

Design parameterization for topology optimization by intersection of an implicit function

Peter D. Dunning^{a,*}

^a*School of Engineering, University of Aberdeen, Aberdeen AB24 3UE, UK*

Abstract

This paper introduces a new approach to topology optimization where the structural boundary is defined by the intersection of an implicit signed-distance function and a cutting surface. The cutting surface is discretized using finite element shape-function polynomials and the nodal values become the design variables during optimization. Thus, the design parameterization is separated from the analysis discretization, which enables a reduction in the number of design variables, compared with methods that use element-wise variables. The proposed method can obtain solutions with smooth boundaries and the parameterization allows solution by standard optimization methods. Several 2D and 3D examples are used to demonstrate the effectiveness of the method, including minimization of compliance and complaint mechanism problems. The results show that the method can obtain good solutions to well-known problems with smooth, clearly defined boundaries and that this can be achieved using significantly fewer design variables compared with element-based methods.

Keywords: Topology optimization, implicit function, mathematical programming.

1. Introduction

Structural topology optimization aims to simultaneously optimize the size, shape and connectivity, or topology, of a structure. It can obtain more efficient designs, compared with only size or shape optimization, because the optimizer is not constrained by a predetermined layout, which is often a key factor influencing structural performance.

Several approaches have been developed to solve topology optimization problems, which have been successfully applied to many applications [1, 2]. The two most popular approaches are those based on element-wise variables, such as SIMP (Solid Isotropic Material with Penalization) and ESO (Evolutionary Structural Optimization), and those that utilize an implicit function to represent

*Corresponding author

Email address: `peter.dunning@abdn.ac.uk` (Peter D. Dunning)

the boundary, such as level-set and phase field methods. Both the element based and implicit function based approaches normally require gradient or sensitivity information to efficiently drive the solution towards an optimum. For most element density-based methods, the derivative of the objective and constraint functions with respect to element densities are required [1] and for some implicit function approaches, shape and possibly topological derivatives are required [3]. The use of gradient information allows for efficient solution using gradient based optimization approaches. Therefore, most current topology optimization methods are well suited to efficiently solve problems where gradient information is easily and efficiently obtained. However, issues can arise when gradients are either impossible to compute (due to a discontinuous or noisy design space), difficult to compute (due to complex multidisciplinary interaction) or expensive to compute (for example, due to time dependent functions).

Several techniques have been developed to overcome these difficulties. One solution to handle a non-differentiable function is to replace it with a smooth, differentiable approximation. A classic example is the maximum stress function, which can be reformulated using a constraint aggregation approach [4]. However, this technique is problem dependent and may not work for all classes of problem, for example crash or impact scenarios involving contact and plasticity [5]. For problems with difficult to obtain derivatives, finite differences [6] or automatic differentiation [7] can be used. However, these approaches can be computationally expensive because the number of design variables in topology optimization is often high (typically in the order of 10^5 or 10^6 for 3D problems). A semi-analytical approach can be used for some problems, where the derivative is analytically derived, but finite differences are used to compute part of the derivative, such as the derivative of the system matrices with respect to the design variables [8]. When derivatives are expensive to compute, a simplification can be used that assumes the expensive part of the derivative is zero. This approach is often used when the total derivative involves one or more eigenvector derivatives, such as when eigenvector basis functions are used to create a reduced order model in a flutter analysis [9]. The simplified derivative is cheaper to compute, but is only an approximation, which may or may not be accurate enough for good convergence.

The alternative approach to the techniques discussed above is to use an optimizer that does not require gradients, such as a genetic algorithm or particle swarm method. However, the required number of function evaluations for these approaches often scales exponentially with the number of design variables, resulting in an unacceptably large computational cost for current popular topology optimization methods [10].

In summary, current techniques used to overcome difficulties in obtaining derivatives in topology optimization are either problem dependent and may not be effective for all classes of problem (e.g. smooth approximations), result in approximate derivatives, or are expensive because the number of design variables is often large (e.g. finite differencing, automatic differentiation and gradient-free methods).

Therefore, one potential approach to address these difficulties is to formulate

the topology optimization problem in a way that reduces the number of design variables, compared with current popular methods. If the reduced design space is then combined with techniques that reduce the analysis computational cost (see for example [11, 12, 13]) the result may be a computationally tractable approach for performing topology optimization using finite differencing, automatic differentiation or even gradient-free optimization methods. A recent demonstration of this idea was presented by Yoshimura et al. [14] where topology optimization for a multi-objective thermal-fluid problem was performed using a genetic algorithm, assisted by a Kriging surrogate model. Another example was presented by Bujny et al [15], which used the beam implicit function representation introduced by Guo et al [16] combined with an evolutionary strategy to optimize a structure for an impact scenario. Both these methods use a formulation based on a parameterized implicit function to represent the structure. Reducing the number of design variables may also benefit problems where gradients are easily obtained because faster convergence and a better solution may be achieved by using an optimizer that uses second order information [17].

The benefits of using a parameterized implicit function for topology optimization are that the solutions have smooth boundaries, some numerical artifacts associated with element based methods are avoided, such as checkerboard patterns [3], problems can be solved using standard optimizers, such as mathematical programming or gradient-free approaches, and finally that the number of design variables can be reduced because the design parameterization is decoupled from the analysis mesh. Some existing implicit function parameterization methods include: radial basis functions [18], geometric shapes [16], Fourier series [19] and finite element shape functions [20]. Also, projection methods offer similar features [21], as the solutions can have smooth boundaries and the problem can be solved using standard optimizers.

The aim of this paper is to detail a new parameterized implicit function-based topology optimization approach and investigate its numerical performance with a particular focus on how much it can effectively reduce the number of design variables. The new parameterization method was first introduced by the author in a conference paper [22]. The idea is to combine a signed-distance implicit function with a cutting surface. The cutting surface is parameterized and the parameters become the design variables. In this paper, the new method is presented, validated and investigated using problems where analytical gradients are easily and cheaply computed and the problems are solved using an efficient gradient based-optimizer. However, future work will address problems where gradients are not so easily computed. The new method is introduced in Section 2, numerical implementation issues are discussed in Section 3, numerical investigations are presented in Section 4, followed by a discussion on the method in Section 5 and conclusions in Section 6.

2. Design representation and optimization

First, an initial design is chosen and mapped onto a design domain using an implicit signed-distance function, where the boundary is the zero level-set and a positive value indicates a point inside the structure, Figure 1. The structure is then defined by the scalar value of an implicit function:

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

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.

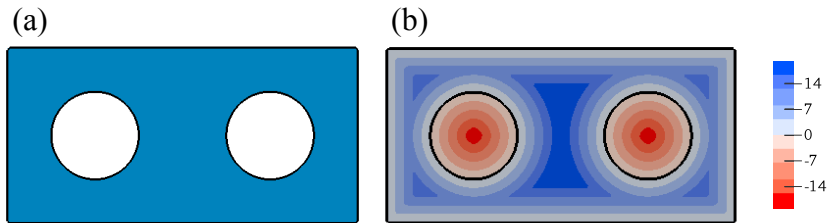


Figure 1: a) Original structure, b) Contours of the signed distance function.

Another way to interpret this representation of the structure is that the boundary is the zero-level set of a higher dimensional implicit function. Therefore, we can say that the boundary is the intersection of a plane (defined by all points where $\phi = 0$) and the implicit function surface in the higher dimension. This is shown by the black line in Figure 2a for the 2D structure in Figure 1. To change the position of the structural boundary we can fix the position of the intersecting plane and change the implicit function, which is the approach in most level-set topology optimization methods [3]. Alternatively, we could change the position of the intersecting plane. This second approach is demonstrated in Figure 2, where the intersecting plane has been moved (shown in red), resulting in the structural boundary moving position, Figure 2b.

The aim of this paper is to create a topology optimization method that exploits the moving intersecting plane mechanism for changing the position and connectivity of the structural boundary. The position of a plane in 3D can be described by 3 variables and we could proceed to develop an optimization method for 2D structures that uses just 3 variables. However, the design space would be very limited and therefore this approach is not practical.

