Higher-order multi-resolution topology optimization using the finite cell method

Jeroen P. Groen[a,1,*], Matthijs Langelaar[b], Ole Sigmund[a], Martin Ruess[c]

[a] *Department of Mechanical Engineering, Solid Mechanics, Technical University of Denmark*
[b] *Faculty of Mechanical, Maritime & Materials Engineering (3mE), Delft University of Technology*
[c] *School of Engineering, University of Glasgow*

Abstract

This article presents a detailed study on the potential and limitations of performing higher-order multi-resolution topology optimization with the finite cell method. To circumvent stiffness overestimation in high-contrast topologies, a length-scale is applied on the solution using filter methods. The relations between stiffness overestimation, the analysis system and the applied length-scale are examined, while a high-resolution topology is maintained. The computational cost associated with nested topology optimization is reduced significantly compared to the use of first-order finite elements. This reduction is caused by exploiting the decoupling of density- and analysis mesh, and by condensing the higher-order modes out of the stiffness matrix.

Keywords: topology optimization, finite cell method, higher-order FEM

1. Introduction

In the past decades density-based topology optimization has become a mature design method, with applications in a variety of industries. Despite the rapid advancements in computer performance, large-scale topology optimization still comes at a high computational cost, dominated by the finite element analysis [1]. In this article, we extensively describe the advantages and limitations of multi-resolution methods to reduce this computational cost. Furthermore, we present an efficient multi-resolution topology optimization algorithm, while maintaining a high-resolution topology.

Currently, the far majority of topology optimization methods, uses the same mesh for both the density description and the analysis. The density elements are then directly mapped on first-order finite elements (FE), which due to their uniform size and shape allow for efficient assembly of the stiffness matrix. The introduction of the Finite Cell Method (FCM) by Parvizian, Düster and Rank showed that a decoupling of the density- and analysis-mesh, in combination with higher-order shape functions, can outperform the above-mentioned approach, for sufficiently smooth density distributions [2, 3, 4]. In this fictitious domain method, multiple density elements are mapped on analysis cells operating at a higher-order basis. This allows for accurate and efficient analysis of data directly derived from X-ray scans or quantitative CT scans without the need for meshing [5, 4, 6, 7],

*Correspondence to: J. P. Groen, Department of Mechanical Engineering, Solid Mechanics, Technical University of Denmark, Nils Koppels Allé, Building 404, 2800 Kgs. Lyngby, Denmark
[1] E-mail: jergro@mek.dtu.dk

however, the method is also very interesting for topology optimization applications. To demonstrate this, the developers of the FCM implemented a heuristic optimization method, showing promising results [8]. In a similar approach, Nguyen et al. report a reduction in computational cost when decoupled meshes, linear shape functions, and a gradient-based optimization method are used [9, 10]. A similar approach is applied in a very popular topology optimization app for hand-held devices [11], while in a more recent study Nguyen et al. have demonstrated the use of higher-order shape functions in combination with this multi-resolution approach [12].

In this paper we go a step further, and dedicate a large part of our attention to the limits at which topology optimization using higher-order multi-resolution methods can be performed, since this will greatly help the method's maturation. It is known that the FCM shows superior convergence compared to first-order finite elements for smooth structures [13], however, in topology optimization highly inhomogeneous topologies belong to the solution space [14]. Filter methods are employed to impose a length-scale on the solution, and we demonstrate that the quality of the corresponding solution depends on both the filter and the properties of the analysis mesh. Using a large number of numerical examples for typical minimum compliance and minimum displacement problems, we find an indication of the parameters for which topology optimization using the FCM results in satisfying topologies (i.e. topologies similar to the ones obtained using standard linear finite elements).

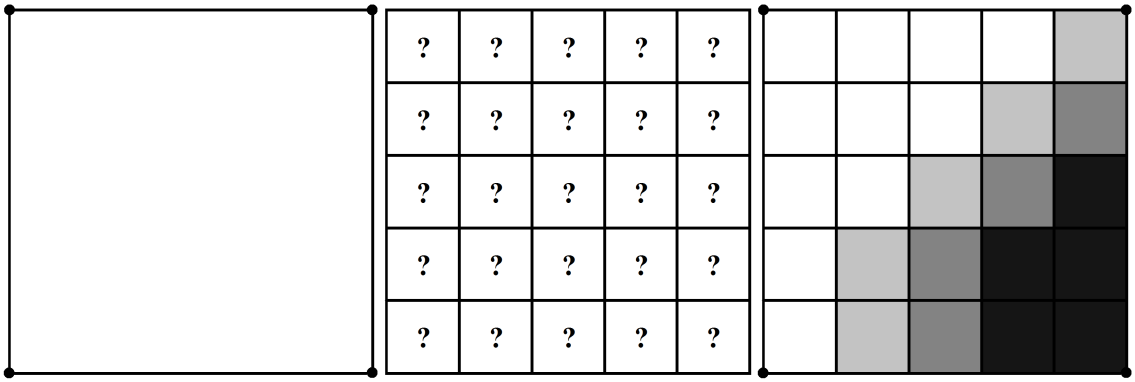
All experiments shown in this paper, have been performed in a MATLAB framework that is created on top of FCMLab: A Finite Cell Research Toolbox for MATLAB, developed by Zander et al [15]. The developed framework is similar to the efficient 88-line topology optimization code [16], and the MATLAB implementation of the Method of Moving Asymptotes (MMA) is used to solve the optimization problem [17]. Using this efficient optimization framework, we present a detailed study on the computational cost of the method, and show its competitiveness compared to the use of linear finite elements. To do so, we present a modification to the FCM, where we condense the internal modes out of the stiffness matrix using the Schur-complement, and show a significant gain in efficiency when higher-order multi-resolution topology optimization is performed.

The paper is organized as follows: The methodology of the FCM is introduced in Section 2. In Section 3 the theory used for the topology optimization is described. The limitations of the method are identified and shown in Section 4. The corresponding tests on the efficiency of the developed method are shown in Section 5, which also includes a detailed discussion on the performance of the method. Finally, Section 6 will present the most important conclusions of this study.

2. The voxel-version of the finite cell method

In the voxel-version of the finite cell method (FCM), separate meshes are used to describe the geometry and to perform the analysis. The elements involved with the analysis mesh are called *cells*, while the geometry (topology) is described by density elements called *voxels* (volume pixels). Both cells and voxels have a uniform shape throughout the design domain and are square in 2D problems and cubic in 3D problems, as can be seen in Figure 1. The distribution of voxels within a cell can be parameterized by the amount of voxels in one cell direction (n_{voxel}), hence the total amount of voxels in a cell (n_{sc}) depends on n_{voxel} and the dimension of the design domain.

The displacement field in a cell with a complex material distribution cannot be interpolated with sufficient accuracy using linear shape functions, therefore the FCM includes the p-version of the FEM [18]. The voxel contributions to the cell stiffness matrix ($\mathbf{k}_c$) are applied using a composed integrations scheme, i.e. the stiffness matrix and load vector are integrated in the voxels and then



(a) Analysis cell (b) Voxels (design elements) (c) Density distribution within a cell

Figure 1: The different types of meshes used in the finite cell method, with $n_{voxel} = 5$

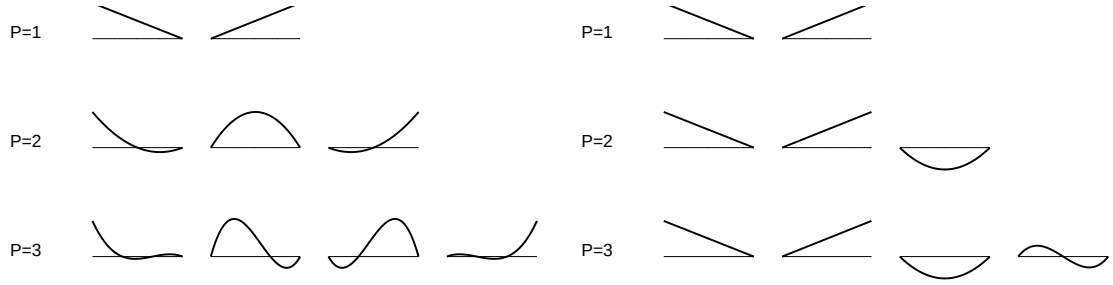
mapped on the cells [2, 4], where the voxel stiffness is interpolated using the solid isotropic material with penalization (SIMP) method [19]:

$$\mathbf{k}_c = \sum_{i=1}^{n_{sc}} \left(E_{min} + \bar{\tilde{\rho}}_i^q (E - E_{min})\right) \mathbf{k}_i^0 \tag{1}$$

where $\bar{\tilde{\rho}}_i$ is the physical density associated with the i^{th} voxel, q is the penalization factor, E is the Young's modulus of a solid voxel, E_{min} is a very small value ($\sim E \cdot 10^{-9}$) to avoid ill-conditioning of the stiffness matrix, and $\mathbf{k}_i^0$ corresponds to the contribution of the i^{th} voxel using a unit stiffness.

2.1. Higher order shape functions

Integrated Legendre polynomials are used to form the higher-order basis. Contrary to Lagrange polynomials, Legendre polynomials are hierarchic, i.e. the shape functions for polynomial degree p are included in the approximation space when degree $p + 1$ is used, as can be seen in Figure 2.



