INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2011; **86**:765–781
Published online 28 December 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.3072

Filters in topology optimization based on Helmholtz-type differential equations

B. S. Lazarov*, † and O. Sigmund

Solid Mechanics, Department of Mechanical Engineering, Technical University of Denmark, Nils Koppels Alle, Building 404, DK-2800 Kgs. Lyngby, Denmark

SUMMARY

The aim of this paper is to apply a Helmholtz-type partial differential equation as an alternative to standard density filtering in topology optimization problems. Previously, this approach has been successfully applied as a sensitivity filter. The usual filtering techniques in topology optimization require information about the neighbor cells, which is difficult to obtain for fine meshes or complex domains and geometries. The complexity of the problem increases further in parallel computing, when the design domain is decomposed into multiple non-overlapping partitions. Obtaining information from the neighbor subdomains is an expensive operation. The proposed filter technique requires only mesh information necessary for the finite element discretization of the problem. The main idea is to define the filtered variable implicitly as a solution of a Helmholtz-type differential equation with homogeneous Neumann boundary conditions. The properties of the filter are demonstrated for various 2D and 3D topology optimization problems in linear elasticity, solved on serial and parallel computers. Copyright © 2010 John Wiley & Sons, Ltd.

Received 3 June 2010; Revised 10 September 2010; Accepted 21 September 2010

KEY WORDS: topology optimization; density filters; sensitivity filters; parallel computing

1. INTRODUCTION

Topology optimization as a design process has matured over the years. The technology has been utilized in many industrial designs. The aim of the optimization process is to obtain a material or design parameter distribution, which minimizes/maximizes the given objective subjected to prescribed constraints. In this paper the minimum compliance problem is considered and the so-called nested approach is utilized in the optimization process. The algorithm consists of three main phases—solving the actual physical problem by numerical discretization, calculating the gradients(sensitivities) of the objective with respect to the design variables, and updating them. The design variables are usually associated with each of the design cells and their value varies between 0 (empty cell) and 1 (solid cell). Although the concept is relatively simple, the solution of the optimization problem poses significant challenges. Direct application of the above three steps results in mesh-dependent designs.

Refining the mesh in topology optimization without any regularization results in a new design rather than in improving the design obtained with the coarse mesh. In the ideal case the mesh refinement would improve the finite element method (FEM) modeling of the same optimal design, and would produce a better description of the boundaries between the empty and the filled cells.

*Correspondence to: B. S. Lazarov, Solid Mechanics, Department of Mechanical Engineering, Technical University of Denmark, Nils Koppels Alle, Building 404, DK-2800 Kgs. Lyngby, Denmark.
†E-mail: bsl@mek.dtu.dk

In order to prevent such behavior, various modifications on the topology optimization algorithm have been proposed in the past. Techniques that have gained popularity are the mesh-independent sensitivity filtering [1, 2], the perimeter control method [3, 4], the density gradient control [5], and the mesh-independent density filtering techniques [6, 7]. Recently, projection schemes that build on density filtering [8, 9] and wavelet parameterization [10, 11] have also been proposed for preventing smeared designs and for imposing minimum length scale in the design.

In addition to the mesh dependency problem, when low order elements are used in the finite element model, the design may result in checkerboards. Checkerboards refer to designs with alternating solid and void cells ordered in checkerboard-like patterns. It has been shown in [12] and [13] that these solutions are physically not acceptable and they exist due to bad numerical modeling which overestimates the stiffness of the checkerboards. Such designs can be partially avoided by utilization of higher order elements or entirely by using filtering schemes.

In this paper the focus is on mesh-independent density filtering techniques. The filtered value of a design variable at a given cell is obtained by averaging over the neighbor cells. This operation can be generalized as a convolution integral [6]. Instead of explicit definition of the integral, the filtering in this work is obtained as a solution of a Helmholtz partial differential equation (PDE). Recently, the Helmholtz PDE has been applied as a sensitivity filter in topology optimization in [14]. Here the PDE is applied as a density filter, therefore it can be utilized in every optimization problem which needs to be regularized and does not have a defined length scale, i.e. discretization leads to mesh-dependent results. The Helmholtz PDE has also been applied for imposing length scale in damage mechanics [15] and in non-local plasticity [16].

We emphasize that we are not proposing a new filtering technique in this paper. Rather, we propose a computationally more efficient and tractable way of implementing the standard density filtering scheme [6, 7]. Our main aims are threefold: (i) avoid storing large arrays of neighborhood information, (ii) circumvent problems of the traditional density filter concerning smaller holes or slits in the design domain, and (iii) ensure efficient computations on large-scale problems and parallel computers. As is the case of the standard density filters the proposed PDE-based filter does not result in black and white designs. Crisp black and white designs can be obtained by adding a simple projection step to the filtered design as suggested in References [8, 9]. In the present paper we show one example of such a projection approach but otherwise emphasize that the computational efficiency of the black-and-white projection schemes is directly proportional to the efficiency of the underlying density filtering step. In the future work we may consider introducing design-dependent non-linear PDE filters in order to obtain black and white designs as an alternative to the standard projection schemes.

Realistic design problems in 3D result in large finite element models, which require utilization of multiple processors. The proposed filter can utilize the finite element solver used for solving the physical problem in the topology optimization process. It inherits the parallel properties of the code used for solving the physical problem and therefore its scalability is the same as the scalability of the FEM solver. Here, only the filtering is discussed with respect to parallel implementation and the problems associated with the parallelization of the whole optimization process will be the subject of subsequent articles.

2. TOPOLOGY OPTIMIZATION—LINEAR ELASTICITY

2.1. Problem formulation

The considered optimization problem is the maximization of the global stiffness (minimization of the compliance) of a mechanical system. The objective is to find a material distribution in the design domain Ω, which minimizes the compliance for a given set of loads and boundary conditions. The total volume occupied with material is constrained to be smaller than a prescribed value V_c. The material behavior is considered to be linear elastic.

The design domain is partitioned into cells in 2D and voxels in 3D, and the aim is to find whether cell i is filled with material ($\rho_i = 1$) or is void ($\rho_i = 0$). ρ_i denotes the design variable

DOI: 10.1002/nme

associated with cell i. In order to utilize gradient-based optimization techniques, the problem is relaxed by allowing intermediate values $0 \leqslant \rho_i \leqslant 1$ for the design parameter to exist. By using FEM discretization for the linear elasticity, the optimization problem can be written as follows [17, 18]:

$$\begin{aligned} \min: \quad & c(\boldsymbol{\rho}) = \mathbf{f}^{\mathsf{T}}\mathbf{u} \\ \text{s.t.}: \quad & \mathbf{K}(\boldsymbol{\rho})\mathbf{u} = \mathbf{f} \\ & V = \sum_{i \in \mathbb{N}_e} \rho_i v_i \leqslant V_c \\ & 0 \leqslant \rho_{\min} \leqslant \rho_i \leqslant 1 \quad \forall i \in \mathbb{N}_e \end{aligned} \tag{1}$$

where $\mathbf{u}$ is the FEM displacement vector, $\mathbf{f}$ is the vector with the applied nodal forces, $\mathbb{N}_e$ is a set of finite element indices, and v_i is the volume of cell i. The finite elements coincide with the design cells. $\mathbf{K}(\boldsymbol{\rho})$ is the standard FEM global stiffness matrix, e.g. [19]

$$\mathbf{K}(\boldsymbol{\rho}) = \sum_{i \in \mathbb{N}_e} \mathbf{K}_i(\rho_i) \tag{2}$$

$\sum$ is the finite element assembly operator and $\boldsymbol{\rho}$ is the vector with the design variables. The design parameters are bounded from below with $\rho_{\min}$, in order to prevent singularity of the stiffness matrix. Intermediate values for ρ_i, $i \in \mathbb{N}_e$ are undesirable in the final solution. In order to enforce values for the design parameter close to 0 or 1, the element stiffness matrix is modified as follows:

$$\mathbf{K}_i = \rho_i^p \mathbf{K}_{0,i} \tag{3}$$

where $\mathbf{K}_{0,i}$ is the stiffness matrix for an element filled with material and $p > 1$ is a penalization parameter. Intermediate densities become expensive by applying the modification given by (3). Usually p is set to be $p \geqslant 3$. This approach is known in the literature as SIMP (Solid Isotropic Material with Penalization) [17].