To provide more design freedom, the intersecting plane is replaced by a cutting surface that is defined in the higher dimension. Therefore, the method introduced in this paper uses two scalar valued functions: the implicit signed-distance function of the initial structure, $\alpha(x)$, and the cutting surface, $\beta(x, \mathbf{b})$, where \mathbf{b} is a vector of design parameters that define the cutting surface. The

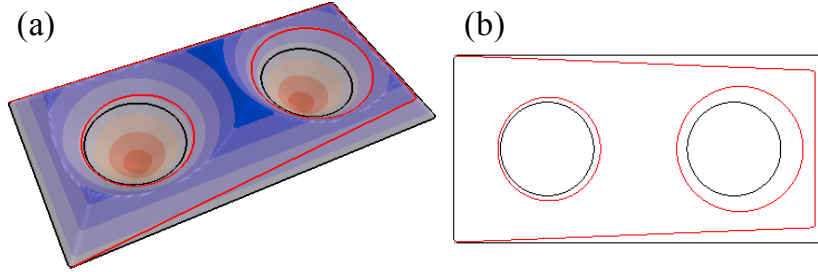


Figure 2: a) Implicit function surface with zero level-set contour (black) and new position of the intersecting plane (red), b) Resulting change in the structural boundary.

boundary of the structure is then defined by the set of points where the value of the signed-distance function equals the value of the cutting surface. Alternatively, the structural boundary is defined as the zero level-set of the implicit scalar-valued function that is the sum of the signed-distance and cutting surface functions.

Thus, the definition of $\phi(x)$ in Eq. (1) can be written as:

$$\phi(x) = \alpha(x) - \beta(x, \mathbf{b}) \quad (2)$$

There are several possible methods to parameterize the cutting surface, such as radial basis functions, polynomials or spline functions. In this paper, the cutting surface is parameterized using finite element shape functions, which is detailed in Section 3.1.

During optimization, the signed-distance function remains fixed and the structure boundary changes position (and possibly connectivity) as the parameters of the cutting surface change. However, the design space is then limited by the current signed-distance function. Therefore, once an optimal cutting surface has been found for the current signed-distance function, a new signed-distance function is generated from the boundary of the current structure. The optimal cutting surface for the new signed-distance function is then found. This leads to an optimization method with an inner and outer loop. The aim of inner loop is to find the optimal cutting surface for the current signed-distance function. The outer loop then generates a new signed-distance function from the new structural boundary (as defined by the old signed-distance function and optimal cutting surface). At the start of the inner loop, the cutting surface is initialized as: $\beta(x, \mathbf{b}) = 0$ (identical to the zero level-set), so that the structure boundary is the same as the optimum found by the previous inner loop (or initial structure). A convergence proof for this approach is not presented. However, the results of the numerical examples (Section 4) demonstrate that the reinitialization does not cause convergence problems.

The final aspect of this initial conceptual presentation of the proposed method is to link the implicit structure description to a computational analysis tool, so that the objective and constraint functions can be evaluated for a given

set of design variables, \mathbf{b} . There are several approaches in the literature to generate finite element (FE) matrices from an implicit function, such as the ersatz material (or density) method [23], smoothed Heaviside function [24], eXtended finite element method [25] or re-meshing to generate a boundary fitted mesh [26]. In this paper the inner-loop optimization problem is solved using a mathematical programming method, thus it is desirable to use a method that enables an explicit link between the design variables and objective and constraint functions, so that analytical gradients can be computed. Therefore, the smoothed Heaviside function approach proposed by Wang et al [24] is used in this paper, although, in general, any method that can generate FE matrices from an implicit function can be used with the cutting surface parameterization introduced in this work. Further details on the numerical implementation are discussed in Section 3.2.

3. Numerical implementation

3.1. Cutting surface parameterization

The proposed method uses two meshes that cover the design domain: one to discretize the cutting surface (cutting surface mesh) and one to perform the finite element analysis (analysis mesh), as shown in Figure 3.

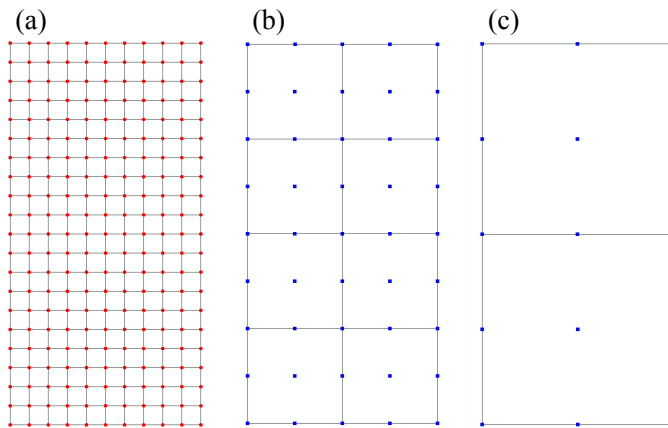


Figure 3: a) Analysis mesh using 200 Q4 elements, b) Cutting surface mesh using eight (2×4) Q9 elements (45 design variables), c) Cutting surface mesh using two Q9 elements (15 design variables).

The value of $\beta(x, \mathbf{b})$ is defined at nodes in the cutting surface mesh and interpolated using polynomial shape functions. The nodal values of $\beta(x, \mathbf{b})$ are then the design variables \mathbf{b} . The implicit function, $\phi(x)$, and signed-distance function, $\alpha(x)$, are discretized on the analysis mesh. Values of $\phi(x)$ at analysis element integration points are obtained by interpolation using the analysis

element shape functions. Equation (2) can be written in terms of the discretized variables:

$$\phi(\mathbf{b}) = \boldsymbol{\alpha} - \mathbf{A} \cdot \mathbf{b} \quad (3)$$

where ϕ and $\boldsymbol{\alpha}$ are vectors of the discretized implicit and signed-distance functions, respectively, and \mathbf{A} is a coefficient matrix that maps the cutting surface values to the nodes of the analysis mesh. The matrix \mathbf{A} is constant during optimization and is only computed once.

Typically, the analysis and cutting surface meshes are different, thus the analysis discretization is decoupled from the design parameterization, which enables a reduction in the number of design variables, compared with element-based approaches. In developing the new method, three different element types were investigated for the discretization of the cutting surface in 2D. These were 4 node bilinear elements (Q4), 8 node quadratic serendipity elements (Q8) and second order polynomial elements with 9 nodes (Q9). Several benchmark problems were solved using each element type and the general trend observed was that solutions obtained using Q9 elements achieve a better objective function value in fewer iterations, compared with the other two element types. Compared with the Q4 element, the Q9 element has more flexibility in representing different shapes for the same number of design variables, due to the extra polynomial terms. The Q8 element did not perform satisfactorily, because the shape functions allow a non-monotonic link between the design variables, \mathbf{b} and implicit function ϕ , which disrupts and slows down convergence. This is demonstrated by the example in Figure 4 where the problem failed to converge in 400 iterations when using Q8 elements, whereas less than 200 iterations were required for convergence when using Q4 or Q9 elements.

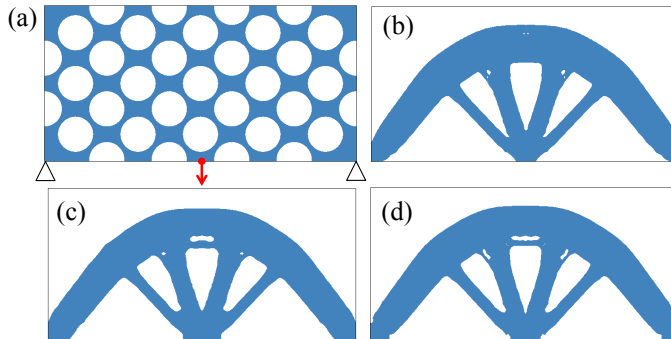


Figure 4: Michell structure. a) Initial design, loading and boundary conditions. Solution using: b) Q4 elements (158 iterations), c) Q8 elements (400 iterations) and d) Q9 elements (187 iterations).

In this paper, all 2D examples use Q4 and Q9 elements to discretize the analysis mesh and cutting surface, respectively, Figure 3. In 2D, if one cutting surface Q9 element is used for every four Q4 analysis elements then the nodes

of the two meshes will be coincident, \mathbf{A} becomes the identity matrix and the method resembles one using nodal design variables. This scenario should be avoided, as it is known that using nodal implicit function values directly for optimization results in an ill-conditioned problem [20].

Finally, the signed-distance function is computed before the start of each inner loop using the fast marching method [27].