(a) Lagrange polynomials (b) Integrated Legendre polynomials

Figure 2: Set of one-dimensional standard (left) and hierarchic (right) shape function for $p = 1, 2, 3$ [20]

The corresponding one dimensional set of shape functions can be defined as:

$$\begin{aligned} N_1(\xi) &= \frac{1}{2}(1-\xi) \\ N_2(\xi) &= \frac{1}{2}(1+\xi) \\ N_i(\xi) &= \phi_{i-1}(\xi), \quad i = 3, 4, ..., p+1 \end{aligned} \tag{2}$$

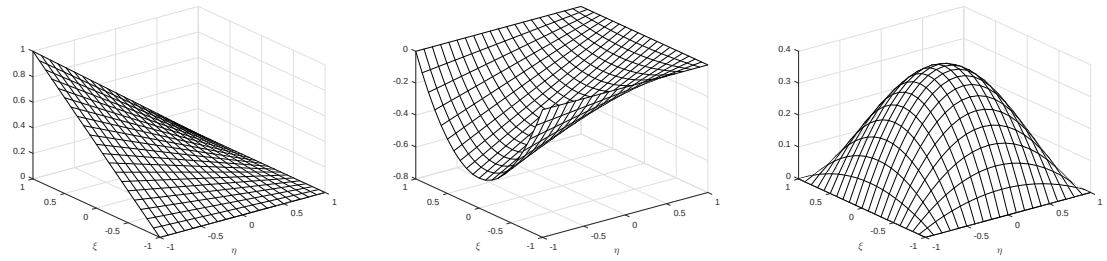
where $N_i(\xi)$ corresponds to the i^{th} shape function, and where ϕ corresponds to an integrated Legendre polynomial. With the integrated Legendre polynomials as basis functions, the displacement field can be interpolated:

$$u(\xi) = N_1(\xi)u_1 + N_2(\xi)u_2 + \sum_{i=3}^{p+1} N_i(\xi)u_i \tag{3}$$

Here u_1 and u_2 correspond to the nodal displacements, while u_i correspond to the amplitudes of the higher-order shape functions. The one-dimensional shape functions can be used in two- or three-dimensional problems, by combining the bases in the *tensor product space* [15]:

$$\begin{aligned} N^{2D}_{i,j}(\xi,\eta) &= N^{1D}_i(\xi)N^{1D}_j(\eta), \quad i,j = 1,2,...,p+1 \\ N^{3D}_{i,j,k}(\xi,\eta,\zeta) &= N^{2D}_{i,j}(\xi,\eta)N^{1D}_k(\zeta) \quad i,j,k = 1,2,...,p+1 \end{aligned} \tag{4}$$

In the current code, the full *tensor product space* was used, due to its support in FCMLab [15]. Alternatively, the *trunk space*, which is a deflated version of the *tensor product space*, will produce an equivalent solution quality with less degrees of freedom, especially for higher-order polynomial degrees [18, 20]. On a standard quadrilateral element, three different types of modes can be distinguished, shown in Figure 3. The *nodal modes*, and *edge modes* are shared with adjacent cells while the *internal modes* are local to one cell.

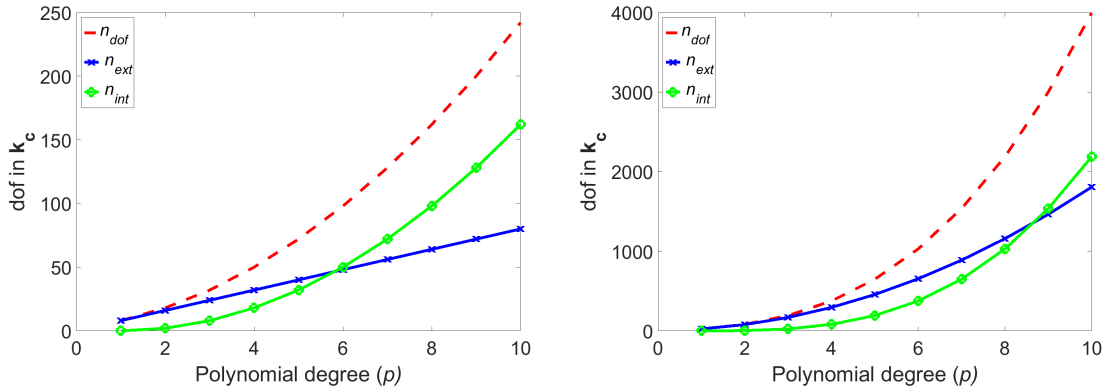


(a) Nodal-mode (b) Edge-mode (c) Internal-mode

Figure 3: 2-Dimensional mode types [15]

2.2. Static condensation

A disadvantage of a higher-order basis is the large amount of internal modes with increasing p-degree, i.e. modes that are specific to only one cell. Figure 4 shows the number of internal modes and the total number of modes at different p, for both 2D, and 3D problems.



(a) 2D problems

(b) 3D problems

Figure 4: Number of internal degrees of freedom (n_{int}) and external degrees of freedom (n_{ext}) in $\mathbf{k}_c$ as a function of p for both 2D and 3D problems

These internal modes can be eliminated from the global system of equations by condensing them out of the stiffness matrix, decreasing the computational cost of the analysis. Furthermore, the procedure results in a drastic decrease in the condition number, which is highly beneficial when iterative solvers are considered [21]. The global system of equations can be re-ordered such that the condensed matrix becomes the Schur complement of $\mathbf{K}_{ii}$ in $\mathbf{K}$.

$$\begin{bmatrix} \mathbf{K}_{ee} & \mathbf{K}_{ei} \\ \mathbf{K}_{ei}^T & \mathbf{K}_{ii} \end{bmatrix} \left\{ \begin{array}{c} \mathbf{U}_e \\ \mathbf{U}_i \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{F}_e \\ \mathbf{F}_i \end{array} \right\} \tag{5}$$

where $\mathbf{U}$ is the displacement vector, $\mathbf{F}$ is the force vector, subscript e denotes the external modes, while subscript i corresponds with the internal modes. From the second row of the system of equations it follows that,

$$\mathbf{U}_i = \mathbf{K}_{ii}^{-1}[\mathbf{F}_i - \mathbf{K}_{ei}^T \mathbf{U}_e] \tag{6}$$

Substitution in the first row of Equation 5 yields,

$$[\mathbf{K}_{ee} - \mathbf{K}_{ei}\mathbf{K}_{ii}^{-1}\mathbf{K}_{ei}^T]\mathbf{U}_e = \mathbf{F}_e - \mathbf{K}_{ei}\mathbf{K}_{ii}^{-1}\mathbf{F}_i \tag{7}$$

which is the condensed system of equations,

$$\mathbf{K}^*\mathbf{U}_e = \mathbf{F}^* \tag{8}$$

where $\mathbf{U}_e$ and $\mathbf{F}^*$ denote the condensed systems displacements and loads, respectively. Since the indices of $\mathbf{K}_{ii}$ are purely local to one cell, we can assemble the condensed stiffness matrix ($\mathbf{K}^*$) efficiently by mapping the contributions of the condensed cell stiffness matrices ($\mathbf{k}_c^*$):

$$\mathbf{k}_c^* = [\mathbf{k}_{c,ee} - \mathbf{k}_{c,ei}\mathbf{k}_{c,ii}^{-1}\mathbf{k}_{c,ei}^T] \tag{9}$$

3. Topology optimization

Topology optimization can be seen as a material distribution problem. The goal is to find an optimum material distribution that minimizes an objective function $\hat{F}$. This function is subject to m constraints $\hat{G}_i$, of which the first is generally a volume constraint. The design domain Ω is discretized into *voxels* (volume pixels) to which design variables are assigned, all variables together form the design vector $\boldsymbol{\rho}$. The material distribution is allowed to vary between 0 and 1 for gradient based optimization.

The discretized topology optimization problem will have a mesh-dependent solution. Furthermore, numerical artifacts, similar to the well-known checkerboard patterns, need to be omitted from the solution space. To do so, several established filter methods are used in our model, such as the sensitivity filter, the density filter, the density filter with projection, and robust topology optimization [22]. The description of each of these filters and their corresponding sensitivities can be found in Appendix AppendixA. Using these filter methods the design vector $\boldsymbol{\rho}$ is linked to the physical density in each voxel $\tilde{\bar{\boldsymbol{\rho}}}$, hence the discretized optimization problem can be written as [23]:

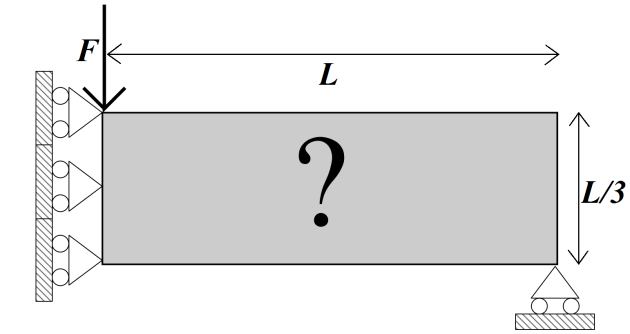
$$\begin{aligned}
\min_{\boldsymbol{\rho}} &: \hat{F}(\boldsymbol{\rho}) = F(\boldsymbol{\rho}, \mathbf{U}_e) \\
\text{s.t.} &: \mathbf{K}^* \mathbf{U}_e = \mathbf{F}^* \\
&: \hat{G}_1(\boldsymbol{\rho}) = \mathbf{v}^T \tilde{\bar{\boldsymbol{\rho}}}(\boldsymbol{\rho}) - V_{max} \leq 0 \\
&: \hat{G}_i(\boldsymbol{\rho}) = G_i(\boldsymbol{\rho}, \mathbf{U}_e) \leq 0, \qquad i = 2, ..., m \\
&: \mathbf{0} \leq \boldsymbol{\rho} \leq \mathbf{1}
\end{aligned} \tag{10}$$

where $\mathbf{v}$ is the vector containing the element volumes, and V_{max} is the maximum allowed volume of the material in the design domain. The optimization problem described above is a nested topology optimization problem, i.e. the equilibrium equations are satisfied for each optimization step using the finite cell method (FCM). For the design update the MATLAB implementation of the method of moving asymptotes is used [17].

