



Minimum length-scale constraints for parameterized implicit function based topology optimization

Peter D. Dunning¹

Received: 25 August 2017 / Revised: 23 November 2017 / Accepted: 10 December 2017 / Published online: 12 January 2018
© The Author(s) 2018. This article is an open access publication

Abstract

This paper introduces explicit minimum length-scale constraint functions suitable for parameterized implicit function based topology optimization methods. Length-scale control in topology optimization has many potential benefits, such as removing numerical artifacts, mesh independent solutions, avoiding thin, or single node hinges in compliant mechanism design and meeting manufacturing constraints. Several methods have been developed to control length-scale when using density-based or signed-distance-based level-set methods. In this paper a method is introduced to control length-scale for parameterized implicit function based topology optimization. Explicit constraint functions to control the minimum length in the structure and void regions are proposed and implementation issues explored in detail. Several examples are presented to show the efficacy of the proposed method. The examples demonstrate that the method can simultaneously control minimum structure and void length-scale, design hinge free compliant mechanisms and control minimum length-scale for three dimensional structures.

Keywords Length-scale constraints · Topology optimization · Implicit function · Parameterization

1 Introduction

Length-scale control in topology optimization has been a topic of interest since the early days of its development, almost 30 years ago (Bendsøe and Kikuchi 1988). Initially, length-scale control was part of methods devised to solve numerical issues, such as mesh dependent solutions and checker-board patterns (Sigmund and Petersson 1998), and to avoid thin or single node hinges in compliant mechanism design problems (Luo et al. 2008b; Sigmund 2009). Researchers have also applied length-scale control methods to obtain manufacturable designs (Liu et al. 2015; Allaire et al. 2016; Vatanabe et al. 2016; Lazarov et al. 2016). Another potential benefit of minimum length-scale control is the implicit improvement in stress and buckling performance. Minimum void length-scale promotes rounded internal corners, hence implicitly reduces stress concentrations. Whereas, increasing the minimum

structural length-scale results in thicker members that have a higher buckling resistance.

There are now several different approaches to topology optimization (Sigmund and Maute 2013; Deaton and Grandhi 2014), including density-based methods (Bendsøe and Sigmund 2004), Evolutionary Structural Optimization (ESO) (Munk et al. 2015) and level-set methods (van Dijk et al. 2013). Density-based and ESO methods utilize element-wise design variables to indicate where material exists. These methods have been successfully applied to solve many problems. However, one of the main drawbacks of element-wise design variables is that a smooth boundary is not available throughout the optimization. This can be important for certain types of problem, for example problems involving fluid-structure interaction (Jenkins and Maute 2016).

Thus, alternative methods have been proposed where the exact position of the boundary is known throughout the optimization. The most popular of these is the level-set method, where the boundary is describe by the zero level-set of an implicit signed-distance function. Although the zero-level set is usually only approximately extracted from the discretization of the implicit function. The traditional level-set optimization method uses shape derivatives to define a velocity function that is used to update the boundary by

✉ Peter D. Dunning
peter.dunning@abdn.ac.uk

¹ School of Engineering, University of Aberdeen,
Aberdeen AB24 3UE, UK

solving a Hamilton-Jacobi type PDE (Wang et al. 2003; Allaire et al. 2004). The traditional approach can be slow to converge (Luo et al. 2008a; van Dijk et al. 2013) and requires additional techniques to handle constraints beyond a simple volume constraint (Dunning and Kim 2015).

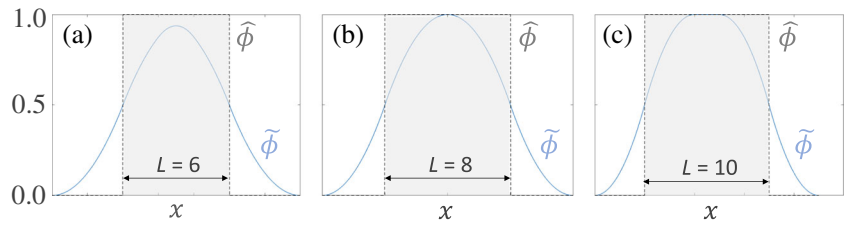
To address the drawbacks of the traditional level-set approach, several methods have been proposed to parameterize the implicit function such that standard optimization methods (e.g. nonlinear programming) can be used to efficiently solve problems with arbitrary nonlinear constraints. In this paper, these methods are referred to as parameterized implicit function based topology optimization. Several different parameterization schemes have been proposed. A popular approach is to use radial basis functions (Norato et al. 2004; De Ruiter and Van Keulen 2004; Luo et al. 2007; Pingen et al. 2010). Other parameterization methods are the spectral level-set method (Gomes and Suleman 2006), Kriging-based (Hamza et al. 2014) and an approach based on the intersection of a cutting plane and an implicit signed-distance function, recently proposed by the author (Dunning 2017).

Some of the main techniques created for length-scale control in density-based methods are: introducing a perimeter constraint, local density gradient constraints and filtering (Sigmund and Petersson 1998; Liu and Ma 2016). The perimeter constraint cannot explicitly control length-scale and the local density gradient constraint method adds a large number of constraints to the problem. It was also noted that the original filtering methods do not explicitly control length-scale and thin hinges in compliant mechanisms can still occur. Thus, more advanced filtering methods have been developed to enable explicit length-scale control. Sigmund (2009) introduced the so-called robust topology optimization method based on dilation, erode and combinations filters to explicitly control length-scale. However, this is only guaranteed if the dilated and eroded designs have the same topology as the original, which is not always the case (Wang et al. 2011). Furthermore, it is computationally expensive, as it requires multiple finite element analyses each iteration. Density projection schemes, which are similar to filtering methods, can also control length-scale in the solid or void regions by restricting the minimum radius of the projection (Guest et al. 2004). A multiphase project method can be used to control minimum length-scale in both solid and void regions (Guest 2009). To avoid the computational cost of the robust formulation, Zhou et al. (2015) introduced explicit constraint formulations for minimum length-scale in both solid and void regions for a three-field density-based method. An alternative to filtering is the “MOLE” method (Poulsen 2003), where an explicit length-scale constraint is formulated by checking the monotonicity of the density gradient along certain directions for each element.

For the level-set methods that utilize a signed-distance function, techniques have been created to control length-scale by exploiting the geometric information contained within the signed-distance function. One approach is to set constraints on the maximum and minimum value of the signed-distance function within a domain around the medial axis (essentially the set points that have at least two closest boundary points) (Guo et al. 2014b). This method has also been extended and applied to density-based methods, which imposes constraints on the maximum and minimum density values within the medial axis domain (Zhang et al. 2014). The method proposed by Xia and Shi (2015), also used the medial axis (referred to as the ‘skeleton’) to control length-scale by constraining the distance from the boundary to the skeleton. Wang et al. (2016) formulated length-scale constraints by comparing the geometry and level-set offset areas, again supported by the signed-distance function. The structural skeleton idea is also used by Zhang et al. (2017c) to control minimum hole size. Allaire et al. (2016) proposed thickness constraints by constructing penalty function integrals of the signed distance function. A thickness-control functional based on the signed-distance function was introduced by Liu et al. (2015) to promote uniform member thickness, which is important for parts made by injection molding. An alternative to utilizing the signed-distance function is to add a fictitious quadratic energy functional to the objective function in order to control features in the level-set method (Luo et al. 2008b; Chen et al. 2008). However, this approach cannot satisfy the minimum structural size to an exact value.

Several methods have recently been developed that utilize components, or voids, to parameterize the design (Guo et al. 2014a; Zhang et al. 2016a, 2017a, b, d). These methods allow the exact position of the boundary to be known throughout the optimization. They also allow easier control of length-scale by adding simple geometric constraints to feature dimensions (Zhang et al. 2016b; Hoang and Jang 2017). These methods are not considered as parameterized implicit function methods, which is the focus of this paper, because the parameterization is of discrete components or voids and therefore they do not directly parameterize an implicit function. Also, Chen et al. (2007) proposed a method that combines components and void features with a spline-based parameterized implicit function to perform parametric shape and topology optimization. This method has some control over length-scale, as the maximum and minimum size of primitive features can be explicitly defined through geometric constraints. It is interesting to note that Liu et al. (2016) define two different types of void: interior voids and boundary voids. This distinction is useful in the context of minimum void scale for machining constraints (i.e.

Fig. 1 Filtering a discrete feature with $R = 4$: **a** $L < 2R$, **b** $L = 2R$, **c** $L > 2R$



minimum tool size). In this paper, minimum length-scale constraints are applied to both interior and boundary voids.