3.2. Smoothed Heaviside function

The smoothed Heaviside approach is used to generate FE matrices from the nodal implicit function values. In practice this is achieved by multiplying the material properties at each element integration point by the value of the smoothed Heaviside function. For example the numerical integration of an element stiffness matrix is:

$$K_e = \sum_{i=1}^{N_g} H(\phi(x_i)) B_i^T E_i B_i w_i \quad (4)$$

where K_e is the element stiffness matrix, N_g is the number of integration points, B_i , E_i and w_i are the strain displacement matrix, material stiffness matrix and integral weight at point i , respectively, and $H(\phi(x_i))$ is a smoothed Heaviside function. In this work a polynomial form of $H(\phi)$ is used [24]:

$$H(\phi) = \begin{cases} \rho_{min} & \phi < -\Delta, \\ \frac{3(1-\rho_{min})}{4} \left(\frac{\phi}{\Delta} - \frac{\phi^3}{3\Delta^3} \right) + \frac{(1+\rho_{min})}{2} & -\Delta \leq \phi < \Delta, \\ 1 & \Delta \leq \phi, \end{cases} \quad (5)$$

where Δ is the smoothing length of the approximation and ρ_{min} is a parameter that defines the stiffness of the void region, in a similar way to the minimum density value in an element-based topology optimization method.

One of the benefits of using an implicit function for topology optimization, compared with element-based methods, is that the solution has smooth, well-defined boundaries. However, this is partially lost when using a smoothed Heaviside function, as the boundary becomes effectively blurred over the smoothing length [3]. To counter this, two strategies are proposed. Firstly a small smoothing length is used throughout the optimization, ideally smaller than the element edge length in the analysis mesh. However, this strategy may cause convergence problems with some examples, especially those that have a small volume ratio. Thus, a second strategy is proposed where the optimization starts with a large smoothing length that is then halved each time the outer loop converges. The optimization continues until a converged solution using a lower limit of the smoothing length is obtained. This strategy is similar to the beta-continuation techniques used in Heaviside projection schemes [21, 28].

3.3. Inner loop optimization

The inner loop optimization problem can be stated as:

$$\begin{aligned} \min_{\mathbf{b}} \quad & f(\phi(\mathbf{b})) \\ \text{s.t.} \quad & g_i(\phi(\mathbf{b})) \leq 0, \quad i = 1 \dots m \\ & \mathbf{b}_{min} \leq \mathbf{b} \leq \mathbf{b}_{max} \end{aligned} \quad (6)$$

where $f(\phi(\mathbf{b}))$ is the objective function, $g_i(\phi(\mathbf{b}))$ the constraint functions, m the total number of constraints, \mathbf{b}_{min} and \mathbf{b}_{max} are the side limits on the design variables.

The solution to the inner loop problem is affected by the choice of optimizer, design variable side limits and convergence criteria. In this paper the inner loop problem is solved using the method of moving asymptotes, as implemented in the package NLOPT [29]. This optimizer was chosen as it has worked well with other topology optimization methods [1, 18, 21]. The side limits are defined from the maximum and minimum signed-distance values of the current structure, with considerations for the boundary of the design domain, Γ_D , and fixed, non-designable regions within the design domain, Ω_{fixed} . First, a parameter is obtained from the discrete values of the signed-distance function:

$$\hat{\alpha} = \max\{0.5 \times \min\{\max\{\alpha\}, -\min\{\alpha\}\}, 2h\} \quad (7)$$

where h is the analysis mesh grid spacing.

The side limits for a design variable are then defined as:

$$b_{i,min} = \begin{cases} 0 & \text{if } b_i \in \Omega_{fixed} \\ -\hat{\alpha} & \text{else} \end{cases} \quad b_{i,max} = \begin{cases} -\alpha_i & \text{if } b_i \in \Gamma_D \\ \hat{\alpha} & \text{else} \end{cases} \quad (8)$$

Three convergence criteria, as defined by the NLOPT package, are used for the inner loop: the maximum number of iterations, the relative change of the objective function and relative change of the design variables. The relative change criteria are both set to 10^{-3} and the maximum number of iterations to 10. The inner loop stops when any of these criteria are met. The criteria are intentionally slack, as this reduces the number of inner loop iterations spent fine tuning the structure, whereas quicker progress is often made by obtaining an approximate solution and generating a new signed-distance function for the next inner loop.

4. Numerical investigations

Four classic benchmark problems are used for the initial numerical investigation of the method introduced in this paper. These are shown in Figure 5, where the objective is to minimize compliance, with an upper limit on the total material volume. The grey areas for the Michell structures show fixed regions within the design domain. The volume constraint for all problems is set to 40% of the design domain, except for cantilever 2:1, where the constraint is

50%. For all examples, the material properties are Young’s modulus 1.0 and Poisson’s ratio 0.3, and the load magnitude is 10. Unless otherwise stated, the parameters for the smoothed Heaviside, Eq. (5), are: $\Delta = 0.5h$ and $\rho_{min} = 10^{-6}$.

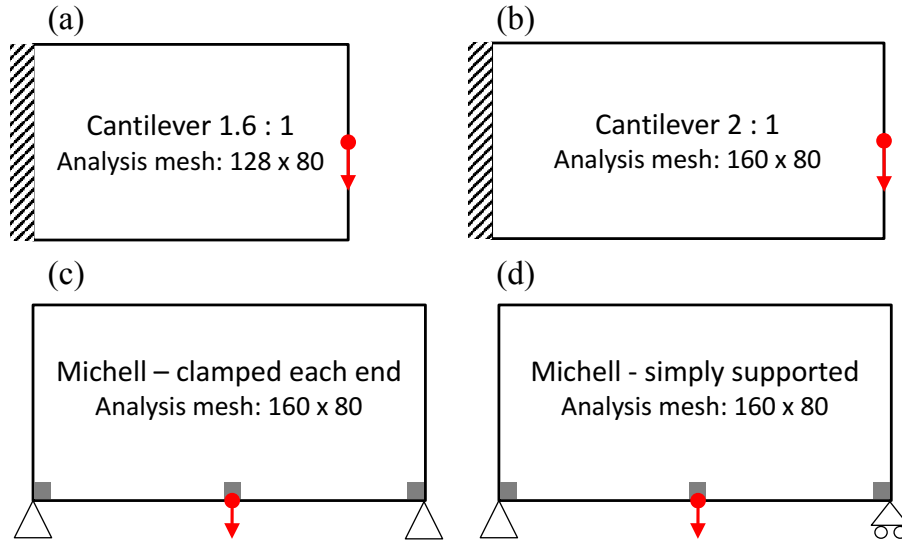


Figure 5: 2D examples. a) Cantilever 1.6:1, b) Cantilever 2:1, c) Clamped Michell structure, d) Simply supported Michell structure.

4.1. Parameterization approach

In this section an example is used to demonstrate the effect of using (or not using) the signed-distance function in the parameterization and the effect of the full-resolution parameterization (i.e. one design variable per analysis mesh node). The signed-distance function can easily be eliminated from the parameterization by setting $\alpha = 0$ in Eq.(2) throughout the optimization. In this case the algorithm terminates at the first inner loop convergence, as no signed-distance function reinitialization is required, and the maximum number of inner loop iterations is increased to 300. The clamped Michell problem, Figure 5c, is solved four times using different parameterizations of the implicit function: $\alpha = 0$ with full-resolution, $\alpha = 0$ with a 16×8 cutting surface mesh discretization, $\alpha =$ signed distance with full-resolution and $\alpha =$ signed distance with a 16×8 cutting surface mesh discretization. The initial design is the same as Figure 4a.

The results are shown in Figure 6, which demonstrate that unsatisfactory solutions are obtained when the signed-distance function is omitted from the parameterization, as a significant amount of intermediate "grey" material is present. This occurs for both the full and reduced-resolution parameterizations.

Therefore, some penalization or regularization is required to obtain discrete 0-1 solutions. From the results we can see that the inclusion of the signed-distance function in the parameterization provides some regularization, as discrete 0-1 solutions are obtained without grey areas. This demonstrates the important role that the signed-distance function has in the direct parameterization of the implicit function presented in this paper. Furthermore, when using the full-resolution parameterization (with signed-distance function), a significant amount of checkerboard type numerical artifacts appear in the solution. This is due to the ill-conditioning of this full-resolution parameterization, which was also observed in [20]. Therefore, the full-resolution solution is not suitable to use as a benchmark to compare with solutions using a reduced-resolution parameterization.