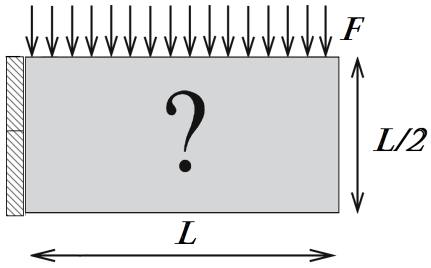
3.1. Definition of test-problems

In this study four representative test-problems are used to test the limitations and the computational cost of performing topology optimization with the FCM. The MBB-beam, which is a typical benchmark example, is used to test the behavior for minimum compliance problems, where, due to the symmetry of the design problem, we model only half of the beam. In another, more challenging minimum compliance problem, a cantilever beam is subject to a uniform traction load. This will demonstrate the performance of the method in regions subject to very small loads. Furthermore, the well-known compliant force inverter is used to identify the performance of the method for mechanism design problems, and the ability of the method to form hinges [24]. Finally, a 3D, and slightly shortened version of the MBB-beam is used to test the computational cost of topology optimization with the FCM. Sketches of the domain and boundary conditions of each of these examples can be found in Figure 5.

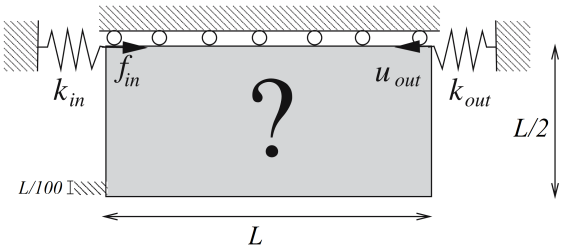
In all 2D-examples, plane stress conditions are assumed. For the 2D-version of the MBB-beam we chose $F = 1$, for the cantilever-beam subject to a distributed load $F = 1/L$, and for both cases $V^* = 0.4$. For the 3D-version the domain is extended to the z-direction with depth $L/2$, while the domain length in the y-direction is increased from $L/3$ to $L/2$, the corresponding maximum volume is $V^* = 0.12$. Similar to the domain, the boundary conditions are extended in the z-direction,



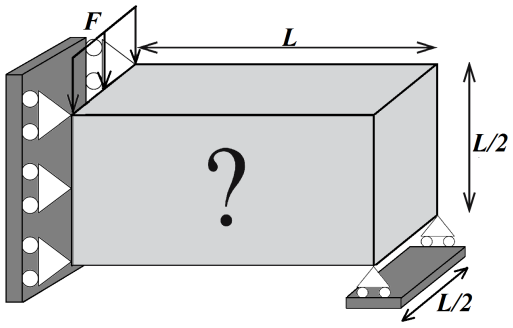
(a) The MBB-beam problem



(b) The cantilever beam, subject to a distributed load



(c) The compliant force inverter problem



(d) The 3D MBB-beam problem

Figure 5: Design domain and boundary conditions for the optimization problems considered.

where the load is now applied as a line load $F = 1/L$. For the mechanism design, the objective is to minimize the displacement u_{out} for a given input force $f_{in} = 1$. The spring coefficients used are $k_{in} = 1$, and $k_{out} = 0.001$, and $V^* = 0.3$. For all optimization problems L is a unit length, a unit Young's modulus is used, $\nu = 0.3$, a penalization factor $q = 3$ is used, leaving just the polynomial degree p, the amount of voxels per cell direction n_{voxel}, and the filter radius R as free parameters. A common formulation to write the objective is,

$$\min_{\boldsymbol{\rho}} \hat{F}(\boldsymbol{\rho}) = \mathbf{L}_e^T \mathbf{U}_e \tag{11}$$

where vector $\mathbf{L}_e$ takes different forms for the different types of problems. In the minimum compliance problems $\mathbf{L}_e = \mathbf{F}^*$, while for the mechanism design problem $\mathbf{L}_e$ is a vector which contains all zeros except for the index corresponding to u_{out}, which is set to one. The sensitivity of the objective w.r.t. physical density $\bar{\tilde{\rho}}_i$ can be calculated by adjoint sensitivity analysis [25],

$$\frac{\partial \hat{F}}{\partial \bar{\tilde{\rho}}_i} = \boldsymbol{\lambda}_{c,e}^T \Big(\frac{\partial \mathbf{k}_{c,ee}}{\partial \bar{\tilde{\rho}}_i} - \frac{\partial \mathbf{k}_{c,ei}}{\partial \bar{\tilde{\rho}}_i} \mathbf{k}_{c,ii}^{-1} \mathbf{k}_{c,ei}^T + \mathbf{k}_{c,ei} (\mathbf{k}_{c,ii}^{-1} \frac{\partial \mathbf{k}_{c,ii}}{\partial \bar{\tilde{\rho}}_i} \mathbf{k}_{c,ii}^{-1}) \mathbf{k}_{c,ei}^T - \mathbf{k}_{c,ei} \mathbf{k}_{c,ii}^{-1} \frac{\partial \mathbf{k}_{c,ei}^T}{\partial \bar{\tilde{\rho}}_i} \Big) \mathbf{u}_{c,e} \tag{12}$$

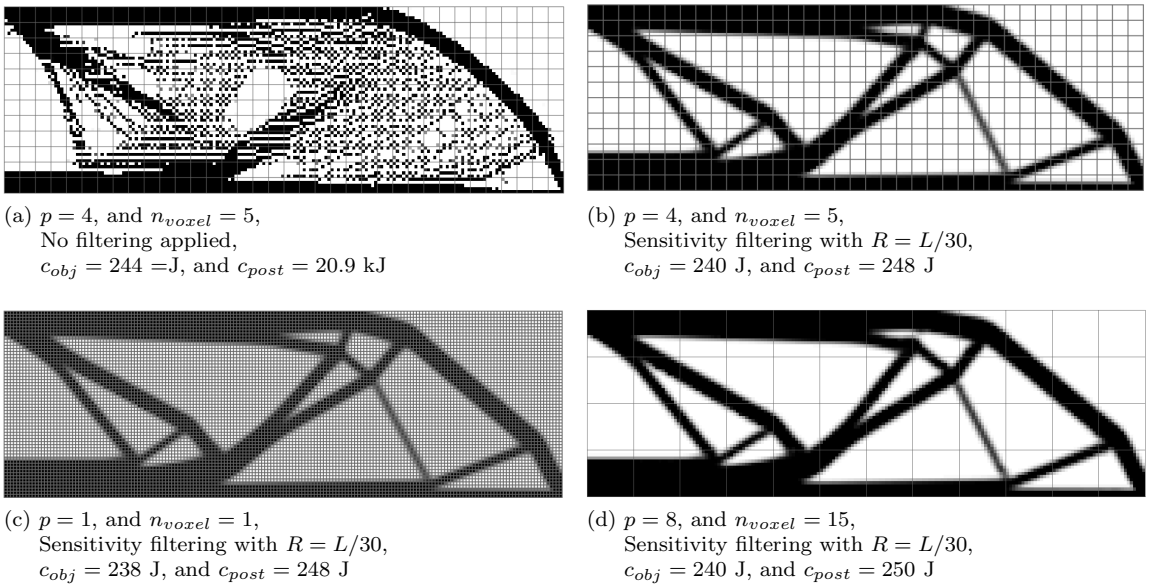
where $\boldsymbol{\lambda}$ is the adjoint vector that can be obtained using

$$\boldsymbol{\lambda}_{c,e}^T = -(\mathbf{K}^*)^{-1} \mathbf{L}_e^T \tag{13}$$

it can be seen that for compliance minimization problems $\boldsymbol{\lambda} = -\mathbf{U}_e$.

4. Limitations of higher-order multi-resolution topology optimization

The solution space in topology optimization examples consists of highly inhomogeneous topologies. When first-order finite elements are utilized these inhomogeneities can be exploited by the optimizer, resulting in checkerboard patterns, where the stiffness of these checkerboards is overestimated [26, 14]. To circumvent this problem, filter methods are used to impose a length-scale on both the material and void. In a similar fashion the heterogeneities in the solution space can be utilized when higher-order multi-resolution topology optimization is performed. To demonstrate this, consider the MBB-beam optimization example, where $p = 4$, and $n_{voxel} = 5$. In Figure 6 (a) a solution is shown, where no filter method is used.



(a) $p = 4$, and $n_{voxel} = 5$,
No filtering applied,
$c_{obj} = 244$ =J, and $c_{post} = 20.9$ kJ

(b) $p = 4$, and $n_{voxel} = 5$,
Sensitivity filtering with $R = L/30$,
$c_{obj} = 240$ J, and $c_{post} = 248$ J

(c) $p = 1$, and $n_{voxel} = 1$,
Sensitivity filtering with $R = L/30$,
$c_{obj} = 238$ J, and $c_{post} = 248$ J

(d) $p = 8$, and $n_{voxel} = 15$,
Sensitivity filtering with $R = L/30$,
$c_{obj} = 240$ J, and $c_{post} = 250$ J

Figure 6: Optimization of the MBB-beam example using a discretization of 180×60 voxels.