The optimization can be achieved by several different iterative procedures [17, 18]. Here, the design parameters are updated based on the Optimality Criteria (OC) scheme. The sensitivity of the objective function is always negative and can be written as

$$\frac{\partial c}{\partial \rho_i^e} = -p\rho_i^{p-1}\mathbf{u}^{\mathsf{T}}\mathbf{K}_{0,i}\mathbf{u}. \tag{4}$$

2.2. Density filtering

The optimal solution obtained by direct implementation of the steps discussed in the previous subsection is mesh dependent and suffers from the well-known checkerboard pathology for low order elements. Various modifications of the algorithm have been proposed in the past, in order to prevent these problems. Here, the focus is on mesh-independent density filtering techniques. The main idea is to modify the cell density, and hence the element stiffness, to become dependent on the density field in a given neighborhood. Density filtering has been introduced in [7] and the existence of the solution has been mathematically proven in [6].

The filtering is achieved by using a convolution product of a filter function and the density

$$\tilde{\rho}(\mathbf{x}) = (F * \rho)(\mathbf{x}) = \int_{\mathbb{B}_R} F(\mathbf{x} - \mathbf{y})\rho(\mathbf{y})\mathrm{d}\mathbf{y} \tag{5}$$

where $\mathbb{B}_R$ is a sphere in 3D and a circle in 2D with center $\mathbf{x}$ and radius R. The filter function $F(\cdot)$ is required to be non-negative in the support domain $\mathbb{B}_R$, and the following integral to be equal to 1:

$$\int_{\mathbb{B}_R} F(\mathbf{x})\mathrm{d}\mathbf{x} = 1. \tag{6}$$

As noted in [6] this definition of the filtering operator requires that the density field is extended over the whole space. By requiring that (6) is fulfilled, the volume of the material is the same

DOI: 10.1002/nme

for the filtered and the unfiltered density field. Therefore, the volume constraint can be imposed on the unfiltered field. Alternatives where (6) is not fulfilled can be implemented as well, however, the volume constraint in this case should be imposed on the filtered field. Usually, the integral (5) is replaced with the following expression:

$$\tilde{\rho}_j = \frac{\sum_{i \in \mathbb{N}_{e,j}} w(\mathbf{x}_i) v_i \rho_i}{\sum_{i \in \mathbb{N}_{e,j}} w(\mathbf{x}_i) v_i} \tag{7}$$

where the weighting function $w(\mathbf{x}_i)$ is taken to be linearly or exponentially decaying with increasing distance and $\mathbb{N}_{e,j}$ is the set of elements lying in the filter support domain for cell j. The element stiffness, when the density filter is applied in the optimization process, is a function of the filtered density

$$\mathbf{K}_i = \tilde{\rho}_i^p \mathbf{K}_{0,i} \tag{8}$$

and the sensitivities of the objective function are calculated by using the chain rule

$$\frac{\partial c(\tilde{\boldsymbol{\rho}}(\boldsymbol{\rho}))}{\partial \boldsymbol{\rho}} = \frac{\partial c}{\partial \tilde{\boldsymbol{\rho}}} \frac{\partial \tilde{\boldsymbol{\rho}}}{\partial \boldsymbol{\rho}}. \tag{9}$$

As mentioned in [9] when a density filter is applied, the original design variables $\rho_i, i \in \mathbb{N}_e$ do not have any physical meaning. Therefore, one should always present the filtered density field as a solution to the optimization problem.

2.3. Sensitivity filtering

Sensitivity filtering has been introduced in [2]. The idea is to base the design updates on filtered sensitivities, instead of the real sensitivities given by (4). Although the exact objective function is not precisely known, the technique has been successfully utilized in many optimization problems. These filters have shown very robust and reliable behavior. The sensitivity filter is given by

$$\widehat{\frac{\partial c}{\partial \rho_j}} = \frac{\sum_{i \in \mathbb{N}_{e,j}} w(\mathbf{x}_i) \rho_i \dfrac{\partial c}{\partial \rho_i}}{\rho_j \sum_{i \in \mathbb{N}_{e,j}} w(\mathbf{x}_i)} \tag{10}$$

where $w(\mathbf{x}_i)$ is a linearly or an exponentially decaying function. The updates of the design variables in the case of sensitivity filtering are based on the sensitivities calculated by (10).

2.4. Parallel implementation

The filtering defined by (5) or (10) introduces some implementation issues. In parallel computing the original design domain is partitioned into smaller subdomains. Each of them is assigned to a process which is assumed to operate autonomously. When necessary the processes exchange messages, in order to transfer information from one subdomain to another. The model is very general and is applicable on both distributed and shared memory machines. The FEM mesh consists of nodes and elements, and usually no information is available about the neighbor elements. Applying the standard filtering techniques for finding the neighbor cells requires a search for the neighbors to be performed. The problem is demonstrated on the left graph in Figure 1. If a cell is close to the border of the subdomain, information about the cells from the neighbor subdomain is necessary. Parallel search among the partitions is a very expensive operation and would require large amounts of communications, especially, if the filter radius becomes larger. Alternatively, the search procedure can be performed as a preprocessing step and the information about the neighbor cells can be stored and reused later. This approach increases significantly the memory utilization. For two-dimensional problems the number of the neighbor cells is proportional to $(R/a)^2$, and for three-dimensional problems it is proportional to $(R/a)^3$, where a is the length of the cell side. For 100 000 cells, filter radius from 2 to 10 times the cell size and x86-64 architecture, the memory

DOI: 10.1002/nme

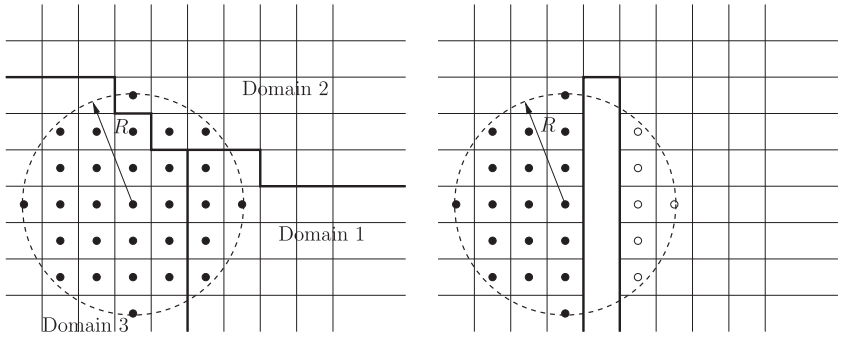

Figure 1. Density and sensitivity filtering—filter domain.

necessary to store the information is approximately equal to 12 MB, 43 MB, 102 MB, 200 MB, 345 MB, 548 MB, 818 MB, 1.1 GB, and 1.6 GB for $R/a = 2, 3, 4, 5, 6, 7, 8, 9, 10$, respectively. For unstructured tetrahedral meshes the memory requirements have to be multiplied by 6. Increasing the filter radius will also increase the amount of communications and the storage space necessary to keep locally the design variables from the neighbor domains contributing to the local filtered field.