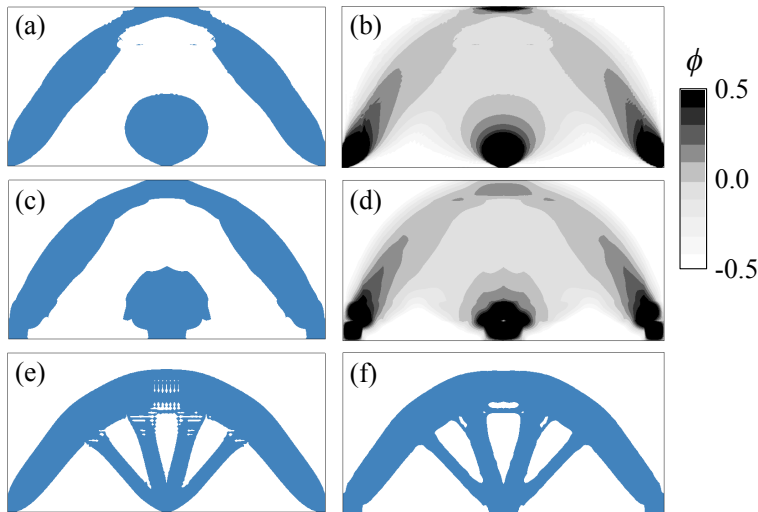


Figure 6: Effect of parameterization approach. a-b) $\alpha = 0$, full-resolution, c-d) $\alpha = 0$, 16×8 cutting surface mesh, e) $\alpha = \text{signed distance}$, full-resolution, f) $\alpha = \text{signed distance}$, 16×8 cutting surface mesh.

4.2. Mesh dependency

The effect of changing the analysis mesh discretization, for a fixed cutting surface mesh, is now investigated. The cantilever 2:1 problem, Figure 5b, is solved with a cutting surface mesh of 16×8 elements and three different analysis meshes: 80×40 , 160×80 and 320×160 elements. All problems use the same smoothing length for the Heaviside, calculated as half an element edge length for the 160×80 analysis mesh. The initial design is the same as Figure 4a.

The results are shown in Figure 7. The solutions using the 160×80 and 320×160 element analysis meshes are almost identical, which demonstrates a degree of mesh independency. However, the solution using the 80×40 analysis

mesh has a simpler topology. This is because the ϕ field is discretized on the nodes of the analysis mesh. Thus, as the analysis mesh becomes coarser, the minimum length of member that can be represented increases and thin members disappear from the solution. Also, there are more numerical artifacts in the 80×40 elements solutions. Both mesh independency and occurrence of numerical artifacts could be improved by adding some rigorous length scale control to the proposed method.



Figure 7: Mesh dependency. Cantilever 2:1 solutions using a cutting surface mesh of 16×8 elements and an analysis mesh of: a) 80×40 , b) 160×80 and c) 320×160 elements.

4.3. Initial design

In this section the effect of the initial design is investigated using the cantilever 1.6:1 problem. The results for four different initial designs are shown in Figure 8, which shows that the initial design does effect the solution. The results also show that this method can sometimes create new holes during optimization, as the solutions with one and three initial internal holes end up with three and five internal holes respectively. However, for 2D problems, it was observed that generally the best solutions were found using an initial design that contained many internal holes. This approach is also used in several other level-set methods [20, 23, 24].

4.4. Reducing the number of design variables.

The effect of reducing the number of design variables is now investigated by solving all four benchmark problems with different discretizations of the cutting surface. The initial design for all examples is the same as in Figure 4a, except for the cantilever 1.6:1 problem, where the initial design with the most holes from Figure 8 is used. The results are summarized in Table 1 and the optimized designs are shown in Figures 9 to 12. In general, the shapes and topologies of the solutions agree well with the known analytical solutions and solutions obtained using other methods [1, 3]. Table 1 shows that, for all examples, compliance increases as the number of design variables decreases. Thus, the reduction in design space results in a worse value of the objective function. However, for most of these examples, the reduction is relatively small compared with the reduction in the number of design variables. For the two cantilever examples, reducing the number of design variables by a factor of four results in an increase in compliance of between 1% and 4%. For the two Michell structure examples, this is also true when comparing solutions using 2145 and 561 variables, but a

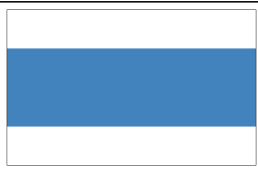
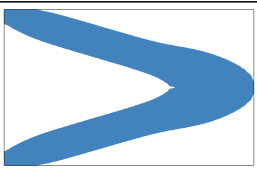
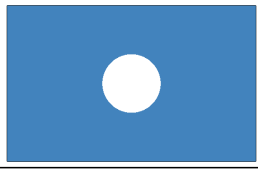
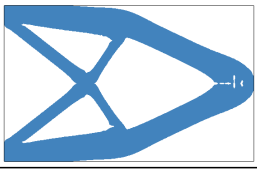
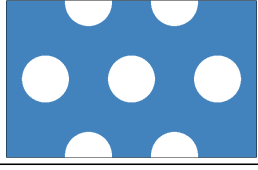
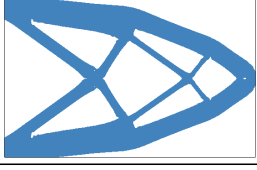
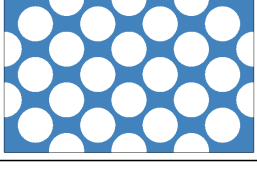
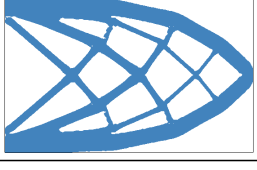
Initial design	Solution	Compliance	Iterations
		4993	222
		4429	203
		4348	131
		4331	85

Figure 8: Effect of initial design on the solution of the cantilever 1.6:1 problem, using a cutting surface mesh of 25×16 elements.

larger increase of between 9% and 13% is observed comparing solutions using 561 and 153 variables.

In this paper the derivatives are computed analytically and the most expensive computational procedure in each iteration is the FE analysis. Thus, in this case, the number of iterations is approximately linearly proportional to computational time. From the results in Table 1 it is observed that, generally, when the number of design variables is reduced, more iterations are required to reach convergence. The exceptions are the cantilever 2:1 and clamped boundary Michell problems that use the least number of design variables, which take fewer iterations. This is caused by early termination due to an over-reduction of the design space. However, the result of the simply supported Michell problem for the least number of design variables demonstrates that convergence failure can also occur when the design space is reduced too much. Excluding this failed convergence result, the maximum increase in iterations is approximately 50% when reducing the number of design variables by a quarter. When derivatives are cheap to compute (as they are here), this represents an increase in the computational time with a reduction in the number of design variables. However, if finite differences were used to compute derivatives, then computational time would reduce. For example, if finite differences were used for the cantilever 2:1 problem, 160,950 FE analyses would be required when using 2145 design variables, but only 62,382 are required when using 561 design variables (a reduction of 61%).

Table 1: Summary of results for 2D minimization of compliance examples.

Example	Cutting surface discretization	No. variables	Compliance	Iterations
Cantilever 1.6:1	25×16	1683	4331	85
	12×8	425	4462	110
	6×4	117	4614	122
Cantilever 2:1	32×16	2145	5969	75
	16×8	561	6049	111
	8×4	153	6136	79
Clamped boundary Michell	32×16	2145	1109	178
	16×8	561	1141	187
	8×4	153	1291	69
Simply supported Michell	32×16	2145	1618	155
	16×8	561	1675	189
	8×4	153	1829	400

Convergence histories for the Cantilever 2:1 problem are given in Figure 13. This shows that convergence is reasonably smooth, with only a few minor oscillations. The evolution of α , β and the structure ($\phi \geq 0$) using 16×8 cutting surface elements is shown in Figure 14. This illustrates that, for this problem, the topology of the solution is reached in around 30 iterations and the remaining iterations are mainly used to optimize the shape of the structure.