4.1. The need for restriction of the solution space

The compliance c_{obj} of the highly heterogeneous solution in Figure 6(a) cannot be estimated correctly by the analysis model ($p = 4$, $n_{voxel} = 5$). The solution space of the displacement field does not allow the sharp variations in strains or displacement corresponding to these heterogeneous cells, hence the displacement is underestimated, making the cells artificially stiff. When the compliance is recalculated in a post-verification step c_{post}, using a highly accurate analysis model ($p = 3$, $n_{voxel} = 1$), it can be seen that the stiffness of the structure is greatly overestimated.

This problem, where the solution field cannot be computed accurately over highly non-smooth regions is well-known, and generally resolved using a local overlay mesh [4]. For topology optimization applications, the topology is not known a-priori and hence we argue that an imposed

length-scale on both solid and void will resolve the problem as well. Here, the length-scale is imposed using the mesh-independent filter methods (see Appendix AppendixA), that regularize the solution. If the filter fully covers an analysis cell, the occurrence of artificially stiff patterns is banned from the solution space. Nevertheless, we argue that the introduction of a length-scale also works well when the filter is much smaller than the analysis cell if p is high enough. The introduction of this small length-scale will make sure that the presence of a density discontinuity within a cell can be captured by a sufficiently accurate analysis model.

In topology optimization problems the goal is to locally maximize the mutual energy term $\boldsymbol{\lambda}_{c,e}^T \mathbf{k}_c^* \mathbf{u}_{c,e}$ (equal to the strain energy density for minimum compliance problems), such that the objective is minimized. Therefore, to obtain well-connected structures it is sufficient that the value of $\boldsymbol{\lambda}_{c,e}^T \mathbf{k}_c^* \mathbf{u}_{c,e}$ is worse for discontinuous structures compared to the value for well-connected designs, and that this negative effect is sufficiently reflected in the design sensitivities. If this is the case, the solution will be directed towards a well-connected design, hence a very accurate representation of the solution field over the discontinuity is not required.

To demonstrate this proposition we go back to the MBB-beam problem of Figure 6 (a) where the strain energy density of the discontinuous cells is greatly overestimated. In Figure 6 (b), it is shown that an imposed length-scale, although even small, already results in an acceptable topology that is exactly similar to the design obtained when first-order FE are used, shown in Figure 6 (c). The introduction of a length-scale also works well for high n_{voxel} in combination with a high polynomial degree, as explained above. This is illustrated by Figure 6 (d), where the cell size is 15 voxel-lengths h, while the filter radius is just $2h$.

A relation exists between the filter method on the one side, and the quality of the analysis model on the other side, determining whether an acceptable solution can be obtained. An analysis model of low quality, i.e. low p, high n_{voxel}, combined with a large R may result in an acceptable solution, however, a high quality analysis model with small filter radius R may still overestimate the cell stiffness in the presence of a density discontinuity. In the previous examples the chosen filter-radius $R = L/30$ corresponds to 2 voxel-lengths h. In the following we specify R in terms of h, since this is an indicator for the locally imposed length-scale on the physical density, and therefore directly linked to the occurrence of these artificially stiff patterns.

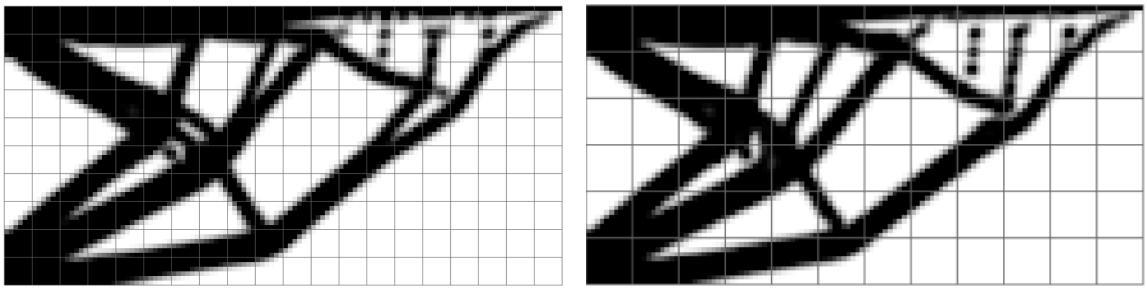
4.2. The validity of optimized solutions

The solutions obtained using the finite cell method (FCM) as analysis model, have to be compared to the well-established solutions obtained using linear finite elements. When these solutions are similar in both performance and topology they are deemed acceptable/satisfactory. To test the performance of the optimized designs, we propose two different methods to check the validity of the results:

- **Post-verification:** It is extremely important to post-verify the objective and constraints, with a high-quality analysis model ($p = 3,\ n_{voxel} = 1$). The artificially stiff patterns shown in Figure 6 (a) can be immediately identified by comparing c_{post} and c_{obj}. Furthermore, post-verification of the objective is the only way to get a fair comparison between generated topologies using different analysis models.

- **Visual check:** The effect of a discontinuity in a low load region does not always translate into a large difference between c_{post} and c_{obj}. For the problem where the cantilever is subject to a distributed load, disconnected patches of material may occur in the low load region of the upper right corner, as can be seen in Figure 7. The effect of these disconnected regions

on the behavior of the structure is small, hence they should be identified via a visual check of the solution.



(a) $p = 3$, and $n_{voxel} = 6$, $c_{obj} = 14.8$ J, and $c_{post} = 16.3$ J

(b) $p = 5$, and $n_{voxel} = 10$, $c_{obj} = 14.8$ J, and $c_{post} = 17.5$ J

Figure 7: Optimization of the distributed load optimization example using a discretization of 120×60 voxels, using sensitivity filtering with $R = 2h$

The visual check may not always work for large 3D examples, e.g. discontinuities can exist within a closed cell. However, in this case one can also consider the strain energy density for the solid voxels in the post-verified solution. If these strain energy densities are several orders of magnitude smaller than for other solid voxels these solid voxels are non-load carrying and indicate the presence of a discontinuity.

4.3. The occurrence of artificially stiff patterns in low-load regions

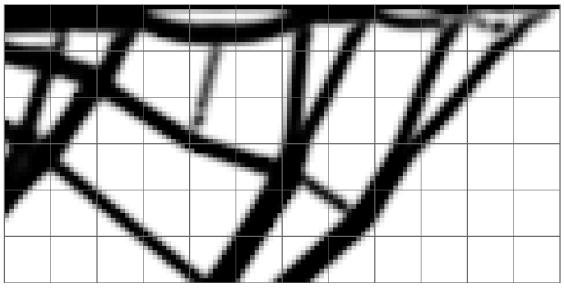
Artificially stiff patterns are most likely to occur in regions where the sensitivity analysis cannot sufficiently reflect the difference between a well-connected and an artificially stiff pattern. This effect can be captured best by the the distributed load problem (see Figure 7), where at the low-load regions the strain energy density is small compared to the rest of the domain. Correspondingly, the sensitivities are very small, hence it can be more beneficial to have a disconnected structure with a slightly lower strain energy density, than a well-connected structure that requires more material Therefore, the analysis model has to have a sufficiently high p, such that the negative effect of an artificially stiff pattern can be reflected in the design sensitivities, and hence the optimizer will end up with a well-connected design.

If a sensitivity or density filter is used, it is also possible that gray material, i.e. material of intermediate density, is introduced at these low-load regions. This happens as well when first-order finite elements are used, see e.g. Figure 8(a). However, if these regions of gray material show some local variation between dense and less-dense voxels, as is the case in Figure 7 (a) and (b), then this is still considered an unacceptable solution. This local variation between dense and less-dense voxels is caused by the optimizer exploiting the analysis model, and will never occur when first-order finite elements are used. Therefore even small variations as in Figure 8(b) are deemed unacceptable.

4.4. The effect of the filter method and filter radius

Maximum design resolution is obtained for a small filter radius R. However, a small R requires a very accurate analysis model, which means that the method will be computationally much more

(a) $p = 1$, and $n_{voxel} = 1$,
$c_{obj} = 14.1$ J, and $c_{post} = 14.3$ J



(b) $p = 6$, and $n_{voxel} = 10$,
$c_{obj} = 14.2$ J, and $c_{post} = 15.6$ J

Figure 8: Close up of the low load regions of the distributed load optimization example for a discretization of 240×120 voxels, using sensitivity filtering with $R = 2h$

expensive than the use of first-order finite elements. Therefore, the smallest value of R has to be found, which allows well-connected solutions for an analysis system of moderate quality. To do so, all 2D experiments have been performed using sensitivity filtering, for different R, p, for a fixed $n_{voxel} = 10$. We have chosen to keep a fixed, large number of voxels per cell to allow for structural members smaller than the cell-size. The lowest polynomial degrees, for each R that do not result in discontinuous structures can be found in Figure 9.