In summary, there have been many schemes developed to explicitly control length-scale in density-based methods, level-set methods that utilize a signed distance function and methods based on moving components, or voids. However, the author is unaware of any general scheme to explicitly control minimum-length scale for parameterized implicit function based topology optimization methods. Therefore, this paper introduces a new method that can control the minimum length-scale, in both solid and void regions, for parameterized implicit function based topology optimization. The formulation of the length-scale constraints is introduced in Section 2. Numerical issues and implementation details are presented in Section 3. The parameterization scheme used in this paper is briefly reviewed in Section 4, followed by examples in Section 5 and finally conclusions.

2 Length-scale constraint formulation

2.1 Fundamental idea: filtering a discrete feature

The approach to topology optimization studied in this paper utilizes an implicit function $\phi(x)$ to define the structure and void regions:

$$\begin{cases} \phi(x) > 0 \Leftrightarrow x \in \Omega, \\ \phi(x) = 0 \Leftrightarrow x \in \Gamma, \\ \phi(x) < 0 \Leftrightarrow x \notin \Omega, \end{cases} \quad (1)$$

where $\phi(x)$ is the implicit function, x is a point in the design domain (Ω_D), Ω and Γ are the structural domain and boundary, respectively. The design domain and implicit function are usually discretized on a fixed regular grid with edge length, h . The implicit function is then parameterized such that the values of $\phi(x)$ at set locations (usually nodes of the fixed grid) are explicitly linked to a set of design variables \mathbf{b} : $\phi(\mathbf{b})$.

The first step in building the length-scale constraint functions is to normalize the implicit function such that it takes values between 0 and 1. The binary case is then:

$$\hat{\phi}(x) = \begin{cases} 1 & x \in \Omega, \\ 0 & x \notin \Omega. \end{cases} \quad (2)$$

A linear filter with radius, R , equal to half the desired minimum length-scale is then applied to the normalized implicit function:

$$\tilde{\phi}(x) = \frac{\sum_{i \in \mathbb{H}} w(x_i) \hat{\phi}_i}{\sum_{i \in \mathbb{H}} w(x_i)}, \quad w(x_i) = R - |x - x_i|, \quad (3)$$

where \mathbb{H} is the neighborhood of points lying within the filter radius of point x .

The filtered function, (3), can be used to identify discrete features with a size less than the minimum desired length-scale. To illustrate this, a simple one-dimensional example is shown in Fig. 1, where the shaded area shows a discrete feature with width L and the solid line is the filtered function. It can be seen that when $L < 2R$, then $\max(\tilde{\phi}) < 1$, otherwise if $L \geq 2R$, then $\max(\tilde{\phi}) = 1$. Therefore, for structural features, the length-scale is violated if the maximum of the filtered function is less than 1. For length-scale in the void region a similar observation can be made: if $L < 2R$, then $\min(\tilde{\phi}) > 0$, otherwise if $L \geq 2R$, then $\min(\tilde{\phi}) = 0$. So the minimum void length-scale is violated if the minimum of the filtered function is greater than zero.

2.2 Indicator function

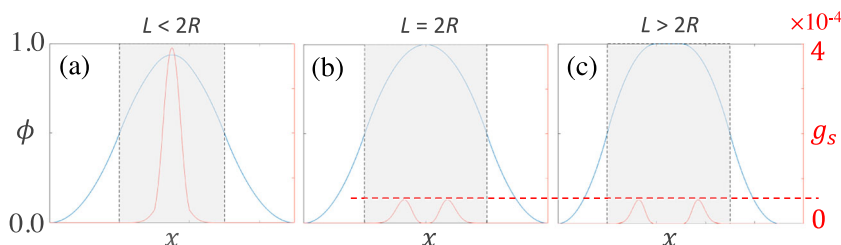
To build the length-scale constraint functions we need to identify locations (nodes) where the filter function should be measured. According to the observations in Section 2.1 this should be where the filter function is a maximum or minimum, i.e. $\nabla \tilde{\phi} = 0$. To achieve this we use the indicator function idea introduced by Zhou et al. (2015):

$$I_s = \hat{\phi}_i \exp\left(-c \|\nabla \tilde{\phi}_i\|^2\right) \quad (4)$$

$$I_v = \left(1 - \hat{\phi}_i\right) \exp\left(-c \|\nabla \tilde{\phi}_i\|^2\right) \quad (5)$$

where I_s and I_v are the structure and void indicator functions, respectively, and c is an exponential decay constant that is set with respect to the normalized filter radius: $r = R/h$, where h is the fixed grid edge length. The value suggested by Zhou et al. (2015) is $c = r^4$, which allows the inflection region of the filter function ($\|\nabla \tilde{\phi}_i\| \approx 0$) to be captured accurately, while accounting for the effect of discretization. This value was also found to be suitable for the application in this paper. It is important to note that because the decay constant, c , is calculated based on a normalized filter radius, the finite difference scheme

Fig. 2 Constraint values for a feature with length: **a** $L < 2R$, **b** $L = 2R$, **c** $L > 2R$



used to compute the spatial gradient of the filter function assumes a normalized grid spacing of $h = 1$. Thus, in 2D, the spatial gradient magnitude is computed as:

$$\|\nabla\tilde{\phi}_i\|^2 = 0.25 \left((\tilde{\phi}_{i+x} - \tilde{\phi}_{i-x})^2 + (\tilde{\phi}_{i+y} - \tilde{\phi}_{i-y})^2 \right) \quad (6)$$

where $i + x, i - x, i + y$ and $i - y$ are the grid points to the right, left, above and below point i , respectively.

2.3 Constraint functions

The constraint functions for minimum length-scale in the structure and void regions are formulated by multiplying the appropriate indicator function, (4) or (5), respectively, by a measure on the difference between the filter function and the ideal value if the length-scale is satisfied (i.e. 1 for structure and 0 for void). This value is calculated and summed over all locations in the domain where the implicit function is specified (i.e. at the grid nodes):

$$g_s = \sum_{i=1}^N (1 - \tilde{\phi}_i)^2 \hat{\phi}_i \exp(-c\|\nabla\tilde{\phi}_i\|^2) \leq \epsilon_s \quad (7)$$

$$g_v = \sum_{i=1}^N \tilde{\phi}_i^2 (1 - \hat{\phi}_i) \exp(-c\|\nabla\tilde{\phi}_i\|^2) \leq \epsilon_v \quad (8)$$

where N is the number of nodes in the domain and ϵ_s and ϵ_v are the constraint tolerances. See Section 3.1 for further discussion on setting tolerance values.

These constraint functions are similar to those proposed by Zhou et al. (2015) for density-based topology optimization. However, they have been adapted here for use with parameterized implicit function based topology optimization methods. The main difference is that the Zhou et al method was developed for a three-field density-based topology optimization method, which uses a filtered density field that is then passed through a threshold function. The filtered and threshold fields are directly used in the constraint formulation. Here, the filter function is only required to formulate the length-scale constraints and is not otherwise required. This feature is useful, as it allows the proposed constraints to be added to any parameterized implicit function based method, without having to change the original formulation.

3 Numerical issues and implementation

First, the normalized implicit function, (2), is approximated by a smooth function to enable differentiation:

$$\hat{\phi} = \begin{cases} 0, & \phi < -\Delta, \\ \frac{3}{4} \left(\frac{\phi}{\Delta} - \frac{\phi^3}{3\Delta^3} \right) + \frac{1}{2}, & -\Delta \leq \phi < \Delta, \\ 1, & \Delta \leq \phi, \end{cases} \quad (9)$$

where Δ is the smoothing length of the approximation. A value of $\Delta = 2h$ was chosen based on numerical experience.

3.1 Constraint tolerance

The constraint tolerance values, ϵ in (7, 8), cannot be set to zero because the indicator functions, (4, 5), are nonzero over a domain where the filter function would indicate that the length-scale is not satisfied. To illustrate this, the point-wise values of the constraint function are plotted in Fig. 2 for the analytical 1D example from Fig. 1. This shows that when the length-scale is satisfied (i.e. $L \geq 2R$) then the maximum value of the point-wise constraint function does not decrease with increasing values of L . For the 1D analytical example this maximum value can be calculated as: $\delta_g = r^{-4} \cdot \exp(-2)$ (see Appendix A). Therefore, if a nodal value of the constraint function is greater than this maximum value (i.e. $g_{s,i} > \delta_g$), then the length-scale in the local region is expected to be violated and $g_{s,i}$ should be reduced. On the other hand, if it is equal to or below the maximum value ($g_{s,i} \leq \delta_g$), then this value of $g_{s,i}$ does not need to be reduced. This observation provides a way