The search procedure for the neighbor cells for non-convex domains requires information about the actual geometric boundaries and a very careful implementation of the actual code. The problem can be seen on the right-hand graph in Figure 1, where the intersection of the filter domain and the design domain results in two non-connected domains.

3. DENSITY FILTERING AS A SOLUTION OF A HELMHOLTZ-TYPE PDE

3.1. Isotropic filtering

Instead of using the explicit form of the convolution integral (5), the density filter can be defined implicitly as a solution of the Helmholtz PDE

$$-r^2\nabla^2\tilde{\rho}+\tilde{\rho}=\rho \tag{11}$$

with homogeneous Neumann boundary conditions

$$\frac{\partial\tilde{\rho}}{\partial\mathbf{n}}=0 \tag{12}$$

imposed on the boundary of the design domain. The solution of (11) can be expressed in the form of the averaging integral (5) with filter function replaced by Green's function [20] of the boundary value problem defined by (11) and (12). r in (11) is a length parameter which plays a similar role as R in the original filter. For $r \geqslant 0$ Green's functions are always positive, $G(\cdot) \geqslant 0$, and they decay monotonically with increasing distance $\|\mathbf{x}-\boldsymbol{\xi}\|$. The following properties hold

$$G(\mathbf{x},\boldsymbol{\xi}) \geqslant 0, \quad x \in \Omega_\xi, \quad \xi \in \Omega_\xi \tag{13}$$

$$\int_{\Omega_\xi} G(\mathbf{x},\boldsymbol{\xi})\,\mathrm{d}\xi = 1 \tag{14}$$

and therefore the filter defined by (11) and (12) preserves the volume, i.e. the material volume is the same before and after the filtering process. If the specified boundary conditions are different from (12), then the integral (14) differs from one and the solution of the PDE has to be normalized. Plots of the Green's functions in 1D are shown in Figure 2. They decay monotonically with increasing distance from the source point. As the filter length parameter approaches zero, $r \to 0$, the ratio $x/r \to \infty$, the Green's function approaches the Dirac's delta function and the filtered field approaches the unfiltered one. A plot of the filtered field for a 1D problem for different length

DOI: 10.1002/nme

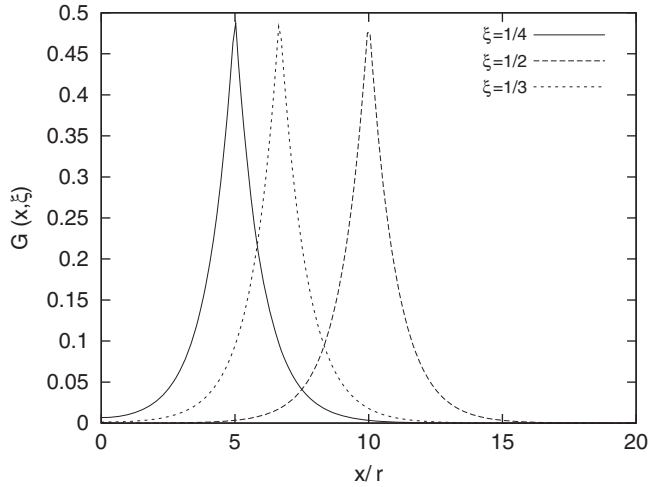

Figure 2. Profile of Green's function of the Helmholtz PDE for different values of the ξ.

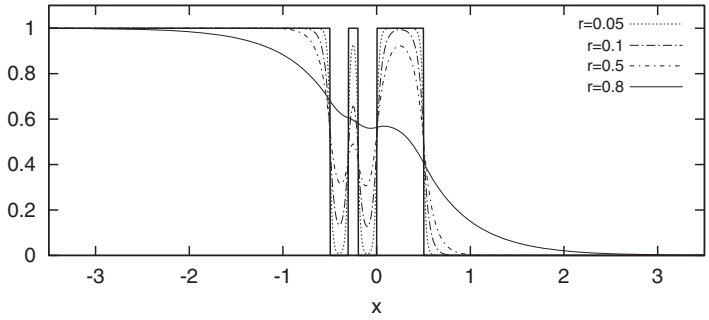

Figure 3. Solution of the Helmholtz equation for different values of the length parameter. The input to the filter is shown with a solid bold line.

parameters is shown in Figure 3. The discrete 0–1 input is shown with a solid bold line. A filter with small length parameter smooths out the sharp corners, and increasing the value of r results in complete removal of the single peak between -1 and 0.

3.2. Anisotropic filtering

The Helmholtz equation can also be written in the following form:

$$\nabla^{\mathsf{T}}\mathbf{K}_d\nabla\tilde{\rho}+\tilde{\rho}=\rho \tag{15}$$

where $\mathbf{K}_d$ is 3×3 positive-definite tensor in 3D and is 2×2 in 2D. The tensor can be written as

$$\mathbf{K}_d=\sum_{i=1}^{d} r_i^2\mathbf{v}_i\mathbf{v}_i^{\mathsf{T}} \tag{16}$$

where r_i is the length scale in the direction defined by the vector $\mathbf{v}_i$, d is the number of dimensions, and the vectors form orthogonal basis $\delta_{i,j}=\mathbf{v}_i^{\mathsf{T}}\mathbf{v}_i$. The original isotropic filter (11) is recovered by replacing the vectors $\mathbf{v}_i$ with the unit vectors $\mathbf{e}_i$ associated with the coordinate axes and by setting $r_1=r_2=r_3$. Different length parameters r_i introduce anisotropy in the filter and therefore anisotropy in the design.

It should be pointed out that both the isotropic (11) and anisotropic (15) PDE filters are equivalent to the classical [7] filter given by (5), with weighting functions in (5) replaced by the Green's functions of the PDEs.

DOI: 10.1002/nme

3.3. Numerical discretization

The design variables are constant within each cell/voxel. As the update in the optimization process is made on element level, the filtered field can be constant within the design cell. A candidate for solving (15) is the Finite Volume Method (FVM). However, usually the associated physical problem is solved by using the FEM and the FVM implementation would require a new solver implementation. For the considered optimization problems the FEM solver for the state problem can be re-utilized for solving the Helmholtz equation. The PDE filter inherits the parallel properties of the FEM implementation and does not require information about the neighbor cells. FEM implementation of the filter can re-utilize the mesh generated for solving the state problem.