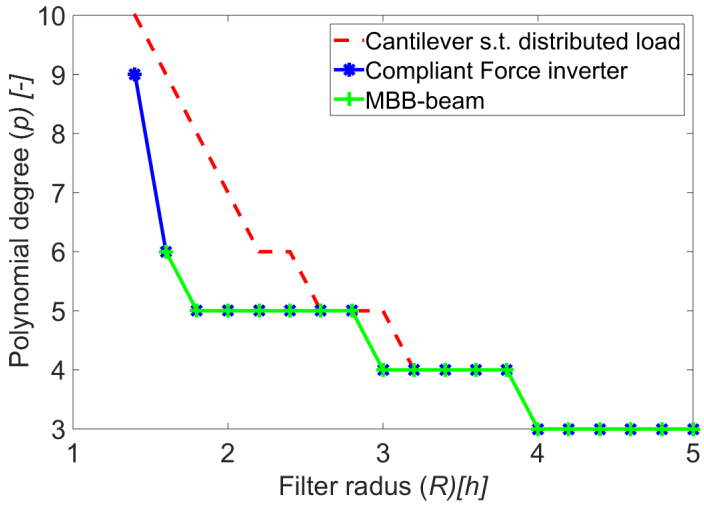


Figure 9: Lowest polynomial degree p that did not result in artificially stiff patterns vs filter radius R, for three different test-problems, using sensitivity filtering

It can be seen that the cantilever beam subject to a distributed load is by far the most critical example, due to the low load or density regions. The MBB-beam and the compliant force inverter example show a similar but less critical behavior for small filter radii. Furthermore, it can be seen that a filter-radius of $1.8h$ or larger requires an analysis system of moderate p-degree. We choose

to use a filter radius of $2h$ in the remainder of this study.

Besides the filter radius the different filter methods have a different effect on the occurrence of artificially stiff patterns. A detailed description of these filter methods can be found Appendix AppendixA. In Figures 8(a) and 10 the solutions obtained for the distributed load example are shown for all different filter methods using $p = 6$, and $n_{voxel} = 10$. As indicated in Figure 9 it is not advised to use these settings, and for all examples the obtained solutions were not deemed acceptable. However, these unacceptable topologies are good indicators of the differences between the filter methods. For both the sensitivity and the density filter the transition between solid and void is gradual, hence the patches of material are not completely disconnected. The density filter using the Heaviside projection can easily end up with an artificially stiff pattern, since it only imposes a length-scale on the solid. Therefore, this filter method will not be considered any further in this study. The modified Heaviside projection filter normally performs better, since it imposes a length-scale on the void. However, since it does not put a length-scale on the solid very thin structural members can occur. In this example one of the structural members ends abruptly, as can be seen in the top right corner. The analysis model is not able to capture this disconnection, and hence this point is artificially stiff. Finally, it can be seen in Figure 10 (d) that the robust formulation imposes a length-scale on both the solid and void. At first sight the structure seems to perform well, however, a close-inspection of the low-load region reveals that the structure performs worse compared to a reference solution obtained with first-order finite elements. The optimizer thus ended up at an artificially stiff local minimum, hence the solution is regarded as unacceptable.

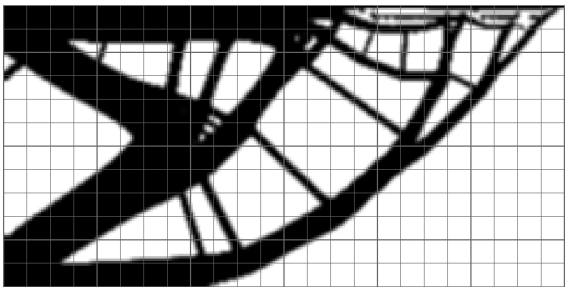
4.5. Experiments on the limits of higher-order multi-resolution topology optimization for R=2h

To get a good overview of the limits at which higher-order multi-resolution topology optimization can be performed, all 2D test-problems were solved for different p, n_{voxel}, for 2 mesh sizes (i.e. coarse and fine) and for the different filter methods (sensitivity filtering, density filtering, density filtering with the modified Heaviside projection, and robust topology optimization) all with $R = 2h$. In total more than a thousand different experiments have been performed, of which the most important observations are summarized below:

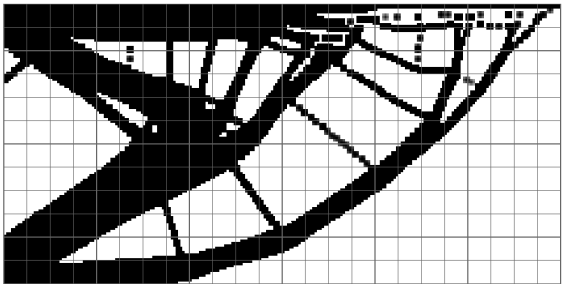
- **Similarities in generated topologies:** The acceptable optimized topologies (i.e. no artificially stiff patterns) are very similar to the optimized topologies obtained when first-order FE are used. Due to the self-adjoint nature of minimum compliance problems, the resulting topologies are almost identical, as can be seen in Figure 6 (b),(c) and (d). The compliant force inverter example is more prone to end up at local minima due to the more complex nature of the objective function. Therefore, the solutions show more variation as can be seen in Figures 11-12, however, their corresponding post-verified objectives are all in the same range.

- **Effect of analysis model on local minima:** The type of analysis model does not seem to have an effect on the optimizer getting stuck in strong local minima. The use of higher-order shape functions allows for hinges at cell nodes or within a cell as can be seen in Figures 11-12. For all the performed experiments, we could not identify that one location for a hinge was favored over another, furthermore, the hinge was able to move freely during the design iterations.

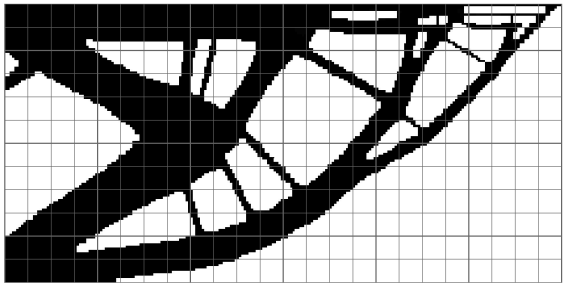
- **Effect of the cell size:** A larger n_{voxel} requires a higher polynomial degree to prevent the formation of discontinuities within a cell. All experiments have shown that the FCM works



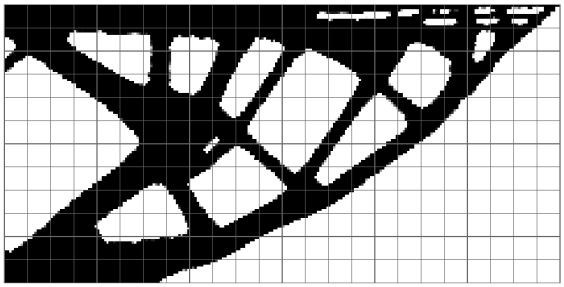
(a) Density filtering, $c_{obj} = 14.5$ J, and $c_{post} = 17.3$ J



(b) Density filtering with Heaviside projection, $c_{obj} = 13.38$ J, and $c_{post} = 330.6$ J



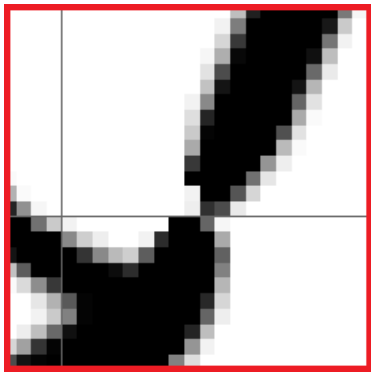
(c) Density filtering with modified Heaviside projection, $c_{obj} = 13.5$ J, and $c_{post} = 33.5$ J



(d) Robust topology optimization with $\eta = 0.25$, $c_{obj} = 14.9$ J, and $c_{post} = 16.3$ J

Figure 10: Optimization of the distributed load optimization example using $p = 6$, $n_{voxel} = 10$, and a discretization of 240×120 voxels. Different filter methods are used with $R = 2h$

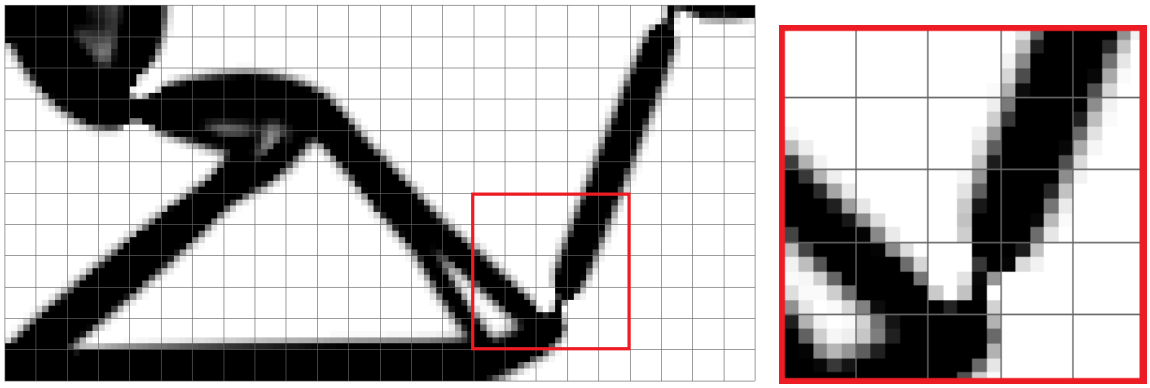
(a) $u_{obj} = -2.33$, and $u_{post} = -2.30$ m