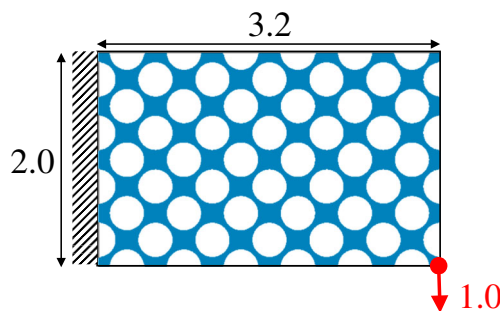
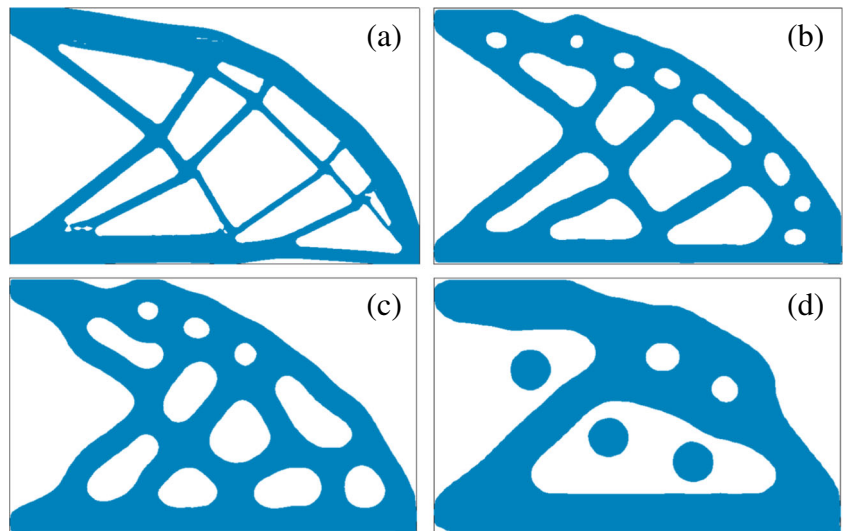


Fig. 3 Cantilever example

Fig. 4 Cantilever solutions with full sensitivity: **a** no length-scale constraint, and minimum structure length of, **b** 0.15, **c** 0.20 and **d** 0.30



of automatically setting an appropriate constraint tolerance, without the need of user selected parameters:

$$\epsilon_s = \begin{cases} g_s - \sum_{i=1}^N \max(g_{s,i} - \delta_g, 0), & L_{\min} < 2R, \\ 1.5g_s, & L_{\min} \geq 2R, \end{cases} \quad (10)$$

where:

$$g_{s,i} = (1 - \tilde{\phi}_i)^2 \hat{\phi}_i \exp(-c \|\nabla \tilde{\phi}_i\|^2) \quad (11)$$

and L_{\min} is the current minimum length-scale.

There are two additional issues to discuss. The first issue is the second condition in (10), where the constraint tolerance is relaxed if the length-scale constraint is satisfied. This was found necessary because it was observed that the length-scale constraint often prevented useful topology

changes (by removing members or closing holes) because this would increase and violate the constraint function. Thus, a relaxed constraint tolerance is used if the length-scale constraint is satisfied. This scheme enables the possibility of escaping from the local minima created by the length-scale constraint. The value of 1.5 in (10) was chosen based on numerical experience.

The second issue is that the constraint tolerance, as calculated by (10), will change every iteration, as the values of the constraint function change. If ϵ is updated every iteration, this may cause convergence problems. Also, the procedure used to calculate L_{\min} requires that a signed-distance function is created (see Section 3.2). Performing this calculation every iteration adds a noticeable computational overhead. Therefore, constraint tolerance values are only updated periodically. In this work, they are recomputed every 10 iterations.

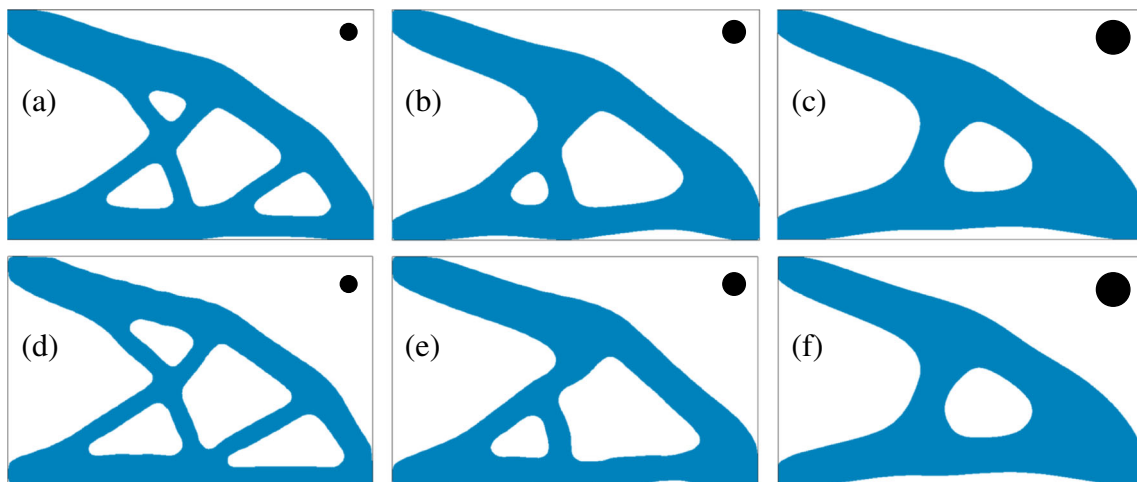


Fig. 5 Cantilever solutions. Using approximate sensitivities: minimum structure length of: **a** 0.15, **b** 0.20 and **c** 0.30. Using the hybrid scheme: minimum structure length of: **d** 0.15, **e** 0.20 and **f** 0.30

Table 1 Cantilever example solution data

Min. length	Full sensitivity		Approx. sensitivity		Hybrid sensitivity	
	Volume	Iteration	Volume	Iteration	Volume	Iteration
0.15	2.75	980	2.64	820	2.41	1225
0.20	3.05	850	2.67	850	2.50	968
0.30	3.37	1200	2.78	750	2.78	752

3.2 Estimating the minimum length

The constraint tolerance strategy introduced in the previous section requires the minimum length, L_{\min} , to be computed automatically. To achieve this we first construct a signed-distance function, ϕ^{sd} from the current implicit function using the fast marching method (Sethian 1999). Next, we adapt the indicator functions, (4, 5), to identify inflection points of the signed-distance function, as this will approximately identify points lying at the center of structural members or holes:

$$I_{sd,s} = H(\phi_i^{sd}) \exp(-\|\nabla\phi_i^{sd}\|^2) \quad (12)$$

$$I_{sd,v} = (1 - H(\phi_i^{sd})) \exp(-\|\nabla\phi_i^{sd}\|^2) \quad (13)$$

Using these indicator functions, we find the set of points: $\mathbb{S} = \{x | I_{sd}(x) > 0.6\}$. This set is then reduced using a trim function to eliminate meaningless points (see Xia and Shi (2015) for a detailed discussion on this issue):

$$\mathbb{I} = \left\{ x \mid 1 - \frac{\nabla^b \phi^{sd}(x) \cdot \nabla^f \phi^{sd}(x)}{|\nabla^b \phi^{sd}(x)| |\nabla^f \phi^{sd}(x)|} > 1.2, x \in \mathbb{S} \right\} \quad (14)$$

where ∇^b and ∇^f are forward and backward finite differences, respectively. Finally, the set \mathbb{I} is further reduced by eliminating any points within $2R$ of the domain corners.

The estimate for L_{\min} in the structure domain is then the minimum value of the signed distance function for the final set of points. For the void region, the maximum value of the signed distance function for the final set of points is used to estimate L_{\min} . This procedure for computing L_{\min} was numerically validated and found to underestimate the true

value by approximately $h/2$. This is reasonable, because the numerical discretization may not exactly capture the value of the signed distance function corresponding to the minimum length. Therefore, a value of $h/2$ is added to the estimate computed by the above procedure to obtain a more accurate value for L_{\min} .

3.3 Sensitivity information

The derivatives of the length-scale constraint functions are shown in Appendix B. These analytical formulae were validated against finite differences and found to be accurate. However, it is observed that unsatisfactory solutions are often obtained using the full, accurate sensitivity information throughout the optimization. The optimizer used is the Method of Moving Asymptotes (MMA) as implemented in the package nlopt (Johnson 2014).

To illustrate this issue, the solutions of a cantilever example (Fig. 3) using the full sensitivities are shown in Fig. 4. The example is taken from Allaire et al. (2016), where the objective is to minimize volume, subject to a compliance constraint of 60 units. The Young's modulus is 1.0 and Poisson's ratio is 0.3. The design domain is discretized using 160×80 bilinear plane stress elements. A total of 2665 (65×41) design variables are used to parameterize the implicit function (see Section 4). The problem is solved with three different minimum length-scale constraints on the structure. It can be seen in Fig. 4 that although the length-scale constraint is satisfied, the solutions are intuitively far from optimal. This is especially noticeable for the solution with a minimum length of 0.3, where material islands exist in the solution.