The filtered density field within each element is approximated as

$$\tilde{\rho}_e = \mathbf{N}_e \tilde{\boldsymbol{\rho}}_e \tag{17}$$

where $\tilde{\boldsymbol{\rho}}_e$ is a vector with the field values associated with the element nodes, $\mathbf{N}_e$ is a vector with the standard finite element interpolation functions, and $\tilde{\rho}_e$ is the approximation of the filtered density field within the element. For zero length parameter the above discretization of the filtered field resembles the mesh-dependent filtering process in [21] for preventing checkerboards. The discretization of (15) is obtained by using the method of weighted residuals. Premultiplying the PDE with weighting functions equal to the nodal interpolation functions, integrating the resulting equation over the design domain and applying Green's formula for integration by parts result in a system of linear equations for the unknown values of the filtered field

$$\mathbf{K}_f \tilde{\boldsymbol{\rho}} = \mathbf{p}_f \tag{18}$$

where $\mathbf{K}_f = \sum_{i \in \mathbb{N}_e} \mathbf{K}_{f,i}$, $\mathbf{p}_f = \sum_{i \in \mathbb{N}_e} \mathbf{p}_{f,i}$, and $\sum$ is the standard finite element assembly operator. The element matrices and the load vectors are given as

$$\mathbf{K}_{f,i} = \int_{\Omega_i} [\nabla \mathbf{N}_e^{\mathsf{T}} \mathbf{K}_d \nabla \mathbf{N}_e + \mathbf{N}_e^{\mathsf{T}} \mathbf{N}_e] \mathrm{d}\Omega \tag{19}$$

$$\mathbf{p}_{f,i} = \int_{\Omega_i} \mathbf{N}_e^{\mathsf{T}} \rho \mathrm{d}\Omega = \rho_i \int_{\Omega_e} \mathbf{N}_e^{\mathsf{T}} \mathrm{d}\Omega. \tag{20}$$

Low order elements are the usual choice for modeling the actual physical problem in topology optimization. For triangular/tetrahedral first-order elements and for quad/cubic first-order elements the Neumann boundary condition (12) is automatically fulfilled and therefore there is no need for any special treatment around the boundaries of the design domain.

The solution of the linear system can be obtained by utilizing the solver for the associated physical problem. As the tangent matrix $\mathbf{K}_f$ is always positive definite, the Conjugate Gradient (CG) method is the most effective iterative algorithm for solving the system (18). For speeding up the convergence effective, preconditioners such as the Factorized Sparse Approximate Inverse [22] or block factorization can be utilized. As the preconditioner is not modified during the whole optimization process, it is strongly recommended to spend a longer time initially for obtaining an effective preconditioner, which will be later reused in all optimization steps. Alternatively, if the optimization problem is relatively small, the matrix $\mathbf{K}_f$ can be factorized and the factorization can be reused in following iterations. The experience of the authors shows that even a simple diagonal preconditioner results in very fast convergence. The size of the filter problem (18) is $\frac{1}{3}$ of the state problem in 3D, and therefore the filter cost is less than $\frac{1}{9} \approx 11.1\%$ of the physical problem. As the linear system (18) is much smaller than the one in (1) and the condition number of $\mathbf{K}_f$ does not depend on the design, the filter solution is in practice much cheaper than the stated percentage.

The sensitivities with respect to the filtered field can be calculated by using the adjoint method [17] and are given as

$$\frac{\partial c}{\partial \tilde{\boldsymbol{\rho}}} = \mathbf{s}_f = \sum_{i \in \mathbb{N}_e} \mathbf{s}_{f,i} \tag{21}$$

DOI: 10.1002/nme

$$\mathbf{s}_{f,i} = -p \int_{\Omega_i} \mathbf{N}_e \tilde{\rho}_i^{p-1} \mathbf{u}_e^{\mathsf{T}} \mathbf{B}^{\mathsf{T}} \mathbf{C}_0 \mathbf{B} \mathbf{u}_e \mathrm{d}\Omega \tag{22}$$

$$\mathbf{B} = \tilde{\nabla} \mathbf{N} \tag{23}$$

where $\tilde{\nabla}$ is the linear elasticity operator, $\mathbf{N}$ is a matrix with interpolation function for the linear elasticity problem, $\mathbf{s}_f$ is a vector with sensitivities with respect to the filtered density field, $\mathbf{s}_{f,e}$ is the contribution of each element to the global sensitivity vector, and $\mathbf{N}_e$ is vector with the interpolation functions for the PDE filter (17).

The linear system of equations (18) can also be written as

$$\mathbf{K}_f \tilde{\boldsymbol{\rho}} = \mathbf{T} \boldsymbol{\rho} \tag{24}$$

$$\mathbf{p}_f = \mathbf{T} \boldsymbol{\rho} \tag{25}$$

where $\mathbf{T}$ is a $m \times n$ matrix, m is the number of the degrees of freedom in the filtered field and n is the number of the cells. Row j of matrix $\mathbf{T}$ contains the values of the integral $\int_{\Omega_j} \mathbf{N}_e^{\mathsf{T}} \mathrm{d}\Omega$ at the degrees of freedom of element j. By using (24) the derivative of the filtered sensitives with respect to the design variable can be written as

$$\frac{\partial \tilde{\boldsymbol{\rho}}}{\partial \boldsymbol{\rho}} = \mathbf{T}^{\mathsf{T}} \mathbf{K}_h^{-1} \tag{26}$$

and the sensitivities of the objective function can be calculated as

$$\frac{\partial c}{\partial \boldsymbol{\rho}} = \mathbf{T}^{\mathsf{T}} \mathbf{K}_h^{-1} \mathbf{s}_f. \tag{27}$$

It should be noted that the actual inverse of the matrix $\mathbf{K}_h$ should never be computed explicitly. The sensitivities should be calculated by first solving $\mathbf{K}_h \mathbf{a}_f = \mathbf{s}_f$, and then applying the averaging operator $\mathbf{T}$ on $\mathbf{a}_f$. For triangular/tetrahedral first-order elements and for quad/hexahedral first-order elements the sensitivity $\frac{\partial c}{\partial \rho_i}$ will be the average of the values of $\mathbf{a}_f$ at the corresponding degrees of freedom for element i.

4. SENSITIVITY FILTERING AS A SOLUTION OF HELMHOLTZ PDE

The sensitivity filter (10) can also be represented as a convolution integral over the support domain of the filter function in a way similar to (5). Detailed description of the filter implementation as a solution of Helmholtz PDE and its behavior for various problems in 2D and 3D is given in [14]. Here, only some of the more important equations are presented and later in Section 5 their behavior is compared to the density filter.

The product of sensitivities and design variables at point $\mathbf{x} \in \Omega$ in the design domain is denoted as

$$f_p(\mathbf{x}) = \rho(\mathbf{x}) \left. \frac{\partial c}{\partial \rho} \right|_{\mathbf{x}} \tag{28}$$

By using the above equation, the implicit form of the convolution integral for sensitivity filtering can be defined as

$$\nabla^{\mathsf{T}} \mathbf{K}_d \nabla \widehat{f_p} + \widehat{f_p} = f_p \tag{29}$$

with homogeneous Neumann boundary condition as in (12). $\mathbf{K}_d$ is a positive-definite tensor and has the same representation as in (16), and $\widehat{f_p}$ is the filtered field. The isotropic version of this filter is obtained by replacing the tensor with a scalar proportional to the length parameter squared. The numerical implementation of the filter is the same as for the density filter. After obtaining the nodal values of the filtered field $\widehat{\mathbf{f}_p}$, the filtered sensitivity for a given element is obtained by averaging the nodal values of the filtered field and dividing by the element density.

DOI: 10.1002/nme

5. NUMERICAL EXAMPLES

The filter behavior is demonstrated on a few selected examples in 2D and 3D. Both the density and sensitivity filters are applied for optimizing the global stiffness of a 2D cantilever beam. The density filter is also demonstrated on a 3D cantilever beam with holes. More topology optimization results obtained with the sensitivity filter can be found in [14]. All of the presented results are obtained using parallel solvers and their behavior is discussed as well. The different meshes are obtained with GMSH [23] and the visualization is made by using ParaView. For all results the volume fraction is chosen to be 50%.