(b) closeup of the hinge

Figure 11: The compliant inverter example for $n_{voxel} = 20$ and $p = 13$. A discretization of 120×60 voxels is used, and sensitivity filtering is applied with $R = 2h$

very well as analysis model up to $n_{voxel} = 15$ or even $n_{voxel} = 20$, see for example Figure 6 (d) and Figure 12. However, a large value of n_{voxel} increases the amount of possibilities in



(a) $u_{obj} = -2.45$, and $u_{post} = -2.35$ m (b) closeup of the hinge

Figure 12: The compliant inverter example for $n_{voxel} = 5$ and $p = 4$. A discretization of 120×60 voxels is used, and sensitivity filtering is applied with $R = 2h$

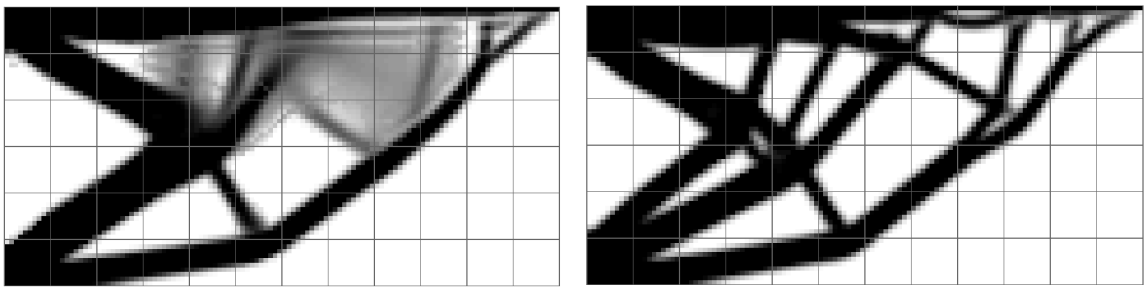
which artificially stiff patterns can occur. Larger values of n_{voxel} thus require more degrees of freedom to prevent the occurrence of these undesired patterns compared to the use of a smaller number of voxels per cell, e.g. $n_{voxel} < 8$. Furthermore, the combination of a high polynomial degree and a high number of voxels, requires the storage of a large number of voxel contributions to the cell stiffness matrix $\mathbf{k}_i^0$, especially for 3D-problems. Thus, the cell size poses a limit on the computational cost, and no values of $n_{voxel} > 15$ are recommended.

- **Effect of filter methods:** The different filter methods have a different effect on the occurrence of artificially stiff patterns. However, we observed great similarities for the settings of the analysis method that did result in artificially stiff patterns. If a type of analysis system produced an acceptable topology using the sensitivity filter, then this analysis system was almost guaranteed to work as well with the density filter, or with the robust topology optimization formulation. In a few cases that work well for the other filter methods, the density filter with the modified Heaviside projection produces artificially stiff patterns, this is because no length-scale is posed on the solid part as is discussed in Section 4.4. Nevertheless, this filter method worked well in almost all cases.

- **Effect of the mesh-size:** All 2D experiments have been performed on two different mesh sizes to find the effect of the mesh-size. The MBB-beam has been modelled on a mesh of 180×60 elements, and on a mesh of 360×180 elements. The cantilever beam is modelled on a coarse mesh of 120×60 elements, and on a fine mesh of 240×120 elements, while for the force-inverter example a mesh of 120×60 and a mesh of 160×80 has been used. No differences could be found between the experiments performed on a coarse mesh and the experiments performed on a fine mesh. The reason is that the filter radius has been linked to the voxel-width h.

- **Artificially stiff patterns in the first iterations:** Artificially stiff patterns can arise during the first $10 - 100$ iterations, as can be seen in Figure 13. Here, the stiffness of the horizontal structural member, that is going through the middle of the top cells, is overestimated. Normally, these patterns are gradually removed by the filter methods, yielding an acceptable

solution. This effect can still be undesired when topology optimization is performed interactively [11]. Furthermore, it is not always the case that the filter methods are able to remove these patterns, recall the topology obtained using the robust formulation in Figure 10(d), where this exact same structural member is part of the final design. This effect is more likely to occur when a cell consists of a large number of voxels, hence too large values for n_{voxel} should be avoided.

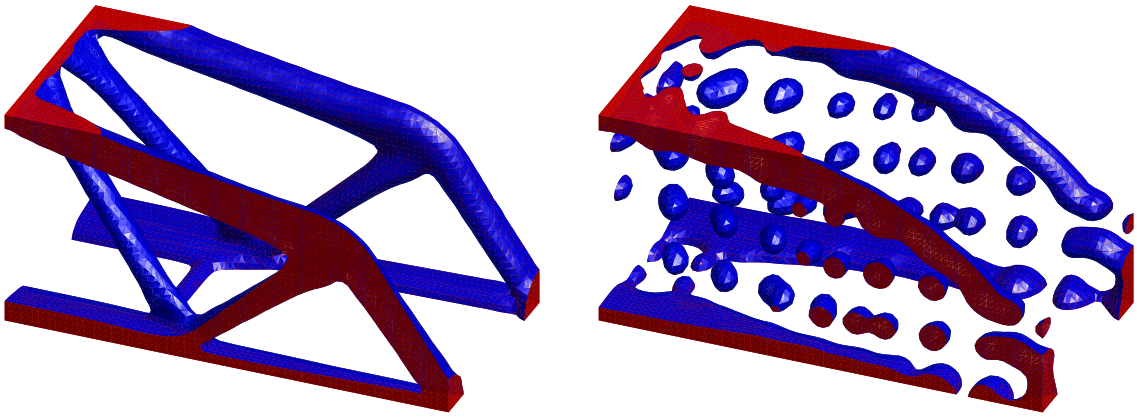


(a) $n_{iter} = 60$ (b) final topology

Figure 13: Optimization of the distributed load optimization example using $p = 7$, $n_{voxel} = 10$, and a discretization of 120×60 voxels. Sensitivity filtering is applied with $R = 2h$. $c_{obj} = 14.9$ J, and $c_{post} = 15.5$ J

- **Artificially stiff patterns in 3D:** Visualization of 3D topologies can be done with a data visualization program, e.g. ParaView. Similar to artificially stiff cells in 2D, the artificially stiff cells in a 3D problem consist of disconnected regions of material, and it is interesting to note that they seem to occur at the exact same settings for the analysis system as in the 2D MBB-beam example. In Figure 14 the difference between an acceptable topology (a), and one with artificially stiff patterns (b) can be seen clearly. The values for c_{post} were obtained with $p = 2$, and $n_{voxel} = 2$, to avoid memory problems in MATLAB.
- **Effect of the optimization problem:** Due to the occurrence of artificially stiff cells in low load regions, the optimization problem with the distributed load is more challenging and does not work for all combinations of p and n_{voxel} that work for the MBB-beam and the force inverter problems. This shows that the choice of analysis model depends on the optimization problem and corresponding boundary conditions. Furthermore, this shows that the distributed load optimization problem is good for finding the limits of a multi-resolution analysis method, and should be considered when a new method is tested.

All experiments showed great similarities in the settings for the analysis model that prevented the occurrence of artificially stiff patterns. Therefore these settings can be summarized in Table 1. The dark gray colored cells correspond to settings which in some or all of the tested experiments resulted in artificially stiff patterns. For the light-gray cells, the occurrence of artificially stiff patterns depends on the type of optimization problem and filter method. In some cases these settings resulted in artificially stiff patterns for the distributed load problem, but not for the other optimization problems. Furthermore, the density filter using modified Heaviside projection is not guaranteed to work. Therefore, these settings should be used with caution, and post-verification of the results is advised to make sure the optimized solution is correct. The plain cells indicate settings yielding



(a) $p = 2$, and $n_{voxel} = 4$,
$c_{obj} = 500$ J, and $c_{post} = 512$ J

(b) $p = 2$, and $n_{voxel} = 10$,
$c_{obj} = 442$ J, and $c_{post} = 20.7$ kJ

Figure 14: Optimization of the 3D MBB-beam example using a discretization of $80 \times 40 \times 40$ voxels, using sensitivity filtering with $R = 2h$

acceptable solutions, and based on our numerical experiments we believe that these settings are representative for the settings that can be safely used in comparable topology optimization problems. However, it is important to note that the presented results are not fully conclusive, and should be interpreted as an indication of which settings are prone to these artificially stiff patterns. The quality of a solution depends on numerous parameters such as, the type of optimization problem, boundary conditions, filter method, etc, and settings indicated here as acceptable might in some cases still result in overestimation of the stiffness.

Table 1: Results of the experiments on the limitations of higher-order multi-resolution topology optimization. The dark gray colored cells indicate settings that yield artificially stiff patterns, while for the light-gray settings, the occurrence of artificially stiff patterns depends on the type of problem and filter method. The settings for the plain cells are likely to result in acceptable topologies.

$n_{voxel} =$	1	2	3	4	5	6	8	10	12	15
$p = 1$										
$p = 2$										
$p = 3$										
$p = 4$										
$p = 5$										
$p = 6$										
$p = 7$										
$p = 8$										
$p = 9$										
$p = 10$										

We can conclude that higher-order multi-resolution topology optimization can work for a large number of different analysis systems. These findings can be summarized as a (conservative) rule of thumb which holds for experiments performed with a filter radius of $R = 2\ h$,

$$p \geq \text{round}\{0.75\ n_{voxel}\} \tag{14}$$

Furthermore, we observed that for $n_{voxel} \geq 8$ there are settings for which we cannot confidently say that they will not result in artificially stiff patterns. In a future study it would be interesting to reduce the amount of light-gray cells by looking into the use of a local overlay mesh at high-contrast regions [4]. Finally, in terms of accuracy, the settings that are indicated by the plain cells, showed numerical convergence properties that are in line with the accuracy expected of the FCM method [4]. The normalized error between c_{post} and c_{obj}, showed that p-refinement is more beneficial than h-refinement.