A numerical investigation revealed that the sensitivity information for the spatial gradient of the filtered function ($dg_s/d(\nabla\tilde{\phi})$) can prevent the optimizer from closing holes and removing material islands. Two schemes are investigated that aim to avoid the convergence issues introduced by the full sensitivities. The first scheme uses approximate sensitivities for the length-scale constraint functions, by assuming: $dg_s/d(\nabla\tilde{\phi}) = 0$. However, when approximate sensitivities are used it cannot be guaranteed that the method finds a local minima. Therefore, a hybrid

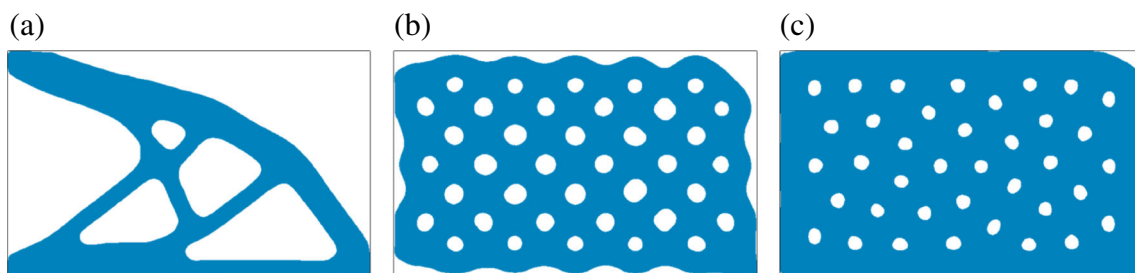


Fig. 6 Cantilever solutions with length-scale constraints active from the start. Minimum structure length of: **a** 0.15, **b** 0.20 and **c** 0.30

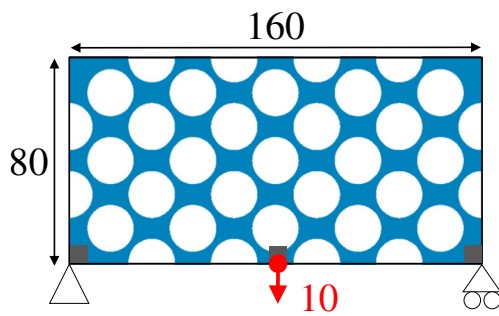


Fig. 7 Michell structure example

scheme is also proposed, whereby a solution is first found using the approximate sensitivities, then the full sensitivities are used to ensure a local minima is found.

Solutions for the cantilever example using both schemes are shown in Fig. 5. Note that the black circle in the top right of solution figures indicates the minimum structure length. Visually, these solutions appear superior, which is confirmed by comparing the objective function values in Table 1. Furthermore, it can be seen that, although the hybrid scheme adds computational cost in increased number of iterations, the use of full sensitivities generally helps improve the objective function. However, this is not always the case, as can be seen for minimum length-scale of 0.30, where no improvement is gained when using the hybrid scheme.

Furthermore, it is interesting to note that approximate sensitivities are used in other length-scale constraint methods (Xia and Shi 2015; Allaire et al. 2016). In particular Xia and Shi (2015) ignored the dependency of the

position of the skeleton on the boundary position. This has a similar effect to the proposed approximate derivative in this paper, as it ignores most of the dependency of the indicator function on the design variables.

3.4 Constraint activation

The final implementation issue is to decide when to activate the length-scale constraints. Two schemes are proposed. The first is to simply include the length-scale constraints from the start. The second idea is to activate them after a solution to the original problem (without length-scale constraints) is obtained.

Numerical testing shows that the first strategy is not suitable, as the length-scale constraints can quickly trap the design in a local minima close to the initial design. This occurs despite the relaxed constraint tolerance technique introduced in Section 3.1. To highlight this, the cantilever problem (Fig. 3) is solved with length-constraints active from the start and the solutions are shown in Fig. 6. Although the solution for a minimum length-scale of 0.15 is reasonable, it is obvious that the solutions for minimum length-scales of 0.2 and 0.3 are stuck in local minima close to the initial design. Note that the solutions obtained in the previous section (Figs. 4 and 5) are obtained using the second strategy where the length-scale constraint is activated after the original problem converges. This type of approach, where explicit length-scale constraints are activated part way through the optimization process is also used in other methods, see for example Zhou et al. (2015), for a density-based example, and Allaire et al. (2016), for a level-set based example.

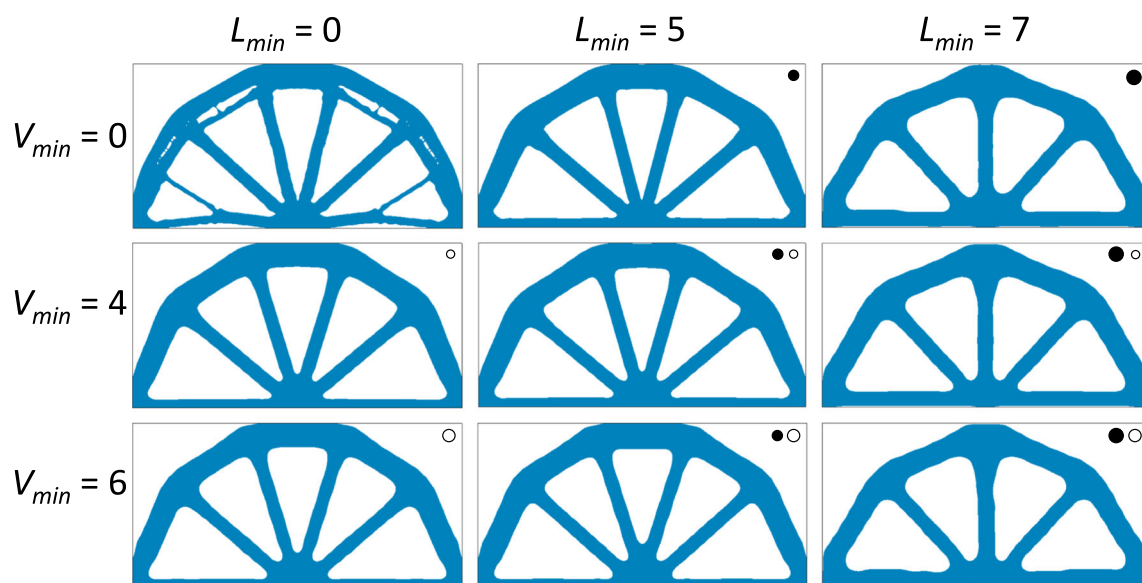


Fig. 8 Michell structure solutions with different structure and void length-scale constraints

Table 2 Michell structure example solution data

Minimum length			
Structure	Void	Compliance	Iteration
None	None	1670	100
5	None	1665	440
7	None	1728	1250
None	4	1665	420
5	4	1665	700
7	4	1718	990
None	6	1670	450
5	6	1672	430
7	6	1754	840

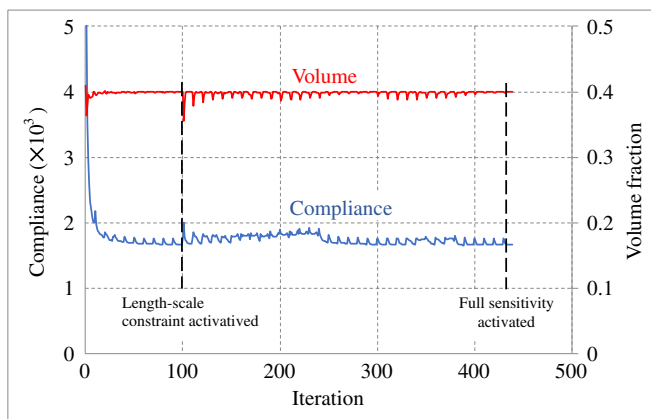
4 Parameterization method

The implicit function parameterization method used in this paper is based on the idea of defining the structure boundary as the intersection of a signed-distance implicit function (created from the initial boundary), α , and a “cutting surface”, β , (Dunning 2017). The cutting surface is parameterized by the design variables, \mathbf{b} , such that the implicit function, ϕ in (1) becomes an explicit function of the design variables:

$$\phi(x, \mathbf{b}) = \alpha(x) - \beta(x, \mathbf{b}). \tag{15}$$

In the original method, the cutting surface was parameterized using finite element shape functions. Here, it is parameterized using spline basis functions. A number of control points are evenly spaced over the design domain in each coordinate direction. The value of the cutting surface at any point (such as the grid nodes) is calculated from the design variables defined at the control points and the spline basis functions. Using this parameterization, (15) is now written in a discrete form:

$$\phi(\mathbf{b}) = \alpha - \mathbf{A} \cdot \mathbf{b} \tag{16}$$



where ϕ and α are vectors of implicit function values defined at the nodes of the fixed grid. The values in the matrix \mathbf{A} are calculated using spline basis functions:

$$A_{i,j} = N_{j,k}(x_i)M_{j,k}(y_i) \tag{17}$$

where (x_i, y_i) are the coordinates of grid node i , $N_{j,k}$ and $M_{j,k}$ are the k^{th} order spline basis functions for the control point spacing in the x and y directions, respectively (see, for example, Rogers 2000). In this paper 3rd order ($k = 3$) spline basis functions are used. Note that the matrix \mathbf{A} remains fixed throughout the optimization and is only calculated once at the start.