5.1. 2D Cantilever beam—Density filtering

The 2D cantilever beam example is shown in Figures 4 and 7. The design domain has length to height ratio 4:1. The displacements are fixed along the left edge and a point load 0.2 is applied in the middle of the right edge. Young's modulus is 1 and Poisson's ratio is $v=0.2$. The thickness is taken to be 0.01. The 2D examples, except the ones shown in Figures 5 and 6, are modeled by using 3D hexahedrons. The SIMP parameter is set to $p=3$. The discrete filtering problem is solved using the preconditioned conjugate gradient (PCG) method. The optimization process is terminated when the largest change in the design variables becomes smaller than 1%. Six different length parameters r are considered. By increasing the value of r small details in the design are removed by the filter. This behavior can be clearly seen by comparing the case for $r=0.02$ with $r=0.03$. For large length parameters the solution consists predominantly of intermediate densities.

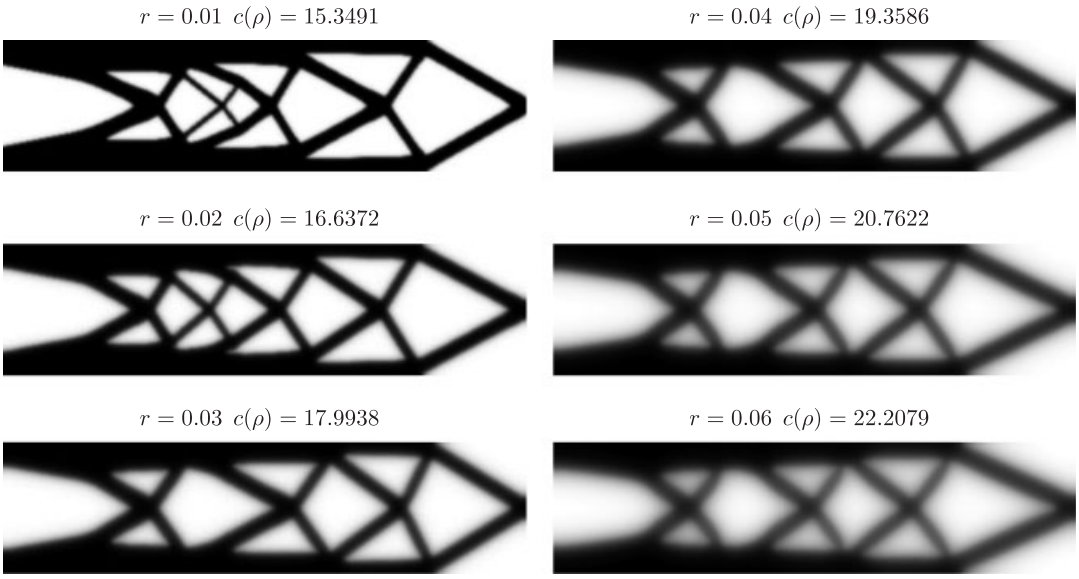

Figure 4. Helmholtz PDE density filtering—mesh 100×400 cells for cantilever beam with dimensions 1×4.

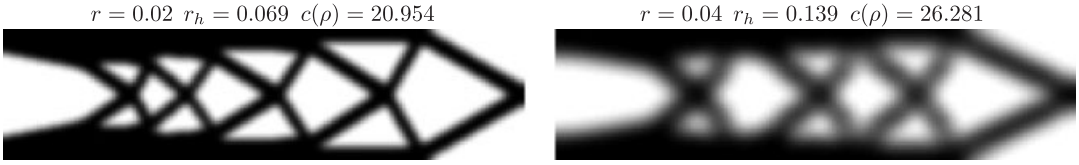

Figure 5. Classical density filter [7] with linearly decaying weighting function—mesh 50×200 cells for cantilever beam with dimensions 1×4.

DOI: 10.1002/nme

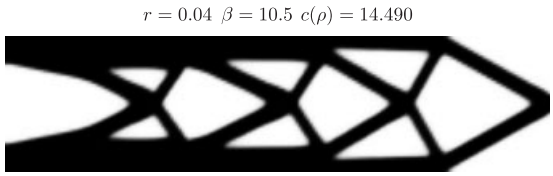

Figure 6. Black and white design by projection [8, 9] given by (35)—mesh 50×200 cells for cantilever beam with dimensions 1×4.

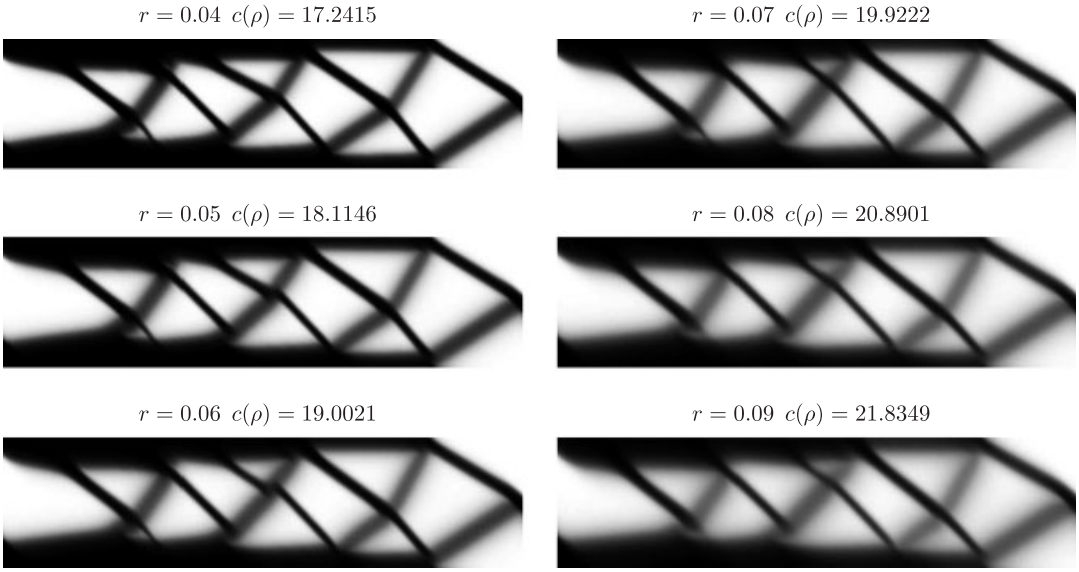

Figure 7. Helmholtz PDE density filtering—mesh 100×400 cells for cantilever beam with dimensions 1×4.

Comparison between filters with different weighting functions can be performed by defining equivalent length scales. A relation between the length parameters can be obtained by equating the second moments of the filter weighting functions. The normalized hat weighting function in 1D can be written as

$$w_h(x) = \frac{1}{r_h}\left(1 - \frac{|x|}{r_h}\right) \quad x \in [-r_h, r_h] \tag{30}$$

where r_h is the radius of the support domain. The second moment can be calculated as

$$I_h = \int_{-r_h}^{r_h} \frac{1}{r_h}\left(1 - \frac{|r|}{r_h}\right) r^2 \mathrm{d}r = \frac{1}{6} r_h^2 \tag{31}$$

The second moment of the Green's functions of the Helmholtz PDE is necessary to compare results obtained using (30). As the Green's functions for a finite domain depend on the position of the point, the equivalent length scale is obtained using the Green's functions of the PDE for an infinite domain. In 1D they are given as

$$w_g(x) = \frac{1}{2r} \exp\left(-\frac{|x|}{r}\right). \tag{32}$$

The second moment becomes

$$I_g = 2r^2. \tag{33}$$

DOI: 10.1002/nme

A relation between the length scales imposed by the weighting functions is obtained by equating (33) and (31)

$$r_h = 2\sqrt{3}r. \tag{34}$$

The results obtained using the classical density filtering technique [7] with length parameter obtained using the above relation are shown in Figure 5. The Green's functions compared to the linear hat function have higher weights for points close to the point where the filtering is performed and therefore the contrast in the optimal design obtained with Helmholtz filtering is expected to be higher for equivalent length scales. This is confirmed by comparing the results for $r = 0.02$ and $r = 0.04$ between Figures 4 and 5. Both weighting functions result in similar topologies and the design obtained by using the hat weighting function contains more gray compared to the Helmholtz filter.