5. Numerical experiments on the efficiency of the method

The numerical cost of performing topology optimization is dictated by the cost of the repeated solving of the analysis equations. In the framework used for this study, the solution is obtained using the direct solvers implemented in MATLAB. For sparse matrices the amount of operations is $O(n_e^{3/2})$ for 2D problems, and $O(n_e^2)$ for 3D-problems, where n_e is the size of the condensed stiffness matrix [27]. The actual cost of the solution also depends on the order of the approximation, linear shape functions will yield a highly diagonal stiffness matrix, while a higher-order basis will increase the bandwidth, thus increasing the corresponding computational cost. With increasing p, the amount of internal modes $n_{c,i}$ grows exponentially. For 3D optimization examples the inversion of $k_{c,ii}$, which costs $O(n_{c,i}^3)$ operations, can thus have a large influence on the computational cost.

To show the computational cost for different values of p and n_{voxel} two different optimization examples will be discussed. The cantilever beam, subject to a distributed load is used with a discretization of 240×120 voxels, and the 3D version of the MBB-beam is used with a discretization of $80 \times 40 \times 40$ voxels. In both optimization examples sensitivity filtering is applied with $R = 2h$. All optimization examples were solved using a single-core MATLAB code on a standard laptop PC. The results for the 2D optimization example can be seen in Table 2. For the 3D MBB-beam it was not possible to test all settings for n_{voxel}, due to problem discretization enforced by hardware limitations. The settings that could be tested are shown in Table 3. Please note that the setting for the analysis model resulting in artificially stiff patterns, as well as the settings expected to be computationally more costly than first-order FE, have been disregarded, where the settings marked with an asterisk (*) indicate that they are more costly than first-order FE at comparable quality of the analysis results.

It can be seen that decoupling of the density-and analysis-mesh, can be computationally more efficient than topology optimization using first-order FE. For 2D problems an increase in speed of 2.9 can be achieved while maintaining a high-resolution topology without artificially stiff patterns. For the settings where we are unsure whether artificial patterns can occur a speed-up of 3.6 can be achieved. Since this is a minor difference we would recommend to perform topology optimization with settings $(p = 2, n_{voxel} = 3)$, $(p = 3, n_{voxel} = 5)$, $(p = 4, n_{voxel} = 6)$, $(p = 6, n_{voxel} = 8)$ or $(p = 7, n_{voxel} = 10)$. A higher value for p or n_{voxel} shows a slight increase in computational cost due to the inversion of $k_{c,ii}$.

For 3D problems the computational cost can be reduced even more. In this optimization example no artificially stiff patterns have been spotted for the settings in the light-gray cells, as can be seen

Table 2: Normalized speed-up per design iteration for the cantilever beam, subject to a uniform pressure load. A discretization of 240 by 120 voxels is used. The settings marked with an asterisk (*) indicate that they are computationally heavier than first-order FE.

$n_{voxel} =$	1	2	3	4	5	6	8	10	12	15
$p = 1$	1.00	3.04								
$p = 2$	0.36*	1.31	2.44	3.29						
$p = 3$	*	0.71*	1.45	2.22	2.93	3.35				
$p = 4$	*	0.43*	0.92*	1.49	2.00	2.68	3.55			
$p = 5$	*	*	0.60*	1.00	1.48	1.87	2.82	3.59		
$p = 6$	*	*	0.38*	0.69*	1.03	1.41	2.00	2.87	3.36	
$p = 7$	*	*	*	0.49*	0.75*	1.01	1.67	2.22	2.70	
$p = 8$	*	*	*	0.35*	0.54*	0.76*	1.19	1.65	2.14	2.75
$p = 9$	*	*	*	*	0.40*	0.56*	0.93*	1.24	1.65	2.04
$p = 10$	*	*	*	*	0.29*	0.41*	0.63*	0.88*	1.17	1.58

Table 3: Normalized speed-up per design iteration for the 3D MBB-beam example. A discretization of 80 by 40 by 40 voxels is used. The settings marked with an asterisk (*) indicate that they are computationally heavier than first-order FE.

$n_{voxel} =$	1	2	4	5	8	10
$p = 1$	1.00	59.7				
$p = 2$	*	1.72	67.1			
$p = 3$	*	*	12.9	31.8		
$p = 4$	*	*	1.68	7.81	46.4	
$p = 5$	*	*	*	0.83*	18.4	35.0
$p = 6$	*	*	*	*	4.13	14.6

in Figure 14 (a), hence a speed-up in computational time of a factor 67 has been achieved. Still, we would not recommend to use these settings blindly, especially since the more 'robust' analysis model ($p = 3, n_{voxel} = 5$) results in a computational speed-up of 32. It is interesting to note that in the 3D-examples a moderately high p seems to be best. The use of $p > 4$ drastically increases the cost of the inversion of $k_{c,ii}$, furthermore, a higher p increases the number of non-zero elements in K^*, which put a large burden on the memory requirements.

Optimization of the 128,000 design variables using first-oder FE costs around 300 seconds per design iteration. When we use $p = 3$, and $n_{voxel} = 5$ this is reduced to 9.5 seconds per iteration, resulting in a total optimization time of only 34 minutes. We are confident that the optimization time can be reduced even further by using an efficient multigrid pre-conditioned iterative solver [28]. The challenge here will lie in finding an efficient multigrid algorithm suited for higher-order methods, which can not be exploited by the artificially stiff patterns.

6. Conclusion

An efficient approach to perform higher-order multi-resolution topology optimization using voxel-version of the finite cell method (FCM) was presented. The most important finding is that a multi-resolution analysis model can overestimate the stiffness of highly inhomogeneous patterns, in

a similar fashion as first-order FE overestimate the stiffness of checkerboard patterns. This problem can be resolved using mesh-independent filter techniques, where we showed the relation between the quality of the analysis model and the imposed length-scale. Using more than thousand examples we demonstrated the limits at which topology optimization, in the framework of higher-order multi-resolution methods, can be performed. Based on a number of representative test-cases, we have identified settings for the analysis models for which acceptable solutions were achieved.

By reducing the size of the stiffness matrix using static condensation we demonstrated that the computational cost can be decreased significantly compared to the use of first-order FE. In 2D optimization examples an increase of speed of a factor 2.9 was achieved, while in 3D topology optimization problems an even more promising speed-up of 32 was possible. A 3D topology optimization problem with $128,000$ design elements was optimized on a standard PC in 34 minutes, using the direct solvers implemented in MATLAB. This overall promising performance paves the way for further development of the methodology by using efficient multigrid pre-conditioned iterative solvers. We are confident that this will show further reduction in computational cost, and might further reveal the potential of the method for large-scale topology optimization.

Acknowledgement

The first author acknowledges the support of the performed research at the Faculty of Aerospace Engineering, Delft University of Technology. The first and third author received support from the Villum foundation through the NextTop project. The authors also wish to thank Krister Svanberg for the MATLAB MMA code.

AppendixA. Filter methods used in this study

Sensitivity filtering

For the sensitivity filter introduced by Sigmund [24], no difference exists between the physical density and the design vector $\boldsymbol{\rho} = \bar{\tilde{\boldsymbol{\rho}}}$, however, the voxel sensitivity is now based on the sensitivities of the surrounding voxels within a mesh-independent radius r_{min}. These filtered sensitivities are then used to update the design vector,

$$\frac{\partial \hat{F}}{\partial \rho_e} = \frac{1}{\max(\rho_e, 0.001) \sum_{i=1}^{n_e} H_{ei}} \sum_{i=1}^{n_e} H_{ei} \rho_i \frac{\partial \hat{F}}{\partial \bar{\tilde{\rho}}_i} \tag{A.1}$$

where n_e is the number of voxel, and H_{ei} is a linear decaying weighting function. The small number is put in the denominator to avoid division by zero. H_{ei} depends on the distance between the voxel center, and the center of the surrounding voxel, as well as the filter radius r_{min}:

$$H_{ei} = r_{min} - \text{dist}(e, i) \tag{A.2}$$

Density filtering

An alternative to the sensitivity filter is the density filter [29, 30]. The physical density of a voxel $\bar{\tilde{\rho}}_e$ is defined as the weighted average of the design variables of neighboring voxels in r_{min}:

$$\bar{\tilde{\rho}}_e = \frac{1}{\sum_{i=1}^{n_e} H_{ei}} \sum_{i=1}^{n_e} H_{ei} \rho_i \tag{A.3}$$

Using the chain rule the sensitivities with respect to the design variables can be obtained:

$$\frac{\partial \hat{F}}{\partial \rho_e} = \sum_{i=1}^{n_e} \frac{\partial \hat{F}}{\partial \bar{\tilde{\rho}}_i} \frac{\partial \bar{\tilde{\rho}}_i}{\partial \rho_e} \tag{A.4}$$