The use of a fixed signed-distance function in the parameterization provides some regularization, but also restricts the design space. Therefore, it is periodically updated during the optimization using the current design boundary ($\phi = 0$). This leads to an optimization process composed of a series of inner-loop problems defined as:

$$\begin{aligned} \min_{\mathbf{b}} f(\phi(\mathbf{b})) \\ \text{s.t. } g_i(\phi(\mathbf{b})) \leq 0, \quad i = 1 \dots m \\ \mathbf{b}_{min} \leq \mathbf{b} \leq \mathbf{b}_{max} \end{aligned} \tag{18}$$

where f is the objective function and g_i are the constraint functions, which can include the length-scale constraint functions introduced in Section 2.3.

The signed-distance function, α remains fixed during each inner-loop optimization and is updated between inner-loop optimizations using the fast marching method (Sethian 1999). The maximum number of iterations for each inner-loop is set to 10. The inner-loop problem is solved using MMA, as implemented in the package nlopt (Johnson 2014). For further details on this parameterization method and its implementation, see Dunning (2017).

For this particular parameterization method, the update of the signed-distance function is conveniently timed to coincide with the required generation of a signed-distance function to estimate the current minimum length, as detailed

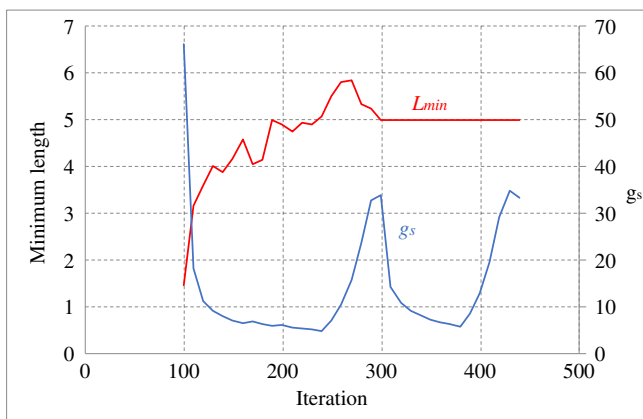


Fig. 9 Michell structure convergence history, with minimum structure length = 5

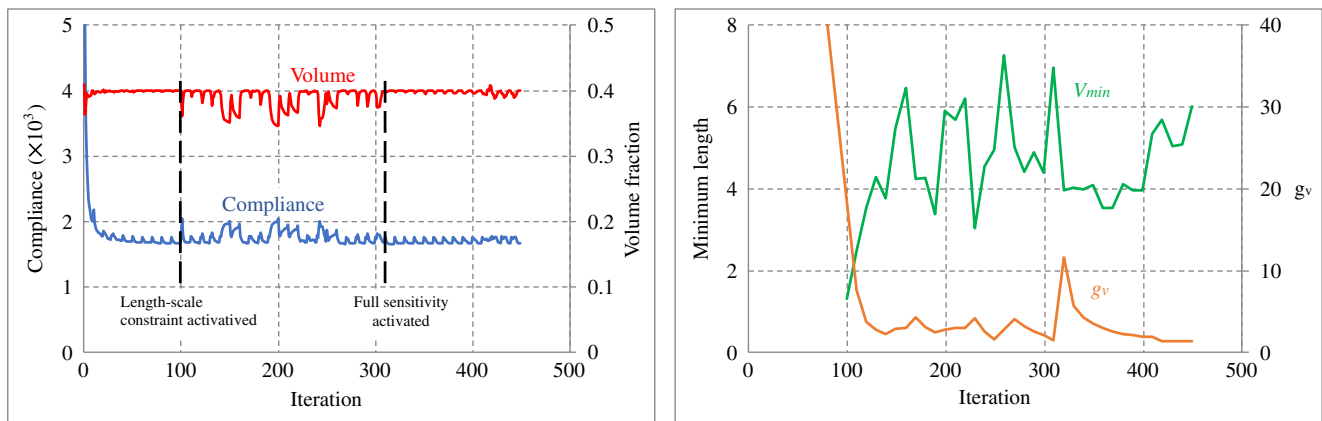


Fig. 10 Michell structure convergence history, with minimum void length = 6

in Section 3.2. Thus, the number of times a signed-distance function is created is reduced, which improves computational efficiency.

A fixed grid finite element approach is used, where the analysis discretization coincides with the discretization of the implicit function. The stiffness and mass properties of elements are computed using the smoothed Heaviside method. For more details on how this approach is implemented with the parameterization scheme, please see Dunning (2017). For the examples in this paper, the smoothing length is set to one element edge length.

5 Examples

5.1 Michell structure

The length-scale constraint functions are applied to a classic Michell arch benchmark structure, Fig. 7. The shaded areas at the corners indicate areas fixed to remain part of the structure domain. The design domain is discretized using 160×80 bilinear plane stress elements. Young’s modulus is 1.0 and Poisson’s ratio is 0.3. The implicit function

is parameterized using 41×21 control points, giving a total of 861 design variables. The objective is to minimize compliance with an upper limit on volume, set to 40% of the design domain.

The problem is solved several times using different structure length-scale (L_{min}) and void length-scale (V_{min}) constraints. The solutions are shown in Fig. 8 and objective function values are shown in Table 2. The black and white circles at the top right of the solutions indicate the minimum structure and void length-scale, respectively. It can be seen that as L_{min} is increased from 5 to 7, the number of holes reduces, allowing fewer and thicker structural members, which also increases the objective function. Another observation is that when a minimum void length-scale is imposed the solutions have slightly more rounded internal corners, especially near the lower central joint where the load is applied. These results show that the proposed method can effectively impose a minimum length-scale in the structure and void regions, independently and simultaneously.

Convergence histories for 3 examples are shown in Figs. 9, 10 and 11. Other examples have similar convergence

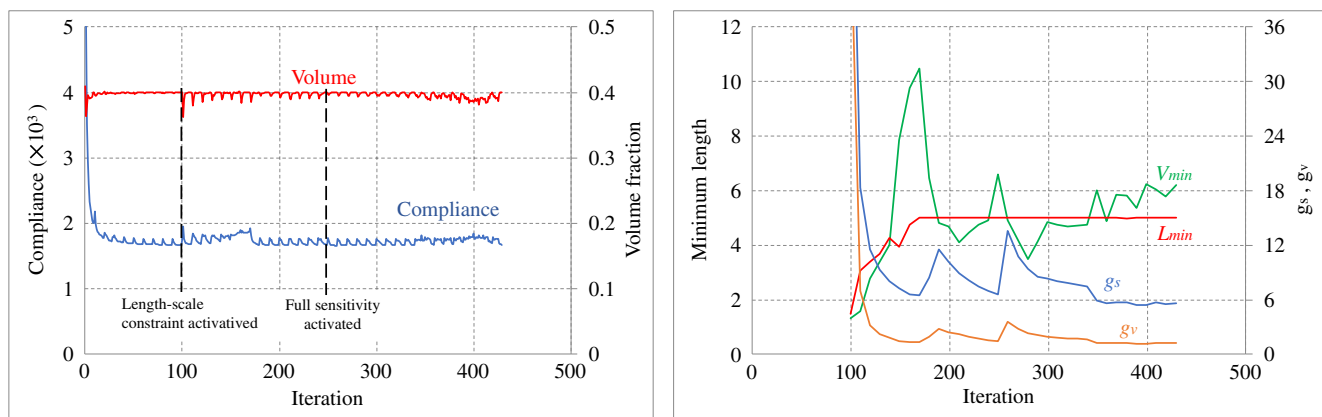


Fig. 11 Michell structure convergence history, with minimum structure length = 5 and minimum void length = 6

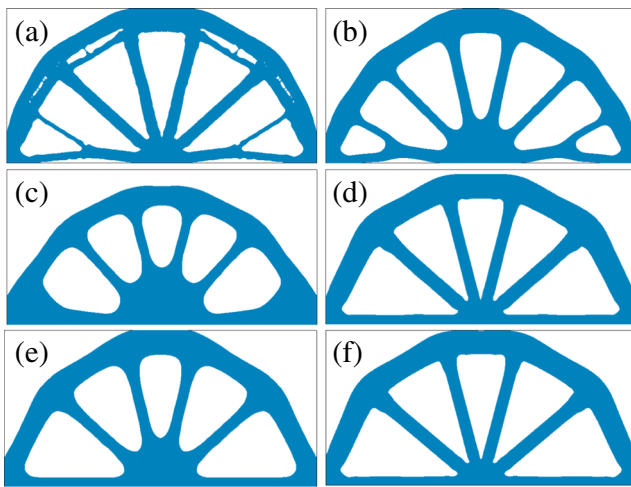


Fig. 12 Michell structure with minimum structure length = 5. Design evolution: **a** initial convergence without length-scale constraints (iteration 100), **b** iteration 150, **c** iteration 230, **d** iteration 300, **e** iteration 380, and **f** solution at iteration 440

behavior and are omitted for brevity. The convergence of the compliance and volume fraction is shown alongside the convergence of the minimum length-scale constraints. The values of g_s and/or g_v , (7, 8), are shown with the actual computed minimum length-scale (using the method detailed in Section 3.2). However, data for constraint convergence is only shown at the end of each inner-loop (every 10 iterations), as the minimum length-scale is only explicitly computed at these times.