High contrast in the design can be achieved by projection techniques as demonstrated in [8, 9]. The same approach can be applied to the Helmholtz filter and the result is shown in Figure 6. The projection is here simply chosen to have the following form:

$$\rho_h = 0.5 + \frac{\tanh(\beta(\tilde{\rho} - 0.5))}{2\tanh(0.5\beta)}. \tag{35}$$

The transformation is not volume preserving and hence the volume constraint has to be imposed on the filtered design. The derivatives of the objective and constraint with respect to the design variables can be obtained using the chain rule

$$\frac{\partial c(\tilde{\boldsymbol{\rho}}(\boldsymbol{\rho}))}{\partial \boldsymbol{\rho}} = \frac{\partial c}{\partial \boldsymbol{\rho}_h} \frac{\partial \boldsymbol{\rho}_h}{\partial \tilde{\boldsymbol{\rho}}} \frac{\partial \tilde{\boldsymbol{\rho}}}{\partial \boldsymbol{\rho}}. \tag{36}$$

As these techniques deserve more detailed analysis, a more systematic study is left for a subsequent article.

The anisotropic filter is demonstrated in Figure 7. The ratio of the long length scale to the short length scale in the anisotropic tensor (16) is 5:1, and the orientation is $\pi/4$ from the horizontal axis. Small details are preserved in the direction of the short length scale and removed in the longer one. The symmetry in the design is destroyed by the anisotropy of the filter.

The mesh independency obtained using the proposed filter is demonstrated in Figure 8. Three designs are obtained for three different mesh densities for filter length parameter $r = 0.02$. The designs are practically the same and the slight differences in the objective functions are due to the discrete representation of the continuous problem.

5.2. 2D Cantilever beam—Sensitivity filtering

Topology optimization results obtained by using Helmholtz equation as a sensitivity filter are shown in Figures 9 and 10. The optimization problem is the same as the one used for demonstrating the density filter. Six different length parameters are selected for the isotropic filter. Similar to the density filter larger length parameter filter small details in the design. For length parameter larger than $r = 0.03$ the gray areas dominate in the design, and the structural elements which can be clearly distinguished for small r are smeared. The anisotropic sensitivity filter is demonstrated in Figure 10. The ratio of the long to the short length parameter is 10:1, and the length parameter shown above the design is associated with the longer one. As for the density filter small details are removed along the direction associated with the long length parameter. The computational cost for the anisotropic and isotropic filters is the same. Implementing a similar anisotropic filter using the standard approach would require very expensive searches in elliptic regions.

5.3. 3D Cantilever

All 3D examples are computed with isotropic density filter. The first example is shown in Figures 11 and 12. The optimization is performed on a cantilever beam with fixed displacements on the right side and load uniformly distributed along the lower left edge. Three ellipsoidal holes are

DOI: 10.1002/nme

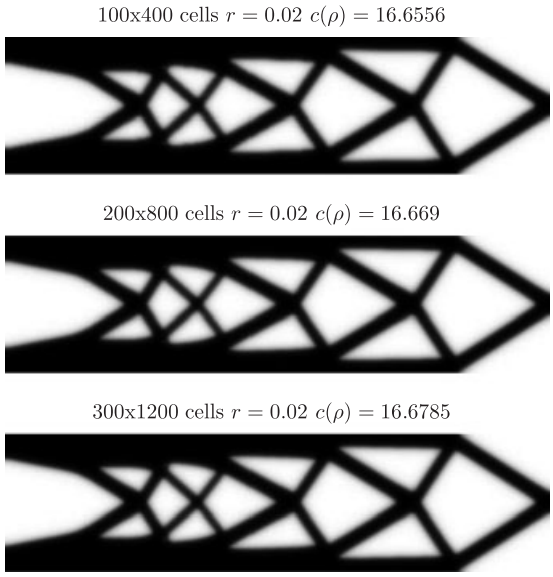

Figure 8. Helmholtz PDE density filtering—mesh independency.

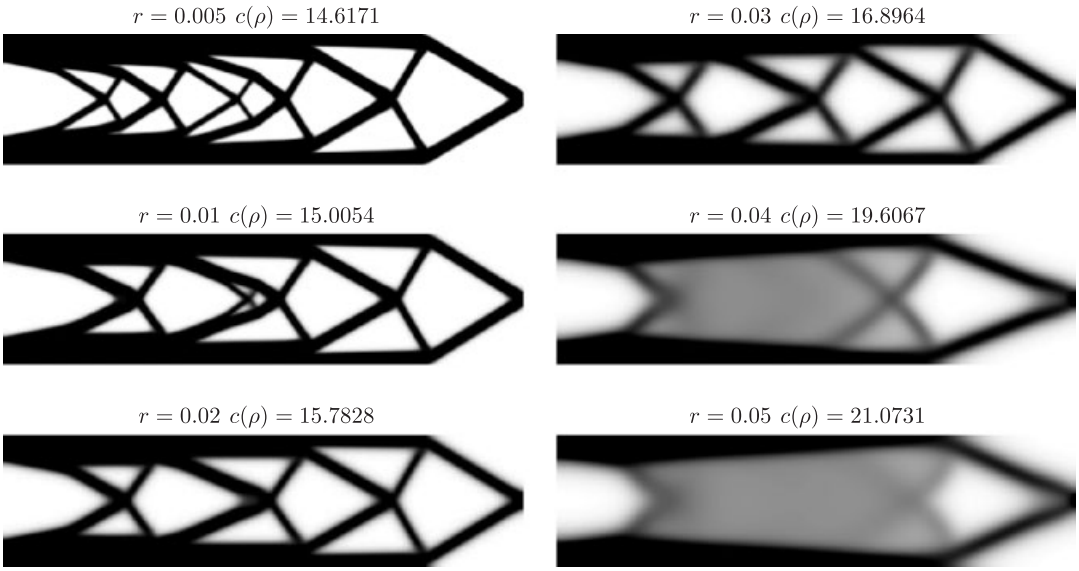

Figure 9. Helmholtz PDE isotropic sensitivity filtering—mesh 100×400 cells for cantilever beam with dimensions 1×4.

created in the original design domain. Two different filter length parameters are selected. The upper design in Figure 12 is obtained with $r = 0.04$, and the lower design is obtained with $r = 0.06$. A careful examination of the two designs shows that the thin walls are filtered away by increasing the filter parameter. The hole observed in the middle of the beam, shown with arrows, in the design with $r = 0.06$ is not observed in the upper one with length parameter $r = 0.04$.