Density filtering using a projection

A disadvantage of both the sensitivity and density filter is that they introduce regions with intermediate densities. To cope with this a projection scheme base on a smoothened Heaviside function has been introduced by Guest et al [31]. In this approach the filtered densities ($\tilde{\boldsymbol{\rho}}$) using the density filter are now referred to as the 'intermediate design vector'. The physical density $\bar{\tilde{\boldsymbol{\rho}}}$ can be calculated using, the threshold method presented by [32],

$$\bar{\tilde{\rho}}_e = \frac{\tanh(\beta\eta) + \tanh(\beta(\tilde{\rho}_e - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1-\eta))} \tag{A.5}$$

where η is the threshold parameter, and β controls the smoothness of the Heaviside function. For $\beta = 0$ the filter gives exactly the same output as the density filter, whereas when β goes towards infinity the Heaviside function is approximated. For $\eta = 0$, the projection corresponds to the Heaviside projection which applies a length-scale on the material, while $\eta = 1$ corresponds to the modified Heaviside projection introduced by Sigmund that applies a length-scale on the void regions [19]. In the experiments an initial value of $\beta = 1$ is used, which is doubled every 50 iterations until a maximum of $\beta = 64$. The sensitivity of the objective function w.r.t. a design variable ρ_e can be written as:

$$\frac{\partial \hat{F}}{\partial \rho_e} = \sum_{i=1}^{n_e} \frac{\partial \hat{F}}{\partial \bar{\tilde{\rho}}_i} \frac{\partial \bar{\tilde{\rho}}_i}{\partial \tilde{\rho}_i} \frac{\partial \tilde{\rho}_i}{\partial \rho_e} \tag{A.6}$$

Robust topology optimization

Robust topology optimization has been introduced by Sigmund as a method to perform manufacturing tolerant topology optimization [33]. Small changes in manufacturing should not lead to large changes in functionality. Another positive effect of this method is that it is able to put a length-scale on both the solid and the void material, eliminating the longstanding problem of one-node connected hinges [34].

In robust topology optimization three different designs are formulated based on the same design vector. These are a dilated ($\bar{\tilde{\rho}}^d$), intermediate ($\bar{\tilde{\rho}}^i$), and eroded ($\bar{\tilde{\rho}}^e$) design, with thresholds η, 0.5, and $1 - \eta$, respectively. The optimization problem is now reformulated as a *min-max* problem. The sensitivities can be obtained using Equation A.6. Analogous to [34] the volume constraint is imposed on the dilated design, and updated every 20 iterations.

[1] N. Aage, B. Lazarov, Parallel framework for topology optimization using the method of moving asymptotes, Structural and Multidisciplinary Optimization 47 (4) (2013) 493–505. doi:10.1007/s00158-012-0869-2.

[2] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, Computer Methods in Applied Mechanics and Engineering 197 (2008) 3768–3782. doi:10.1016/j.cma.2008.02.036.

[3] J. Parvizian, A. Düster, E. Rank, Finite cell method, Computational Mechanics 41 (1) (2007) 121–133. doi:10.1007/s00466-007-0173-y.

[4] D. Schillinger, M. Ruess, The finite cell method: A review in the context of higher-order structural analysis of cad and image-based geometric models, Archives of Computational Methods in Engineering 22 (3) (2015) 391–455. doi:10.1007/s11831-014-9115-y.

[5] M. Ruess, D. Tal, N. Trabelsi, Z. Yosibash, E. Rank, The finite cell method for bone simulations: verification and validation, Biomechanics and Modeling in Mechanobiology 11 (3) (2012) 425–437. doi:10.1007/s10237-011-0322-2.

[6] H. Wille, M. Ruess, E. Rank, Z. Yosibash, Uncertainty quantification for personalized analyses of human proximal femurs, Journal of Biomechanics 49 (4) (2016) 520–527. doi:10.1016/j.jbiomech.2015.11.013.

[7] Z. Yang, S. Kollmannsberger, M. Ruess, A. Düster, E. Rank, An efficient integration technique for the voxel-based finite cell method, International Journal for Numerical Methods in Engineering 91 (5) (2012) 457–471. doi:10.1002/nme.4269.

[8] J. Parvizian, A. D. A, E. Rank, Topology optimization using the finite cell method, Optimization and Engineering 13 (1) (2012) 57–78. doi:10.1007/s11081-011-9159-x.

[9] T. Nguyen, G. Paulino, J. Song, C. Le, A computational paradigm for multiresolution topology optimization (mtop), Structural and Multidisciplinary Optimization 41 (4) (2010) 525–539. doi:10.1007/s00158-009-0443-8.

[10] T. Nguyen, G. Paulino, J. Song, C. Le, Improving multiresolution topology optimization via multiple discretizations, International Journal for Numerical Methods in Engineering 92 (6) (2012) 507–530. doi:10.1002/nme.4344.

[11] N. Aage, M. Nobel-Jørgensen, C. Andreasen, O. Sigmund, Interactive topology optimization on hand-held devices, Structural and Multidisciplinary Optimization 47 (1) (2013) 1–6. doi:10.1007/s00158-012-0827-z.

[12] T. Nguyen, C. Le, J. Hajjar, High-order finite elements for topology optimization, in: 10th World congress on structural and multidisciplinary optimization. Orlando, 2013.

[13] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, E. Rank, Small and large deformation analysis with the p- and b-spline versions of the finite cell method, Computational Mechanics 50 (4) (2012) 445–478. doi:10.1007/s00466-012-0684-z.

[14] O. Sigmund, J. Petersson, Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima, Structural optimization 16 (1) (1998) 68–75. doi:10.1007/BF01214002.

[15] N. Zander, T. Bog, M. Elhaddad, R. Espinoza, H. Hu, A. Joly, C. Wu, P. Zerbe, A. Düster, S. Kollmannsberger, J. Parvizian, M. Ruess, D. Schillinger, E. Rank, Fcmlab: A finite cell research toolbox for {MATLAB}, Advances in Engineering Software 74 (2014) 49 – 63. doi:10.1016/j.advengsoft.2014.04.004.

[16] E. Andreassen, A. Clausen, M. Schevenels, B. Lazarov, O. Sigmund, Efficient topology optimization in matlab using 88 lines of code, Structural and Multidisciplinary Optimization 43 (1) (2011) 1–16. doi:10.1007/s00158-010-0594-7.

[17] K. Svanberg, The method of moving asymptotes: a new method for structural optimization, International Journal for Numerical Methods in Engineering 24 (2) (1987) 359–373. doi:10.1002/nme.1620240207.

[18] I. Babuška, B. Szabó, I. Katz, The p-version of the finite element method, SIAM Journal on Numerical Analysis 18 (3) (1981) 515–545. doi:10.1137/0718033.

[19] O. Sigmund, Morphology-based black and white filters for topology optimization, Structural and Multidisciplinary Optimization 33 (4) (2007) 401–424. doi:10.1007/s00158-006-0087-x.

[20] A. Düster, H. Bröker, E. Rank, The p-version of the finite element method for three-dimensional curved thin walled structures, International Journal for Numerical Methods in Engineering 52 (7) (2001) 673–703. doi:10.1002/nme.222.

[21] R. Guyan, Reduction of stiffness and mass matrices, AIAA Journal 3 (2) (1965) 380–380. doi:10.2514/3.2874.

[22] B. Lazarov, F. Wang, O. Sigmund, Length scale and manufacturability in density-based topology optimization, Archive of Applied Mechanics 86 (1) (2016) 189–218. doi:10.1007/s00419-015-1106-4.

[23] O. Sigmund, K. Maute, Topology optimization approaches, Structural and Multidisciplinary Optimization 48 (6) (2013) 1031–1055. doi:10.1007/s00158-013-0978-6.

[24] O. Sigmund, On the design of compliant mechanisms using topology optimization, Mechanics of Structures and Machines 25 (4) (1997) 493–524. doi:10.1080/08905459708945415.

[25] P. Christensen, A. Klarbring, An introduction to structural optimization, Vol. 153, Springer Science & Business Media, 2008.

[26] A. Díaz, O. Sigmund, Checkerboard patterns in layout optimization, Structural optimization 10 (1) (1995) 40–45. doi:10.1007/BF01743693.

[27] T. Davis, Direct methods for sparse linear systems, Vol. 2, Siam, 2006.

[28] O. Amir, N. Aage, B. Lazarov, On multigrid-cg for efficient topology optimization, Structural and Multidisciplinary Optimization 49 (5) (2014) 815–829. doi:10.1007/s00158-013-1015-5.

[29] B. Bourdin, Filters in topology optimization, International Journal for Numerical Methods in Engineering 50 (9) (2001) 2143–2158. doi:10.1002/nme.116.

[30] T. Bruns, D. Tortorelli, Topology optimization of non-linear elastic structures and compliant mechanisms, Computer Methods in Applied Mechanics and Engineering 190 (2001) 3443–3459. doi:10.1016/S0045-7825(00)00278-4.

[31] J. Guest, J. Prévost, T. Belytschko, Achieving minimum length scale in topology optimization using nodal design variables and projection functions, International Journal for Numerical Methods in Engineering 61 (2) (2004) 238–254. doi:10.1002/nme.1064.

[32] S. Xu, Y. Cai, G. Cheng, Volume preserving nonlinear density filter based on heaviside functions, Structural and Multidisciplinary Optimization 41 (4) (2010) 495–505. doi:10.1007/s00158-009-0452-7.

[33] O. Sigmund, Manufacturing tolerant topology optimization, Acta Mechanica Sinica 25 (2) (2009) 227–239. doi:10.1007/s10409-009-0240-z.

[34] F. Wang, B. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, Structural and Multidisciplinary Optimization 43 (6) (2011) 767–784. doi:10.1007/s00158-010-0602-y.