It is noticed that the convergence of compliance and volume is oscillatory. This is caused by the reinitialization of the implicit function after every inner-loop (10 iterations). The oscillations occur both before and after the length-scale constraints are activated. This behavior is caused by the chosen parameterization method and not the formulation, or implementation of the length-scale constraint method. The convergence of the length-scale constraint functions show the effect of the scheme employed to set the

constraint tolerance, proposed in Section 3.1. When the desired length-scale is found to be satisfied, the value of the constraint function tends to increase, which is the effect of the relaxation condition in (10). Finally, it can be seen that all length-scale constraints are satisfied at the end of the optimization.

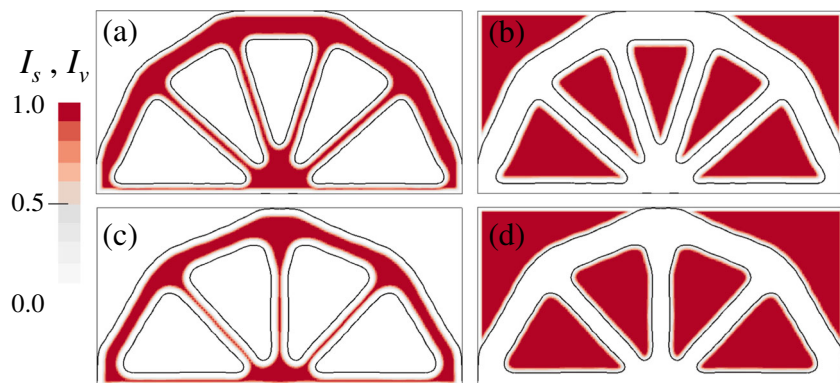
To illustrate how a design typically evolves when the length-scale constraints are active, the case with a minimum structure length of 5 is shown in Fig. 12. It can be seen that, first, the small holes in the top of the arch are closed, which is an efficient way of increasing the minimum structure length in that region. Next, the larger holes at the extreme left and right are closed and the final topology is found, before further iterations of essentially shape optimization to produce the final design. Indicator functions, (4, 5), are shown in Fig. 13 for two solutions that have both structure and void length-scale constraints applied.

5.2 Compliant mechanism

Another motivation for imposing length-scale control is to avoid thin, or single node hinges when designing compliant mechanisms by topology optimization. The classic inverter mechanism example is used to demonstrate the ability of the method proposed in this paper to achieve this goal. The specification for the inverter is taken from Allaire et al. (2016) and shown in Fig. 14. The shaded areas at the corners indicate areas fixed to remain part of the structure domain. The objective is to maximize the displacement at the output location, u_{out} , subject to an upper limit on volume, set to 30% of the design domain. Using symmetry, only the top half of the mechanism is modeled (as shown in Fig. 14). The design domain is discretized using 160×80 bilinear plane stress elements. Young's modulus is 1.0 and Poisson's ratio is 0.3. The implicit function is parameterized using 41×21 control points, giving a total of 861 design variables.

The problem is solved several times with different minimum structure length constraints. Solutions are shown in Fig. 15 and objective function values in Table 3. The solution without a length-scale constraint has the familiar

Fig. 13 Indicator functions for Michell structure. Solution with $L_{min} = 5$, $V_{min} = 6$, indicator for: **a** structure, **b** void. Solution with $L_{min} = 7$, $V_{min} = 4$, indicator for: **c** structure, **d** void



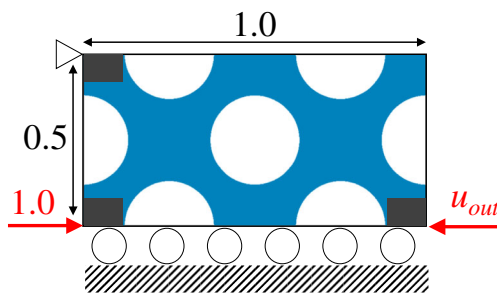


Fig. 14 Inverter mechanism example

thin hinge regions that may be difficult to manufacture in a robust way (Sigmund 2009). The results show the expected behavior that increasing the minimum structure length-scale increases the thickness in the hinge regions, which reduces the performance (smaller u_{out}) of the mechanism. Thus, the proposed minimum structure length-scale constraint can produce hinge-free compliant mechanism designs.

5.3 Three dimensional example

The ability of the proposed method to control length-scale for three-dimensional structures is investigated using an example adapted from Kennedy (2015). The problem is shown in Fig. 16 and the objective is to minimize compliance, subject to an upper limit on volume, set to 30% of the design domain. Young’s modulus is 100.0 and Poisson’s ratio is 0.3. The design domain is discretized using $50 \times 50 \times 100$ trilinear elements. The implicit function is parameterized using $26 \times 26 \times 51$ control points, giving a total of 34,476 design variables. The problem is solved with a minimum structure length of 0.5. The solutions with and without the length-scale constraint are shown in Fig. 17.

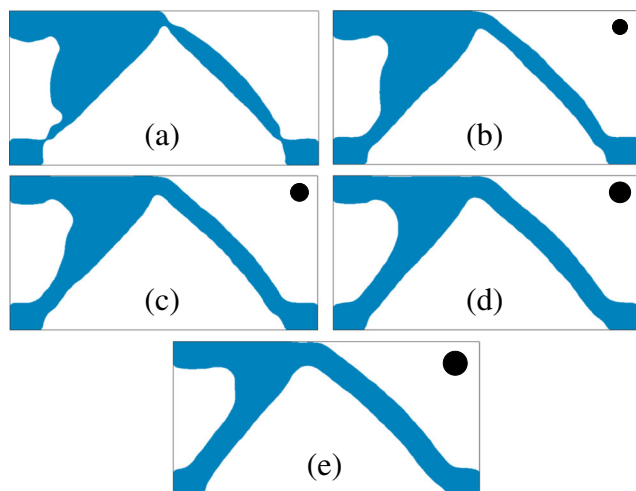


Fig. 15 Inverter mechanism solutions with a no length-scale constraint and a minimum structure length of: b 0.05, c 0.06, d 0.07, and e 0.08

Table 3 Inverter mechanism example solution data

Min. structure length	u_{out}	Iteration
None	25.0	150
0.05	21.0	390
0.06	18.9	410
0.07	16.9	640
0.08	14.5	520

The problem without the length-scale constraint converges in 80 iterations and has a compliance of 2.74 units. Then the minimum structure length-scale constraint is activated and a solution is found after a total of 410 iterations, with a compliance of 2.85 units. The solution without a minimum length-scale, Fig. 17a, has several thin cross members that help resist the applied loads and reduce compliance. However, these members are much thinner than the desired minimum structure length and are removed when the length-scale constraints are activated, Fig. 17b. It is also noticed that the web is thicker to meet the length-scale constraint. This problem demonstrates that the proposed method can control minimum length-scale for three-dimensional problems. This is achieved with only trivial modifications in going from 2D to 3D and the performance of the method for this 3D example is comparable to the performance for the 2D examples.