The second example is again 3D cantilever beam (Figures 13 and 14) with the same load as the first and with a large number of small vertical holes. The size of these holes is comparable with the smaller element size allowed by the filter. The standard filtering technique in this case will introduce difficulties in interpreting the contribution of the neighbor cells to the filtered values. For all 3D examples a threshold of 66.67% gray is utilized for visualizing the results.

DOI: 10.1002/nme

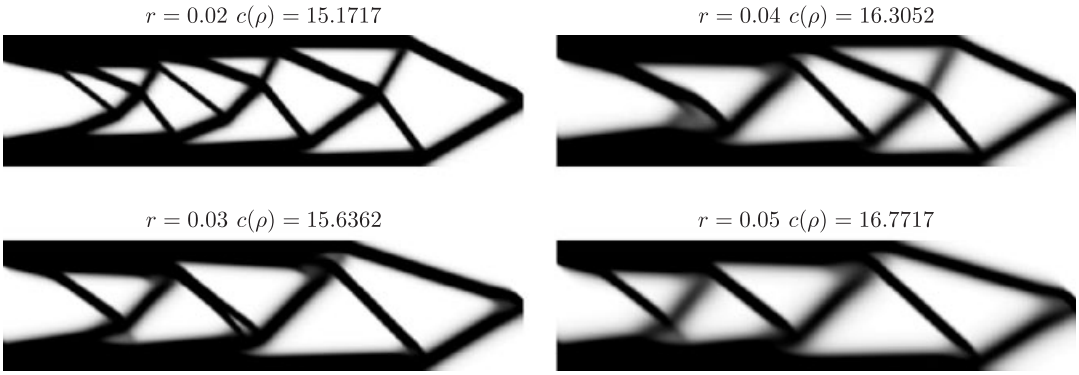

Figure 10. Helmholtz PDE anisotropic sensitivity filtering—mesh 100×400 cells for cantilever beam with dimensions 1×4.

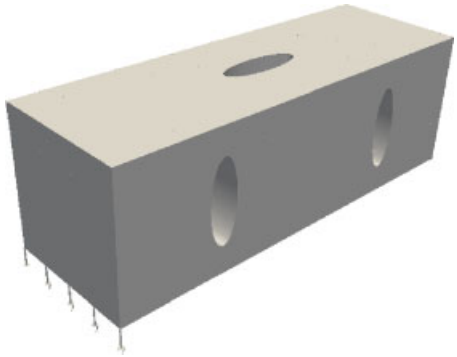

Figure 11. 3D design domain with elliptic holes and applied load.

Figure 12. Helmholtz PDE isotropic density filtering—1M cells for cantilever beam with dimensions $1 \times 1 \times 3$ with elliptic holes ($r = 0.04$ and $r = 0.06$). The arrows show the holes that appear in the design with $r = 0.06$.

5.4. Serial complexity of the classical vs PDE filters in 3D

The complexity of the classical filter is proportional to the volume of the support domain $\mathbb{B}_R$, and can for a regular mesh with cubic elements be evaluated to be approximately equal to $6\pi(r_h/a)^3/4$

DOI: 10.1002/nme

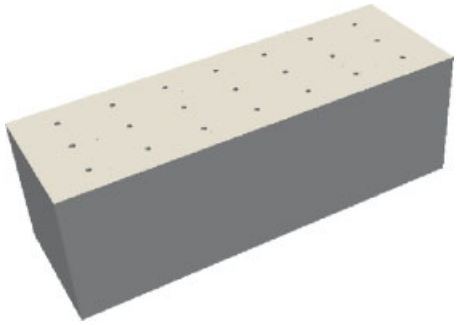

Figure 13. 3D design domain with small vertical holes.

Figure 14. Helmholtz PDE isotropic density filtering—1M cells for cantilever beam with dimensions $1 \times 1 \times 3$ with large number of small cylindrical holes.

FLOPs (floating point operations) per cell where a is the length of the cell edge. For tetrahedral meshes this estimation has to be multiplied by a factor of 6. The FLOPs per cell are estimated for the case when the weights in front of the cell design variables (7) are computed in advance and the information for the neighbors is stored in a pre-processing step.

The number of iterations for the PDE filter is proportional to a^{-1} when the solution is obtained by using a CG method and therefore depends linearly on the ratio r_h/a. Clearly for large ratios r_h/a the PDE filter will always be faster (in terms of FLOPs) than the classical one. For a small filter length comparison can be obtained by using numerical simulations. The cost of the PDE filter is equal to the cost of the CG method, which per one iteration is two inner products, three vector updates, and one matrix vector product.

Numerical simulations on a unit cube with a regular mesh and cell size $a = 0.02$, has been performed in order to compare the PDE filter with the theoretical estimates for the classical filter. The stopping criteria for the CG method is based on the relative residual tolerance to be smaller than 10^{-7}. The number of CG iterations for $r = 0.02, 0.03, 0.04$ are $14, 22, 28$, respectively. The resulting FLOPs per cell together with the perfect theoretical estimate for cubic and for tetrahedral elements are shown in Table I. The estimation for the PDE filter takes into account that the number of nodes in the FEM model is slightly larger than the number of cells. The most unfavorable implementation of the PDE filter is slower than the regular classic case with cubic elements, and faster when tetrahedral elements are utilized. It should be pointed out that the cost of building the search tree and performing the actual search as a pre-processing step is comparable to the cost of building a good preconditioner or setting up a multigrid framework for the PDE solution, which improves significantly the number of FLOPS per cell for the PDE filter. Furthermore, the search tree is not a natural data structure for finite element codes, however, effective preconditioners are necessary for effective solution of the associated state problem.

Comparison of the time necessary for applying the classical filter and the proposed Helmholtz filter is shown in Table II. Implementation of the classical filter with computing of the weights in advance, storing them inside each element and cycling through the elements on a Nehalem E5520 2.27 GHz processor is slower than the proposed PDE filter, although the perfect theoretical estimate shown in Table I gives better results for the FLOPs per cell. The main reason is that the evaluation of the filtered values result in unfavorable utilization of the memory hierarchy. When

DOI: 10.1002/nme

Table I. Floating point operations (FLOPS) necessary for computing filtered field value at a single point, using classical filter (DF) for hexahedral and tetrahedral design elements and for the Helmholtz filter (HF) for different filter parameters r.

r	DF Hex8 (FLOPS)	DF Tet4 (FLOPS)	HF (FLOPS)
0.02	193	1158	896
0.03	644	3864	1342
0.04	1548	9288	1708

Table II. Time necessary for computing filtered field using the Helmholtz filter (HF) and the classical filter with (DF Stor.) and without (DF Octree) storing information about neighbor elements for different filter parameters r.

r	DF Stor. (s)	DF Octree (s)	HF (s)
0.02	0.46	3.59	0.16
0.03	1.42	8.49	0.27
0.04	3.8	16.31	0.34

neighbor information is not computed in advance the classical filter is an order of magnitude slower than the proposed filter implementation. The neighbor search is performed by using Octree data structure which in the test case is very well balanced. For small filter radii the filter cost is negligible compared to the cost for obtaining the solution for the state problem. However, with increasing filter radius, the execution time for the classical filter grows significantly faster than the time necessary for the PDE filter. Therefore, for general applications the PDE filter should be the preferred choice.