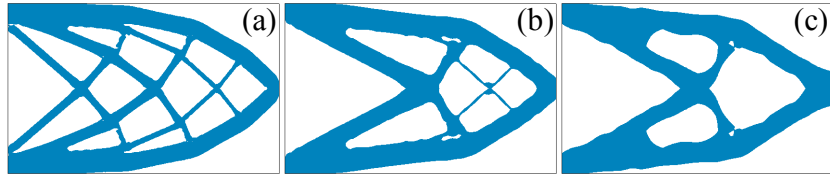


Figure 9: Cantilever 1.6:1 compliance solutions, with cutting surface discretizations: a) 25×16 , b) 12×8 , and c) 6×4 elements.



Figure 10: Cantilever 2:1 compliance solutions, with cutting surface discretizations: a) 32×16 , b) 16×8 , and c) 8×4 elements.



Figure 11: Clamped Michell compliance solutions, with cutting surface discretizations: a) 32×16 , b) 16×8 , and c) 8×4 elements.



Figure 12: Simply supported Michell compliance solutions, with cutting surface discretizations: a) 32×16 , b) 16×8 , and c) 8×4 elements.

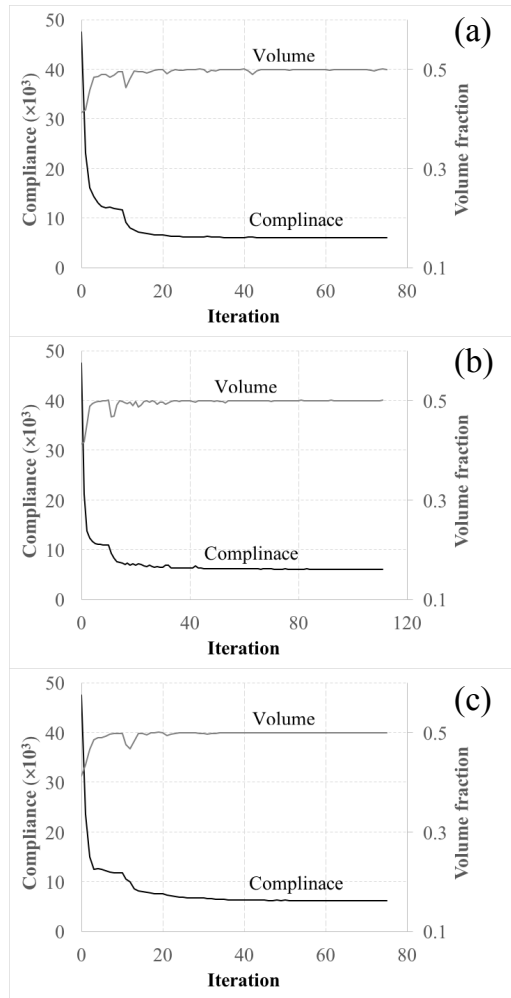


Figure 13: Cantilever 2:1 convergence history for: a) 32×16 , b) 16×8 , and c) 8×4 cutting surface elements.

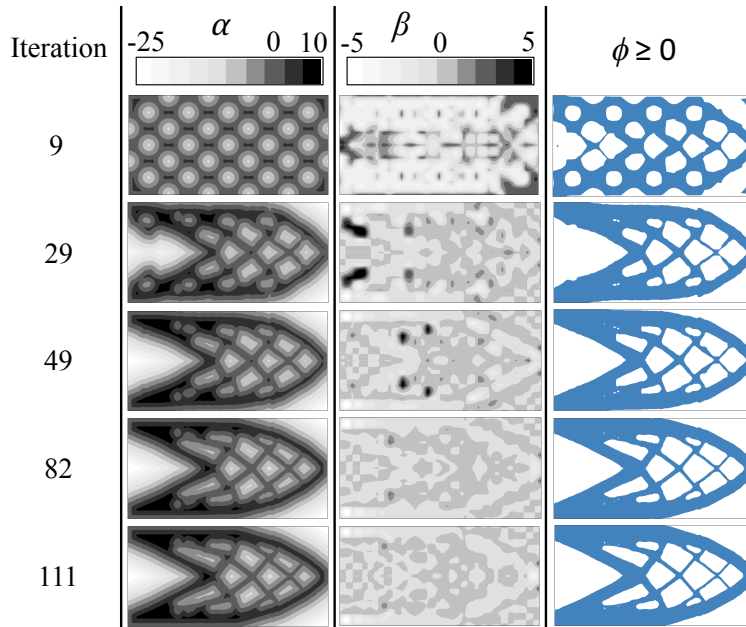


Figure 14: Evolution of Cantilever 2:1 using 16×8 cutting surface elements.

These examples demonstrate that generally the objective will increase (get worse) as the number of variables decreases and that there is a limit to how much the design space can be reduced until the objective value increases significantly. Finally, good solutions were obtained for all examples using a number of design variables approximately 23 times fewer than the number of analysis mesh elements. This represents a significant reduction in the number of design variables compared with methods using element-wise design variables.

The four benchmark examples were also solved for the complementary problem: minimization of volume, subject to an upper limit on compliance. The proposed method yielded feasible solutions for each problem and similar trends are observed: the objective value increases as the number of design variables decreases and there is a limit to how far the design space can be reduced until the objective value increases significantly. Thus, the results for volume minimization are omitted for brevity.

4.5. 3D example

It is now demonstrated that the proposed method is easily extended to 3D. In 3D, the cutting surface visual analogy cannot be made, but the mathematical description is identical to the 2D case, as Eqs. (2) and (3) remain valid. Second order elements with 27 nodes are used to discretize the "cutting surface" in 3D. All other numerical implementation details and parameters remain the same. The new method in 3D is demonstrated using an example from [30], shown in

Figure 15. The grey zones show areas fixed in the vertical direction and, due to symmetry, only one quarter of the design domain is modeled using $24 \times 24 \times 48$ eight-node trilinear elements for the analysis mesh. The application of the symmetry conditions is sufficient to prevent a singular stiffness matrix. The material properties are Young’s modulus 1.0 and Poisson’s ratio 0.3, and the total load magnitude on the whole model is 256.0. The initial design is the domain completely filled with material.

For this 3D problem it was not found necessary to start with internal holes to obtain a good optimal solution. This is because the behavior of level-set methods is different in 2D and 3D. In 3D, new holes, or non-simply connected shapes can easily be created even when the initial design has no internal holes [23].

The parameters for the smoothed Heaviside are the same as for the 2D examples above.

The example is solved for the minimization of compliance problem, with a volume constraint of 20%. Again, the effect of changing the number of design variables is investigated by solving the problem with different discretization of the ”cutting surface”. The results are summarized in Table 1 and the optimized designs are shown in Figure 16. This shows that again the objective value increases as the number of design variables decreases and that there is a limit to how much the design space can be reduced until the objective increases significantly. For this example, compliance increases by 5.3% from using 4225 to 225 design variables. However, the compliance increase is 37% when going from 225 to 27 design variables. It is also observed that the optimized designs for the three solutions using the most design variables are similar to the solution obtained using an element density-based method [30]. However, the solution using just 27 variables does not have the central lower plate that is important for increasing the stiffness of the stool.

In this paper derivatives are cheap to compute and again an increase in iterations equals a proportional increase in the computational time. However, if finite differences were used to compute derivatives then computational time would reduce with a reduction in the number of design variables, as the required number of FE analyses for convergence would be: 587,414 (4225 variables), 158,862 (637 variables) and 53,110 (225 variables).

Table 2: Summary of results for 3D stool example.

Cutting surface discretization on the quarter model	No. variables	Compliance	Iterations
$6 \times 6 \times 12$	4225	1855	139
$3 \times 3 \times 6$	637	1892	249
$2 \times 2 \times 4$	225	1954	235
$1 \times 1 \times 1$	27	3089	131

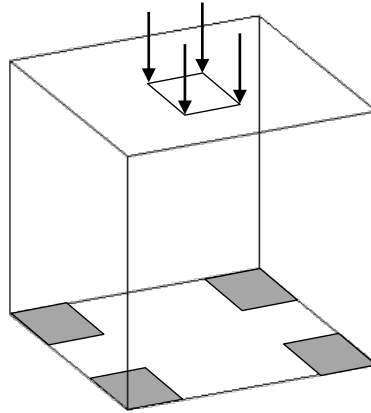


Figure 15: 3D stool example.

