structural systems with multiple embedding components. *Comput Methods Appl Mech Eng* 290:290–313

- Zhang S, Norato JA, Gain AL, Lyu N (2016a) A geometry projection method for the topology optimization of plate structures. *Struct Multidiscip Optim* 54(5):1173–1190
- Zhang W, Yuan J, Zhang J, Guo X (2016b) A new topology optimization approach based on moving morphable components (mmc) and the ersatz material model. *Struct Multidiscip Optim* 53(6):1243–1260
- Zhang S, Gain AL, Norato JA (2017a) Stress-based topology optimization with discrete geometric components. *Comput Methods Appl Mech Eng* 325:1–21
- Zhang W, Li D, Yuan J, Song J, Guo X (2017b) A new three-dimensional topology optimization method based on moving morphable components (mmcs). *Comput Mech* 59(4):647–665
- Zhou M, Wang MY (2013) Engineering feature design for level set based structural optimization. *Comput Aided Des* 45(12):1524–1537
- Zhu J, Zhang W, Beckers P, Chen Y, Guo Z (2008) Simultaneous design of components layout and supporting structures using coupled shape and topology optimization technique. *Struct Multidiscip Optim* 36(1):29–41