5.4 Discussion

The examples in this section have shown that the proposed method is successful in obtaining optimal solutions with desired minimum length-scale. However, the addition of the explicit constraint functions to the problem adds significant challenges and complexity. It is also observed that many more iterations are required to obtain a solution with length-scale control, compared to a problem without (the

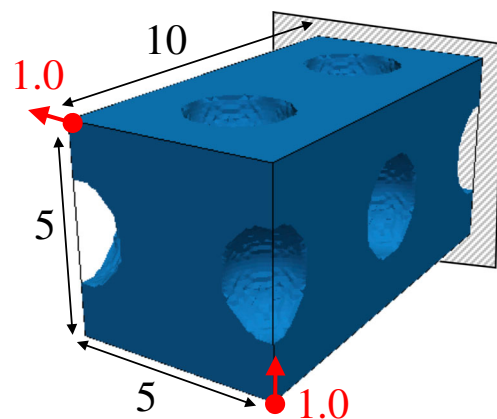
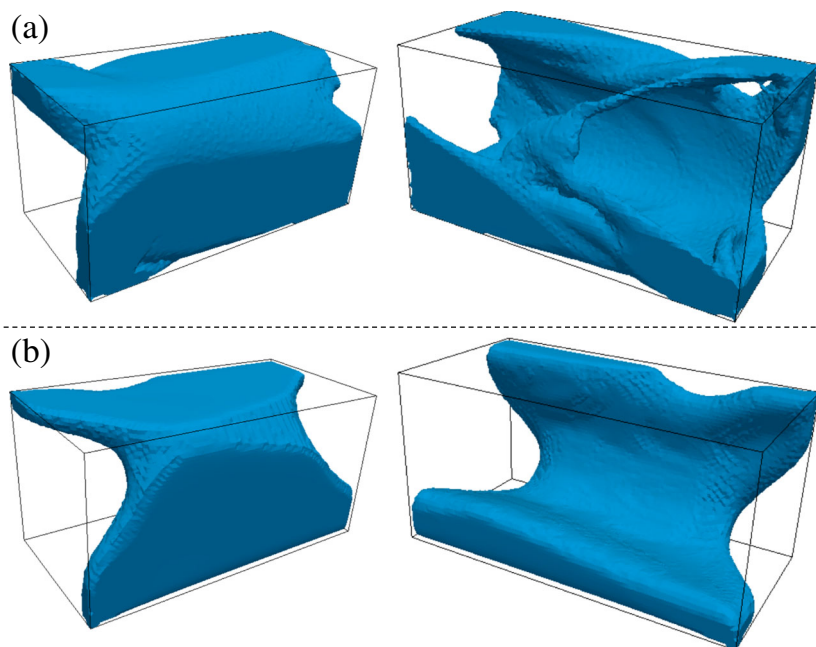


Fig. 16 3D structure example

Fig. 17 3D structure solutions, **a** no length-scale constraint, **b** with length-scale constraint



extreme case is a factor of around 12). One cause of this is that explicit length-scale constraints significantly alter the design space by making large parts infeasible. Therefore, the optimizer can take a large number of iterations to navigate the design space toward an optimum. This is exaggerated by the proposed scheme to dynamically adapt the length-scale constraint tolerance (Section 3.1), which is required to avoid some undesirable local minima.

However, the author is not aware of any alternative general exact length-scale control method suitable for parameterized implicit function based topology optimization. Thus, the method proposed in this paper at least provides a starting point to build and improve upon.

6 Conclusions

This paper introduces minimum length-scale constraint functions suitable for parameterized implicit function based topology optimization. The method is based on comparing the value of a filtered function with a normalized implicit function.

Several implementation aspects are discussed. Firstly, a robust scheme is introduced to set the constraint tolerances for the length-scale constraint functions. The scheme is developed based on observations and calculations from an analytical 1D example. The scheme includes a feature to relax the constraint tolerance to aid the optimizer escaping from local minima created by the introduction of the length-scale constraints. Secondly, it is observed that using the full accurate sensitivity information for the length-scale constraints often results in unsatisfactory solutions,

due to prevention of topology changes. Thus, a hybrid method is proposed, where approximate sensitivity values are initially used to avoid the observed convergence issues for full sensitivities. Then, to help the optimizer find a local minima, the full sensitivities are used. Finally, activating length-scale constraints from the start often results in the design being trapped in a local minima close to the original design. Thus, it is proposed to activate the constraints only after a solution to a problem without length-scale constraints is first obtained.

Several examples are used to demonstrate the ability of the proposed method to control minimum length-scale for different problems and for different reasons. Simultaneous and separate control of minimum structure and void length-scale are demonstrated using a minimum compliance example. The design of hinge free compliant mechanisms is achieved for an inverter problem. The proposed method is also applied to control minimum length-scale for a three-dimensional problem.

Acknowledgements The author would like to thank the Numerical Analysis Group at the Rutherford Appleton Laboratory for their FORTRAN HSL packages (HSL, a collection of Fortran codes for large-scale scientific computation. See <http://www.hsl.rl.ac.uk/>). The author also would like to acknowledge the support of the Maxwell compute cluster funded by the University of Aberdeen. Finally, the author thanks the anonymous reviewers for their helpful comments and suggestions that improved this paper.

Appendix A: Maximum constraint value

In this appendix we calculate the maximum value of the constraint function when $L = 2R$ for a 1D analytical model

of a structural feature. First, the feature is defined by a normalized implicit function:

$$\hat{\phi}(x) = \begin{cases} 0, & x < R, x > 3R, \\ 1, & R \leq x \leq 3R. \end{cases} \quad (19)$$

The analytical linear filter function at any point x is then:

$$\tilde{\phi}(x) = \frac{1}{R^2} \begin{cases} 0.5x^2, & 0 \leq x \leq R, \\ 0.5x^2 - (x - R)^2, & R \leq x \leq 3R, \\ 0.5x^2 - 4Rx + 8R^2, & 3R \leq x \leq 4R. \end{cases} \quad (20)$$

The filtered function is symmetric and thus it is sufficient to continue the calculation on the domain $x \in [0, 2R]$. To construct the constraint function we also need the spatial gradient of the filtered function:

$$\nabla \tilde{\phi}(x) = \frac{1}{R^2} \begin{cases} x, & 0 \leq x \leq R, \\ 2R - x, & R \leq x \leq 2R. \end{cases} \quad (21)$$

From (7), the constraint function is only non-zero when $\hat{\phi}(x) > 0$. Thus, from (19), we only need to calculate the constraint function on the domain $x \in [R, 2R]$ (again using the symmetry of the feature). The constraint function in this domain is then:

$$g_s(x) = \left(1 - \frac{0.5x^2 + (x - R)^2}{R^2}\right)^2 \exp\left(-\frac{c}{R^4}(2R - x)^2\right) \quad (22)$$

To find the maximum value of the constraint function for the 1D analytical case, we differentiate (22) and find solutions for $dg_s/dx = 0$:

$$\begin{aligned} \frac{dg_s}{dx} = & \left[2 \left(1 - \frac{0.5x^2 - (x - R)^2}{R^2}\right) \frac{x - 2R}{R^2} \right. \\ & \left. + \left(1 - \frac{0.5x^2 - (x - R)^2}{R^2}\right)^2 \left(\frac{4Rc - 2xc}{R^4}\right) \right] \\ & \times \exp\left(-\frac{c}{R^4}(2R - x)^2\right) \end{aligned} \quad (23)$$

It can be shown that (23) has the following roots: $x = 2R$ and $x = 2R \pm \sqrt{2}R^2/\sqrt{c}$. The solution $x = 2R$ is the minimum point, as $g_s(2R) = 0$. Thus, $x_{\max} = 2R - \sqrt{2}R^2/\sqrt{c}$, is the location of the maximum value of g_s , as the other solution is outside the range: $x \in [R, 2R]$. Substituting x_{\max} into (22), gives:

$$\delta_g = g_s(x_{\max}) = \left(R^4/c^2\right) \exp(-2) \quad (24)$$

If the value of the exponential decay, c , is chosen as: $c = R^4$, then: $\delta_g = R^{-4} \cdot \exp(-2)$.

Appendix B: Length-scale constraint derivatives

The derivatives of the length-scale constraint functions, (7, 8) with respect to a design variable are detailed in this

appendix. The derivation is similar for both functions, so only the structure length-scale function is shown in detail. Using the product rule, the derivative of (7) with respect to a design variable b_j is:

$$\begin{aligned} \frac{dg_s}{db_j} = & \sum_{i=1}^N \left[-2(1 - \tilde{\phi}_i) \frac{d\tilde{\phi}_i}{db_j} \hat{\phi}_i + (1 - \tilde{\phi}_i)^2 \frac{d\hat{\phi}_i}{db_j} \right. \\ & \left. + (1 - \tilde{\phi}_i)^2 \hat{\phi}_i \left(-c \frac{d(\|\nabla \tilde{\phi}_i\|^2)}{db_j} \right) \right] \\ & \exp\left(-c\|\nabla \tilde{\phi}_i\|^2\right) \end{aligned} \quad (25)$$

The derivative of the normalized implicit function is found using the chain rule and (9):

$$\frac{d\hat{\phi}_i}{db_j} = \begin{cases} \frac{3}{4} \left(\frac{1}{\Delta} - \frac{\phi(x_i)^2}{\Delta^3}\right) \frac{d\phi_i}{db_j} & -\Delta \leq \phi_i < \Delta, \\ 0, & \text{else.} \end{cases} \quad (26)$$