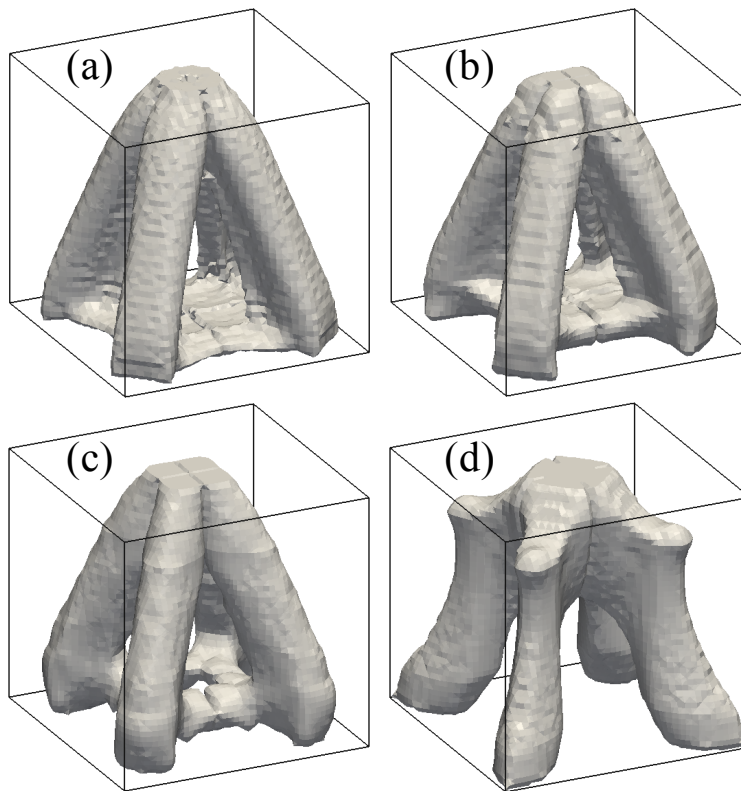


Figure 16: 3D stool compliance solutions, with cutting surface discretizations on the quarter model of: a) $6 \times 6 \times 12$, b) $3 \times 3 \times 6$, c) $2 \times 2 \times 4$, and d) $1 \times 1 \times 1$ elements.

4.6. Compliant mechanism examples

The new method is now used to solve two compliant mechanism problems taken from [17]. The examples are an inverter, Figure 17a, and an amplifier Figure 17b. The objective is to maximize the output displacement, subject to a volume constraint equal to 20% of the design domain. For the inverter, output displacement is defined positive in the opposite direction to the input force and for the amplifier it is defined positive downward, for an input force acting in the left direction. Symmetry is exploited for both examples and only the lower half is modeled. The input force for both examples is 10, material properties are Young’s modulus 1.0 and Poisson’s ratio 0.3 and other parameters are summarized in Table 3. A continuation strategy is used to solve the compliant mechanism problems, where the smoothing length is adapted during optimization (as discussed in Section 3.2). The initial value is $\Delta = 2h$ and the lower limit is $\Delta = 0.5h$. A value of $\rho_{min} = 10^{-2}$ is chosen.

Table 3: Parameters for compliant mechanism examples.

Example	k_{in}	k_{out}	h	Analysis mesh
Inverter	0.001	1.0	0.5	160×160
Amplifier	0.005	1.0	1.0	160×80

Both problems are solved using two cutting surface mesh discretizations and the results are shown in Figure 18, Figure 19 and Table 4. These solutions are similar to those obtained by other methods [18, 31]. The two solutions obtained for the inverter with different numbers of design variables are similar (with the same topology), although the objective decreases by 10.3% when going from 4225 to 1089 design variables. However, the objective increase for the 4225 variable solution is partly obtained by exploiting the numerical discretization, where a compliant hinge is created around one node in the analysis mesh [32]. It would be interesting to add minimum length scale control to the proposed method to avoid this issue.

The solutions for the amplifier are also similar, although slightly more detail and numerical artifacts are present near the output point for the design using 32×16 cutting surface elements. For this example, the objective decreases by only 2.7% when reducing the number of design variables. These two compliant mechanism examples demonstrate that the proposed method can be used to solve problems beyond those involving just compliance and volume. Also, the same trend of the objective function getting worse (decreasing in this case) with a reduction in the number of design variables is observed.

Once again, we can calculate the required number of FE analyses if finite differences were used to compute derivatives. This gives an estimate on the reduction in computational cost when the design space is reduced and finite differences are used. For the inverter, the number of FE analyses would reduce from 798,714 to 184,210 (a 77% reduction). For the amplifier the reduction is from 178,118 to 56,762, giving a potential 68% reduction in computational cost.

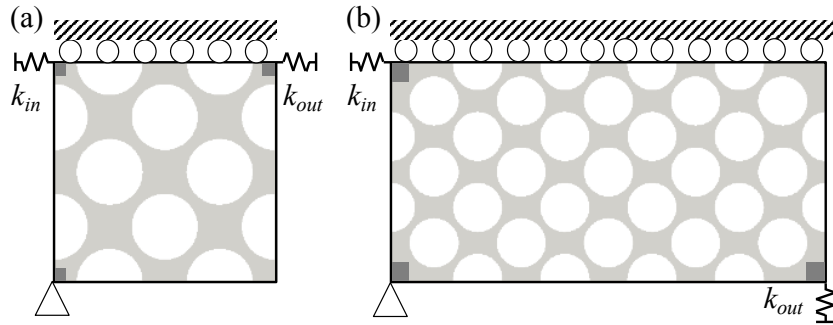


Figure 17: Compliant mechanism examples: a) inverter, b) amplifier.

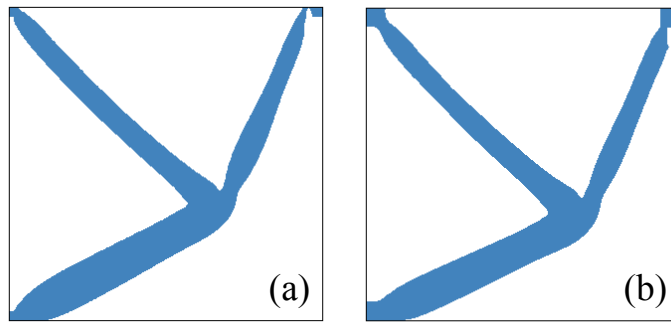


Figure 18: Inverter solutions with cutting surface discretizations: a) 32×32 and b) 16×16 elements.

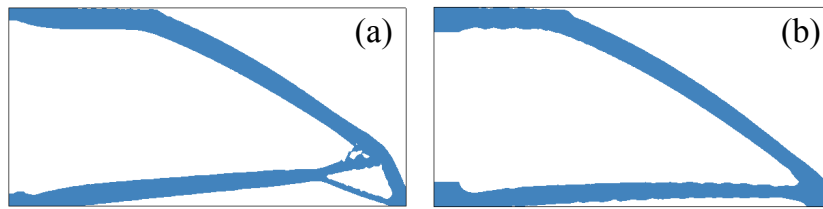


Figure 19: Amplifier solutions with cutting surface discretizations: a) 32×16 and b) 16×8 elements.

Table 4: Summary of results for compliant mechanism examples.

Example	Cutting surface discretization	No. variables	Output displacement	Iterations
Inverter	32×32	4225	2.91	189
	16×16	1089	2.61	169
Amplifier	32×16	2145	5.94	83
	16×8	561	5.78	101

5. Discussion

The potential computational cost saving when using finite differences is discussed for the examples shown in the previous section. It is demonstrated that when the number of design variables is decreased a time saving of at least 60% can be expected. For problems solved using gradient-free methods, we will assume that the number of function evaluations (FE analyses) required for convergence follows an exponential growth function: $m = a(1 + r)^n$, where m is the number of evaluations, n is the number of design variables, r is the growth rate and a is a constant. The ratio of function evaluations using two different design parameterizations is then: $m_1/m_2 = (1 + r)^{(n_1 - n_2)}$. If $r < 10^{-2}$ then $m_1/m_2 \approx \exp[n_1 r (1 - n_2/n_1)]$. Thus, if we know the growth rate, r , (determined by the optimizer performance), and the number of design variables in the original problem, n_1 , (e.g. the number of finite elements or nodes), then we can calculate the ratio of function evaluations, m_1/m_2 , for a given reduction in the number of design variables, n_1/n_2 . Figure 20 plots m_1/m_2 against $r \times n_1$ for different levels of design space reduction (n_1/n_2). Although the potential computational cost reduction is dependent on the optimizer, it can be seen that significant reductions are possible with the order of magnitude reduction in design space reported in this paper. To illustrate this, let's assume the optimizer has a growth rate of $r = 10^{-5}$ and the original problem has $n_1 = 4 \times 10^5$ design variables, so $r \times n_1 = 4$. If we can reduce the number of variables by a similar amount demonstrated by the method proposed in this paper, i.e. $n_1/n_2 = 20$, then $m_1/m_2 \approx 45$, representing about a 45 times reduction in the computational cost. The values of r and n_1 are arbitrarily chosen for illustration and will depend on the optimizer and original problem design space. Also, the actual computational time will still depend on the time for each function evaluation and further reduction can be gained by combining design space reduction with methods to improve analysis efficiency (see for example [11, 12, 13]).