5.5. Parallel scalability

All the presented examples are solved on a cluster consisting of computing nodes interconnected with 1 GB Ethernet. Each node consists of 2 quad core AMD2354 processors. The computational code is parallelized using OpenMPI. For a small number of processes, where the communications are kept within a single node, the Helmholtz sensitivity filter is 20–30% slower compared with the traditional implementation for small filter radius $r_h/a<2$. An increase of the filter radius in the traditional filter increases the computational and the communication time. In the proposed Helmholtz filter the solution time does not depend on the length parameter when appropriate preconditioners are employed. Experiments with block iterative solvers for solving (18) show significant decrease of the time spent in communications, as well as a decrease in the computational time due to the lower number of calls to the sparse matrix elements and better utilization of the memory hierarchy. In this case the Helmholtz filter is competitive with the traditional one even when small filter radius/length parameter is used. For larger radius/length parameter the Helmholtz filter would be the preferred choice, due to lower memory requirements and shorter computational and communication time (Table II).

The scalability of the optimization process when the Helmholtz filter is utilized depends on the scalability of the underlying FEM code. All the demonstrated results are computed with an in-house object-oriented C++ code based on some ideas from [24]. The key point of achieving good scalability for a large number of processes and slow communications is to overlap computations with communications. The scalability of the code on a Cray XT4/XT5 supercomputer is demonstrated in Figure 15. The Helmholtz PDE is solved using PCG and the linear elasticity problem is solved using preconditioned MinRES with FSAI preconditioner [25]. For up to 728 processes the code scales linearly for sufficiently large problems, and for larger number of processes the time spent in communications becomes significant which limits the further scalability. Increasing the problem size allows for utilization of larger number of processes with linear scalability.

DOI: 10.1002/nme

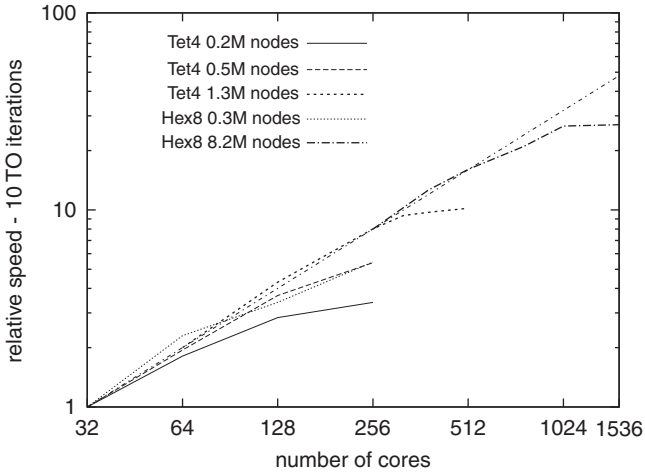

Figure 15. Parallel scalability of the topology optimization code on Cray XT4/XT5 supercomputer.

6. CONCLUSIONS

An alternative approach to density and sensitivity filtering based on the solution of the Helmholtz PDE is proposed for regularizing and introducing length scale in topology optimization problems. The examples clearly demonstrate and confirm the theoretical proof [6] that the obtained results are mesh independent. The numerical implementation avoids some of the bottlenecks observed in traditional filtering techniques. The computational complexity of the proposed filter is proportional to the filter radius which is significantly better than the classical one. The filter can be easily implemented in existing FEM codes and can utilize their solvers. The properties of the filter are controlled by a single length parameter for the isotropic case. Anisotropy can easily be introduced by adding a few more parameters for controlling the filter without increasing the computational cost. Further applications of the proposed filter in design sharpening, convergence speed-up and ensuring manufacturability of the obtained designs will be demonstrated in subsequent articles.

ACKNOWLEDGEMENTS

This work was supported by a Center of Advanced User Support (CAUS) and a hardware (HDW) grant from the Danish Center of Scientific Computing (DCSC), by an Elite Research Prize from the Danish Minister of Research and by a Eurohorcs/ESF European Young Investigator Award (EURYI, http://www.esf.org/euryi) through the grant Synthesis and topology optimization of optomechanical systems.

REFERENCES

1. Sigmund O. On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines* 1997; **25**:493–524.
2. Sigmund O, Petersson J. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization* 1998; **16**(1):68–75.
3. Haber RB, Jog CS, Bendsoe MP. A new approach to variable-topology shape design using a constraint on perimeter. *Structural Optimization* 1996; **11**(1):1–12.
4. Ambrosio L, Buttazzo G. An optimal-design problem with perimeter penalization. *Calculus of Variations and Partial Differential Equations* 1993; **1**(1):55–69.
5. Petersson J, Sigmund O. Slope constrained topology optimization. *International Journal for Numerical Methods in Engineering* 1998; **41**(8):1417–1434.
6. Bourdin B. Filters in topology optimization. *International Journal for Numerical Methods in Engineering* 2001; **50**:2143–2158.
7. Bruns TE, Tortorelli DA. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3443–3459.

DOI: 10.1002/nme

8. Guest JK, Prevost JH, Belytschko T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering* 2004; **61**(2):238–254.
9. Sigmund O. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization* 2007; **33**:401–424.
10. Kim YY, Yoon GH. Multi-resolution multi-scale topology optimization a new paradigm. *International Journal of Solids and Structures* 2000; **37**:5529–5559.
11. Poulsen TA. Topology optimization in wavelet space. *International Journal for Numerical Methods in Engineering* 2002; **53**:567–582.
12. Diaz A, Sigmund O. Checkerboard patterns in layout optimization. *Structural and Multidisciplinary Optimization* 1995; **10**:40–45.
13. Jog CS, Haber RB. Stability of finite element models for distributed-parameter optimization and topology design. *Computer Methods in Applied Mechanics and Engineering* 1996; **130**:203–226.
14. Lazarov BS, Sigmund O. Sensitivity filters in topology optimisation as a solution to Helmholtz type differential equation. *Proceedings of the Eighth World Congress on Structural and Multidisciplinary Optimization*, Lisbon, Portugal, 2009.
15. Peerlings RHJ, De Borst R, Brekelmans WAM, De Vree JHP. Gradient enhanced damage for quasi-brittle materials. *International Journal for Numerical Methods in Engineering* 1996; **39**:3391–3403.
16. Engelen RAB, Geers MGD, Baaijens FPT. Nonlocal implicit gradient-enhanced elasto-plasticity for the modelling of softening behaviour. *International Journal of Plasticity* 2003; **19**:403–433.
17. Bendsoe MP, Sigmund O. *Topology Optimization—Theory*, *Methods and Applications*. Springer: Berlin, Heidelberg, 2003.
18. Sigmund O. A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization* 2001; **21**:120–127.
19. Wiberg NE, Samuelson A. *Finite Element Method Basics*. Studentlitteratur: Lund, Sweden, 1998.
20. Zauderer E. *Partial Differential Equations of Applied Mathematics*. Wiley: New York, U.S.A., 1998.
21. Hammer VB. Checkmate? Nodal densities in topology optimization. *Proceedings of the Second Max Planck Workshop on Engineering Design Optimization*, DTU, Denmark, 2001.
22. Yu KL, Yu YA. Factorized sparse approximate inverse preconditionings I. theory. *SIAM Journal on Matrix Analysis and Applications* 1993; **14**:45–58.
23. Geuzaine C, Remacle JF. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* 2009; **79**(11):1309–1331.
24. Lazarov BS, Augarde CE. Object oriented design of parallel and sequential finite element codes. *Proceedings of the 13th ACME Conference*, Sheffield, U.K., 2005.
25. Lazarov BS, Sigmund O. Factorized parallel preconditioner for the saddle point problem. *International Journal for Numerical Methods in Biomedical Engineering* 2009; DOI: 10.1002/cnm.1366.

DOI: 10.1002/nme