The chain rule is again used to calculate the derivative of the filtered function:

$$\frac{d\tilde{\phi}_i}{db_j} = \frac{\sum_{k \in \mathbb{H}_i} w_{i,k} \frac{d\hat{\phi}_k}{db_j}}{\sum_{k \in \mathbb{H}_i} w_{i,k}} \quad (27)$$

where \mathbb{H}_i is the domain of points within R of node i and the weight function is:

$$w_{i,k} = \begin{cases} R - |x_i - x_k|, & x_k \in \mathbb{H}_i, \\ 0, & \text{else.} \end{cases} \quad (28)$$

Now, we calculate the derivative of the spatial gradient of the filtered function, (6):

$$\begin{aligned} \frac{d(\|\nabla \tilde{\phi}_i\|^2)}{db_j} = & 0.5 \left[(\tilde{\phi}_{i+x} - \tilde{\phi}_{i-x}) \left(\frac{d\tilde{\phi}_{i+x}}{db_j} - \frac{d\tilde{\phi}_{i-x}}{db_j} \right) \right. \\ & \left. + (\tilde{\phi}_{i+y} - \tilde{\phi}_{i-y}) \left(\frac{d\tilde{\phi}_{i+y}}{db_j} - \frac{d\tilde{\phi}_{i-y}}{db_j} \right) \right] \end{aligned} \quad (29)$$

The final derivative required to complete the calculation is the derivative of the implicit function with respect to the design variable, which will depend on the parameterization method. In this paper, there is a linear relationship between the set of parameters and nodal implicit function values, (16). Thus, the required derivative is simply determined from the entries of the matrix A :

$$\frac{d\phi_i}{db_j} = A_{i,j} \quad (30)$$

The approximate derivative, as discussed in Section 3.3, assumes that the derivative of the spatial gradient of the filtered function is zero. Thus, the approximate derivative is:

$$\begin{aligned} \frac{dg_s}{db_j} \approx & \sum_{i=1}^N \left[-2(1 - \tilde{\phi}_i) \frac{d\tilde{\phi}_i}{db_j} \hat{\phi}_i + (1 - \tilde{\phi}_i)^2 \frac{d\hat{\phi}_i}{db_j} \right] \\ & \times \exp\left(-c\|\nabla \tilde{\phi}_i\|^2\right) \end{aligned} \quad (31)$$

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.

References

- Allaire G, Jouve F, Toader AM (2004) Structural optimization using sensitivity analysis and a level-set method. *J Comput Phys* 194(1):363–393
- Allaire G, Jouve F, Michailidis G (2016) Thickness control in structural optimization via a level set method. *Struct Multidiscip Optim* 53(6):1349–1382
- Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *Comput Methods Appl Mech Eng* 71(2):197–224
- Bendsøe MP, Sigmund O (2004) *Topology Optimization: Theory, Methods and Applications*, Springer, Germany
- 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
- De Ruiter M, Van Keulen F (2004) Topology optimization using a topology description function. *Struct Multidiscip Optim* 26(6):406–416
- Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Struct Multidiscip Optim* 49(1):1–38
- Dunning PD (2017) Design parameterization for topology optimization by intersection of an implicit function. *Comput Methods Appl Mech Eng* 317:993–1011
- Dunning PD, Kim HA (2015) Introducing the sequential linear programming level-set method for topology optimization. *Struct Multidiscip Optim* 51(3):631–643
- Gomes AA, Suleman A (2006) Application of spectral level set methodology in topology optimization. *Struct Multidiscip Optim* 31(6):430–443
- Guest JK (2009) Topology optimization with multiple phase projection. *Comput Methods Appl Mech Eng* 199(1):123–135
- Guest JK, Prévost JH, Belytschko T (2004) Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *Int J Numer Methods Eng* 61(2):238–254
- 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) Explicit feature control in structural topology optimization via level set method. *Comput Methods Appl Mech Eng* 272:354–378
- Hamza K, Aly M, Hegazi H (2014) A kriging-interpolated level-set approach for structural topology optimization. *J Mech Des* 136(1):011,008
- Hoang VN, Jang GW (2017) Topology optimization using moving morphable bars for versatile thickness control. *Comput Methods Appl Mech Eng* 317:153–173
- Jenkins N, Maute K (2016) An immersed boundary approach for shape and topology optimization of stationary fluid-structure interaction problems. *Struct Multidiscip Optim* 54(5):1191–1208
- Johnson SG (2014) The NLOpt nonlinear optimization package. <http://ab-initio.mit.edu/nlopt>
- Kennedy G (2015) Large-scale multi-material topology optimization for additive manufacturing. In: *AIAA Scitech 2015*, Kissimmee, FL, pp 1–13
- Lazarov BS, Wang F, Sigmund O (2016) Length scale and manufacturability in density-based topology optimization. *Arch Appl Mech* 86(1-2):189–218
- Liu J, Ma Y (2016) A survey of manufacturing oriented topology optimization methods. *Adv Eng Softw* 100:161–175
- Liu J, Ma Y, Fu J, Duke K (2015) A novel cad/cad/cae integrated design framework for fiber-reinforced plastic parts. *Adv Eng Softw* 87:13–29
- Liu J, Yu H, Ma Y (2016) Minimum void length scale control in level set topology optimization subject to machining radii. *Comput Aided Des* 81:70–80
- Luo J, Luo Z, Chen L, Tong L, Wang MY (2008a) A semi-implicit level set method for structural shape and topology optimization. *J Comput Phys* 227(11):5561–5581
- Luo J, Luo Z, Chen S, Tong L, Wang MY (2008b) A new level set method for systematic design of hinge-free compliant mechanisms. *Comput Methods Appl Mech Eng* 198(2):318–331
- Luo Z, Tong L, Wang MY, Wang S (2007) Shape and topology optimization of compliant mechanisms using a parameterization level set method. *J Comput Phys* 227(1):680–705
- Munk DJ, Vio GA, Steven GP (2015) Topology and shape optimization methods using evolutionary algorithms: a review. *Struct Multidiscip Optim* 52(3):613–631
- 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
- Pingen G, Waidmann M, Evgrafov A, Maute K (2010) A parametric level-set approach for topology optimization of flow domains. *Struct Multidiscip Optim* 41(1):117–131
- Poulsen TA (2003) A new scheme for imposing a minimum length scale in topology optimization. *Int J Numer Methods Eng* 57(6):741–760
- Rogers DF (2000) *An introduction to NURBS: with historical perspective*. Elsevier
- Sethian J (1999) *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision and materials science*. Cambridge University Press, Cambridge
- Sigmund O (2009) Manufacturing tolerant topology optimization. *Acta Mech Sinica* 25(2):227–239
- Sigmund O, Maute K (2013) Topology optimization approaches. *Struct Multidiscip Optim* 48(6):1031–1055
- Sigmund O, Petersson J (1998) Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct Multidiscip Optim* 16(1):68–75
- van Dijk NP, Maute K, Langelaar M, Van Keulen F (2013) Level-set methods for structural topology optimization: a review. *Struct Multidiscip Optim* 48(3):437–472
- Vatanabe SL, Lippi TN, de Lima CR, Paulino GH, Silva EC (2016) Topology optimization with manufacturing constraints: a unified projection-based approach. *Adv Eng Softw* 100:97–112
- Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. *Struct Multidiscip Optim* 43(6):767–784
- Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. *Comput Methods Appl Mech Eng* 192(1):227–246
- Wang Y, Zhang L, Wang MY (2016) Length scale control for structural optimization by level sets. *Comput Methods Appl Mech Eng* 305:891–909
- Xia Q, Shi T (2015) Constraints of distance from boundary to skeleton: for the control of length scale in level set based structural topology optimization. *Comput Methods Appl Mech Eng* 295:525–542

- 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 S, Gain AL, Norato JA (2017a) Stress-based topology optimization with discrete geometric components. *Comput Methods Appl Mech Eng* 325:1–21
- Zhang W, Zhong W, Guo X (2014) An explicit length scale control approach in simp-based topology optimization. *Comput Methods Appl Mech Eng* 282:71–86
- Zhang W, Li D, Zhang J, Guo X (2016b) Minimum length scale control in structural topology optimization based on the moving morphable components (mmc) approach. *Comput Methods Appl Mech Eng* 311:327–355
- Zhang W, Chen J, Zhu X, Zhou J, Xue D, Lei X, Guo X (2017b) Explicit three dimensional topology optimization via moving morphable void (mmv) approach. *Comput Methods Appl Mech Eng* 322:590–614
- Zhang W, Liu Y, Wei P, Zhu Y, Guo X (2017c) Explicit control of structural complexity in topology optimization. *Comput Methods Appl Mech Eng* 324:149–169
- Zhang W, Yang W, Zhou J, Li D, Guo X (2017d) Structural topology optimization through explicit boundary evolution. *J Appl Mech* 84(1):011,011
- Zhou M, Lazarov BS, Wang F, Sigmund O (2015) Minimum length scale in topology optimization by geometric constraints. *Comput Methods Appl Mech Eng* 293:266–282