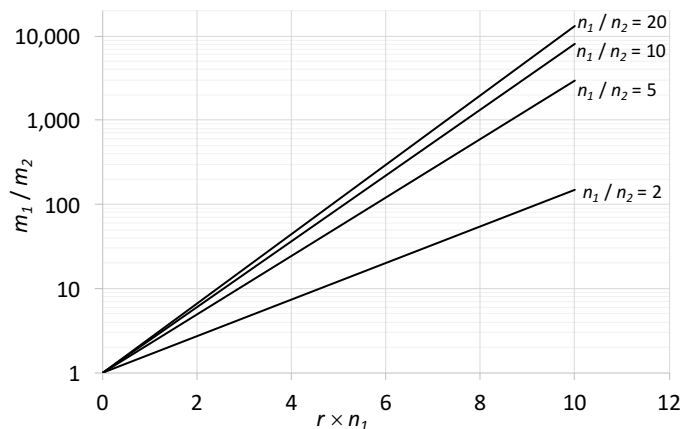


Figure 20: Potential computational cost reduction when using a gradient-free optimizer.

It is observed that some solutions still contain a small amount of numerical artifacts, such as small holes (e.g. Figure 11a) and single node hinges (Figure 18a). These occur due to the use of the smoothed Heaviside function to create the FE matrices from the implicit function. The smoothing length creates small areas of intermediate stiffness material, which the optimizer exploits to create small areas of intermediate stiffness (seen as holes when $\phi \geq 0$ is plotted) and the apparent single node hinge. These types of numerical artifacts can be eliminated using methods for minimum and maximum length scale control. Thus, future research will focus on developing a length scale control technique for the method proposed in this paper.

It was noted in the introduction that there are other methods that can potentially be used to reduce the number of design variables and produce solutions with smooth boundaries. The spectral level-set method can produce good solutions with very few design variables [19]. However, the global nature of the parameterization means that local features cannot be easily represented with a coarse design description [3]. The use of the signed-distance function in the proposed method allows local features to be included in the parameterization, even with a coarse design space. When using a radial basis function parameterization to solve a pipe bend fluid flow problem, Kreissl et al. [33] obtained a reasonable solution using about 7.5 times fewer design variables than nodes in the fluid analysis mesh. Results using fewer design variables were not reported and the author is unaware of similar studies using radial basis functions. Thus, no further comment or comparison is made. The parameterization method using moving morphable geometric shapes (or components) can produce good solutions using very few design variables [16]. However, the solution is dependent on the number and type of components used for the parameterization, which has to be determined *a priori*.

The Heaviside projection method was investigated for potential design space reduction by Guest and Smith Genut [28]. They compared two approaches: static reduction, using a fixed number of design variables (which is the approach in this paper) and an adaptive scheme that starts with a small number of design variables that can increase or decrease during optimization. Some similar results to those presented in this paper were found when using the static scheme in that reducing the number of design variables usually leads to a worse objective function and that there is a limit to how far the design space can be reduced before the objective function becomes significantly worse. From the 2D compliance results presented in [28] this limit is approximately 10 times fewer design variables than the number of analysis mesh elements or nodes. The results using the adaptive scheme show a similar computational time to the static scheme using 10 times fewer design variables. For similar 2D compliance examples, the method proposed in this paper demonstrated a 23 times reduction in the number of design variables and therefore performs better in this respect than the Heaviside projection method. However, the potential computational cost savings for problems with analytical derivatives are not significant. Thus, it is important to discuss the potential performance of the method when using a gradient-free optimizer.

Guest and Smith Genut [28] also investigated the effect of reducing the number of design variables when using a genetic algorithm. They found that, for a fixed number of FE analyses, the Heaviside projection scheme with an adaptive design variable field consistently produced better solutions compared with element-based and full-resolution Heaviside projection parameterizations. This is attributed to faster convergence when fewer design variables are used and that the Heaviside projection produces more feasible solutions, compared with the element-based parameterization. The proposed method in this paper has similar qualities, in that the number of design variables can be significantly reduced without significant loss in objective function and that the number of infeasible solutions found (due to disconnected structure) may be small because the underlying structure in the signed-distance function can promote a fully connected, feasible structure. Thus, the proposed method is promising for use with genetic algorithms and other gradient-free methods. However, this will be investigated in detail in future work.

6. Conclusions

A new topology optimization method is proposed that uses an implicit signed-distance function and cutting surface, where the structure boundary is defined by their intersection. During optimization, the implicit function is fixed and the cutting surface is modified to change the position and connectivity of the boundary. The cutting surface is parameterized using finite element shape functions and the nodal values become the design variables during optimization. The analysis and cutting surface meshes are decoupled and the cutting surface mesh uses fewer elements than the analysis mesh, thus reducing the number of design variables compared with methods that use element-wise design variables. In this paper, problems are solved using a gradient based optimizer, facilitated by the use of a smoothed Heaviside function to generate finite element matrices from the implicit function, although any method that can achieve this can also be used with the cutting surface parameterization.

The performance of the method was investigated and it was found that using serendipity type elements (Q8) for the cutting surface discretization was unsatisfactory due to poor convergence and that using quadratic Q9 elements generally gave better results than using linear Q4 elements. It was also observed that the use of the signed-distance function in the parameterization provided some regularization, as it eliminated large areas of intermediate stiffness material and afforded a degree of mesh independency. The investigation also highlighted that the full-resolution parameterization (one design variable per analysis mesh node) produced solutions with a significant amount of checkerboard-type numerical artifacts. Finally, it was shown that solutions are affected by the initial design and it was recommended that an initial design with many holes is used.

The effect of reducing the number of design variables was extensively investigated using 2D and 3D minimization of compliance and 2D compliant mechanism problems taken from the literature. In each case, feasible, solutions

with smooth boundaries were obtained that agree well with solutions obtained using other methods. However, the reduction in design space could result in rougher boundaries and some solutions still contained a small amount of numerical artifacts, which could be eliminated by introducing length scale control. It was observed that there is a limit to how far the number of design variables can be reduced before the objective value becomes significantly worse. Despite this, good solutions to all problems are obtained using significantly fewer design variables compared with the number of analysis mesh elements, which can provide a significant reduction in computational cost when using finite differences or gradient-free methods.

Appendix A. Sensitivity calculation

This appendix details the sensitivity calculation for compliance. A similar procedure can be used to calculate the sensitivity for other functions. Compliance, C , can be written as:

$$C = \mathbf{u}^T \mathbf{K}_s \mathbf{u} \quad (\text{A.1})$$

where \mathbf{u} is the displacement vector and \mathbf{K}_s is the global stiffness matrix:

$$\mathbf{K}_s = \sum_{j=1}^{N_e} \mathbf{K}_{e,j} \quad (\text{A.2})$$

where $\mathbf{K}_{e,j}$ is the j^{th} element stiffness matrix, as defined by Eq. (4) and N_e is the number of analysis mesh elements. It is well-known that compliance is self-adjoint and the derivative with respect to a design variable b_k is:

$$dC/db_k = \mathbf{u}^T \frac{d\mathbf{K}_s}{db_k} \mathbf{u} = \mathbf{u}^T \left(\sum_{j=1}^{N_e} \frac{d\mathbf{K}_{e,j}}{db_k} \right) \mathbf{u} \quad (\text{A.3})$$

where the derivative of the element stiffness matrix is:

$$\frac{d\mathbf{K}_e}{db_k} = \sum_{i=1}^{N_g} (\mathbf{B}_i^T \mathbf{E}_i \mathbf{B}_i w_i) \frac{dH}{d\phi(x_i)} \frac{d\phi(x_i)}{db_k} \quad (\text{A.4})$$

Using Eq. (3):

$$\phi(x_i) = \mathbf{N}_i \boldsymbol{\phi} = \mathbf{N}_i (\boldsymbol{\alpha} - \mathbf{A} \cdot \mathbf{b}) \quad (\text{A.5})$$

where \mathbf{N}_i is the shape function interpolation vector that defines the value of ϕ at integration point i from the nodal ϕ values. The derivative is:

$$d\phi(x_i)/db_k = -\mathbf{N}_i \cdot \mathbf{a}_k \quad (\text{A.6})$$

where \mathbf{a}_k is the k^{th} column in \mathbf{A} . The derivative of the Heaviside function with respect to $\phi(x_i)$ is:

$$\frac{dH(\phi)}{d\phi(x_i)} = \begin{cases} \frac{3(1-\rho_{min})}{4} \left(\frac{1}{\Delta} - \frac{\phi(x_i)^2}{\Delta^3} \right) & -\Delta \leq \phi < \Delta, \\ 0 & \text{else} \end{cases} \quad (\text{A.7})$$

Equations (A.6) and (A.7) are then substituted into Eq. (A.4), which is then substituted into Eq. (A.3) to complete the calculation. Note that most of the components of the sensitivity calculation remain constant throughout the optimization and only Eq. (A.7) needs to be computed each iteration before the substitutions into Eqs. (A.4) and (A.3).

Acknowledgements

The author would like to thank 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/>).

References

- [1] M. Bendsoe, O. Sigmund, *Topology optimization: Theory, Methods and Applications*, Springer-Verlag, Germany, 2004.
- [2] J. D. Deaton, R. V. Grandhi, A survey of structural and multidisciplinary continuum topology optimization: post 2000, *Structural and Multidisciplinary Optimization* 49 (1) (2014) 1–38.
- [3] N. P. van Dijk, K. Maute, M. Langelaar, F. van Keulen, Level-set methods for structural topology optimization: a review, *Structural and Multidisciplinary Optimization* 48 (3) (2013) 437–472.
- [4] C. Le, J. Norato, T. Bruns, C. Ha, D. Tortorelli, Stress-based topology optimization for continua, *Structural and Multidisciplinary Optimization* 41 (4) (2010) 605–620.
- [5] F. Duddeck, S. Hunkeler, P. Lozano, E. Wehrle, D. Zeng, Topology optimization for crashworthiness of thin-walled structures under axial impact using hybrid cellular automata, *Structural and Multidisciplinary Optimization* 54 (3) (2016) 415–428.
- [6] J. Du, N. Olhoff, Topological optimization of continuum structures with design-dependent surface loading—part i: new computational approach for 2d problems, *Structural and Multidisciplinary Optimization* 27 (3) (2004) 151–165.
- [7] L. H. Olesen, F. Okkels, H. Bruus, A high-level programming-language implementation of topology optimization applied to steady-state Navier-Stokes flow, *International Journal for Numerical Methods in Engineering* 65 (7) (2006) 975–1001. doi:10.1002/nme.1468.
- [8] L. Van Miegroet, P. Duysinx, Stress concentration minimization of 2d filets using X-FEM and level set description, *Structural and Multidisciplinary Optimization* 33 (4) (2007) 425–438. doi:10.1007/s00158-006-0091-1.
- [9] B. K. Stanford, P. D. Dunning, Optimal topology of aircraft rib and spar structures under aeroelastic loads, *Journal of Aircraft* 52 (4) (2015) 1298–1311.

- [10] O. Sigmund, On the usefulness of non-gradient approaches in topology optimization, *Structural and Multidisciplinary Optimization* 43 (5) (2011) 589–596. doi:10.1007/s00158-011-0638-7.
- [11] O. Amir, M. Stolpe, O. Sigmund, Efficient use of iterative solvers in nested topology optimization, *Structural and Multidisciplinary Optimization* 42 (1) (2010) 55–72.
- [12] S. Wang, E. de Sturler, G. H. Paulino, Large-scale topology optimization using preconditioned krylov subspace methods with recycling, *International Journal for Numerical Methods in Engineering* 69 (12) (2007) 2441–2468.
- [13] J. Zhao, C. Wang, Dynamic response topology optimization in the time domain using model reduction method, *Structural and Multidisciplinary Optimization* 53 (1) (2016) 101–114. doi:10.1007/s00158-015-1328-7.
- [14] M. Yoshimura, K. Shimoyama, T. Misaka, S. Obayashi, Topology optimization of fluid problems using genetic algorithm assisted by the kriging model, *International Journal for Numerical Methods in Engineering* (2016) n/a–n/adoi:10.1002/nme.5295.
- [15] M. Bujny, N. Aulig, M. Olhofer, F. Duddeck, Evolutionary level set method for crashworthiness topology optimisation, in: M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (Eds.), *ECCOMAS 2016, Crete Island, Greece, 2016*, pp. 1–14.
- [16] X. Guo, W. Zhang, W. Zhong, Doing topology optimization explicitly and geometrically - a new moving morphable components based framework, *Journal of Applied Mechanics* 81 (8) (2014) 081009.
- [17] S. Rojas-Labanda, M. Stolpe, Benchmarking optimization solvers for structural topology optimization, *Structural and Multidisciplinary Optimization* 52 (3) (2015) 527–547.
- [18] Z. Luo, L. Tong, M. Y. Wang, S. Wang, Shape and topology optimization of compliant mechanisms using a parameterization level set method, *Journal of Computational Physics* 227 (1) (2007) 680–705.
- [19] A. A. Gomes, A. Suleman, Application of spectral level set methodology in topology optimization, *Structural and Multidisciplinary Optimization* 31 (6) (2006) 430–443.
- [20] N. van Dijk, M. Langelaar, F. Keulen, Explicit level-set-based topology optimization using an exact Heaviside function and consistent sensitivity analysis, *International Journal for Numerical Methods in Engineering* 91 (1) (2012) 67–97.
- [21] J. K. Guest, Topology optimization with multiple phase projection, *Computer Methods in Applied Mechanics and Engineering* 199 (1) (2009) 123–135.
- [22] P. Dunning, Topology optimization with an implicit function and parameterized cutting surface, in: M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (Eds.), *ECCOMAS 2016, Crete Island, Greece, 2016*, pp. 1–12.

- [23] G. Allaire, F. Jouve, A. Toader, Structural optimization using sensitivity analysis and a level-set method, *Journal of Computational Physics* 194 (1) (2004) 363–393.
- [24] M. Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, *Computer Methods In Applied Mechanics and Engineering* 192 (1-2) (2003) 227–246.
- [25] P. Wei, M. Y. Wang, X. Xing, A study on X-FEM in continuum structural optimization using a level set model, *Computer-Aided Design* 42 (8) (2010) 708–719.
- [26] S. Wang, M. Y. Wang, A moving superimposed finite element method for structural topology optimization, *International Journal for Numerical Methods in Engineering* 65 (11) (2006) 1892–1922.
- [27] J. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1999.
- [28] J. K. Guest, L. C. Smith Genut, Reducing dimensionality in topology optimization using adaptive design variable fields, *International journal for numerical methods in engineering* 81 (8) (2010) 1019–1045.
- [29] The NLOpt nonlinear optimization package, <http://ab-initio.mit.edu/nlopt> (2014).
- [30] T. Borrvall, J. Petersson, Large-scale topology optimization in 3d using parallel computing, *Computer Methods in Applied Mechanics and Engineering* 190 (46) (2001) 6201–6229.
- [31] O. Sigmund, On the design of compliant mechanisms using topology optimization, *Journal of Structural Mechanics* 25 (4) (1997) 493–524.
- [32] T. A. Poulsen, A simple scheme to prevent checkerboard patterns and one-node connected hinges in topology optimization, *Structural and Multidisciplinary Optimization* 24 (5) (2002) 396–399.
- [33] S. Kreissl, G. Pingen, K. Maute, An explicit level set approach for generalized shape optimization of fluids with the lattice boltzmann method, *International Journal for Numerical Methods in Fluids* 65 (5) (2011) 496–519.