Struct Multidisc Optim (2009) 39:217–225
DOI 10.1007/s00158-008-0337-1

BRIEF NOTE

A simple heuristic for gray-scale suppression in optimality criterion-based topology optimization

Albert A. Groenwold · L. F. P. Etman

Received: 14 March 2007 / Accepted: 3 November 2008 / Published online: 8 January 2009

Abstract We propose a very simple heuristic to suppress gray-scale material in topology optimization in optimality criterion-like implementations. Gray-scale suppression (GSS) is recommended for use in conjunction with the SIMP material description, although gray-scale suppression in itself is adequate to effect predominantly black-and-white designs. Minimal changes are required to incorporate the gray-scale suppression technique in Sigmund's popular 99-line Matlab code for topology optimization.

Keywords Topology optimization · SIMP · Optimality criterion (OC) · Gray-scale suppression

1 Introduction

In topology optimization, we seek to introduce topological features into a structure, such that the distribution of material is optimal in some sense, subject to any number of linear and/or nonlinear inequality constraints. In general, one of these constraints limits the amount of material that may be present in the final solution; this constraint is often referred to as the 'volume constraint'.

A number of methods have been developed to find approximate solutions for the difficult discrete topology optimization problem, which is intractable. Arguably, the simplest and most popular of these is the so-called solid isotropic material with penalization (SIMP) method. Independently proposed by Bendsøe (1989) and Rozvany and Zhou (1991), the SIMP method relies on a penalty parameter in a relaxed continuous setting to effect solid-void or black-and-white designs through penalization of intermediate densities.

The SIMP method is associated with two major difficulties, namely non-existence of a solution (the so-called mesh-dependence problem), and the presence of intermediate density or gray-scale material. We will herein restrict ourselves to the latter, which has received relatively little attention in the literature. The importance of being able to generate truly discrete or black-and-white solutions is emphasized by, for example, Sigmund (2007).

Methods previously proposed to generate predominantly black-and-white designs include 'pure' post-processing methods like contour plotting, which can in general not satisfy the constraints. Established discrete optimization methods like branch-and-bound (B&B) cannot be used, due to the very high dimensionality of the optimization problem. Even neighborhood search (NHS) methods are for this reason effectively disqualified. Methods based on rounding are computationally efficient, but are often not guaranteed to satisfy the constraints, while they may also prove to be very ineffective if very high fractions of gray material are present before

Based on the paper 'Predominantly black-and-white topology optimization via gray-scale filtering', *Proc. Seventh World Congress on Structural and Multidisciplinary Optimization*, Seoul, Korea, May 2007.

A. A. Groenwold (✉)
Department of Mechanical Engineering,
University of Stellenbosch,
Matieland 7602, South Africa
e-mail: albertg@sun.ac.za

L. F. P. Etman
Department of Mechanical Engineering,
Eindhoven University of Technology,
Eindhoven, The Netherlands

rounding is initiated. Other methods proposed include volumetric penalization by Zhou and Rozvany (1991) and Bruns (2005), a nonlinear diffusion technique by Wang et al. (2004), hierarchical constraining of the slope of density by Zhou et al. (2001), a discrete dual method by Beckers (1996), a hierarchical neighborhood search method by Svanberg and Werme (2005), and a family of morphology-based filters by Sigmund (2007).

In this paper, we propose a very simple heuristic for gray-scale suppression, which merely requires the modification of the OC statement. The method requires little post-processing, the solution is predominantly black-and-white, and the volume constraint is adhered to. Not-withstanding its simplicity, the method seems to result in good predominantly black-and-white solutions, in particular when used in combination with a continuation strategy on the gray-scale suppression parameter.

The development of our paper is as follows: In Section 2, we summarize the topology optimization problem. In Section 3, we present the optimality criterion (OC) method, whereafter we immediately introduce our new heuristic OC statement with gray-scale suppression in Section 4. We then present numerical results in Section 5, and offer concluding remarks in Section 6.

Finally, in the Appendix, we list the required changes to incorporate our gray-scale suppression mechanism into the popular 99-line Matlab implementation by Sigmund (2001) of the OC-like method for the topology optimization problem.

2 The classical topology optimization problem

For linear elastostatic structures, the 'classical' topology optimization problem may be expressed as

$$\begin{aligned} &\min_{\boldsymbol{x}} \; f_0(\boldsymbol{x}) \\ &\text{subject to } f_1(\boldsymbol{x}) \leq 0, \\ &\qquad \boldsymbol{K}\boldsymbol{q} = \boldsymbol{r}, \\ &\qquad x_i \in [0,1], \qquad i = 1, 2, \cdots, n. \end{aligned} \tag{1}$$

We assume that the design domain is discretized using the finite element method (FEM). $f_0(\boldsymbol{x})$ represents the to-be-specified objective function, with the corresponding vector of elemental densities $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$ (the design variables). f_1 represent a to-be-specified prescribed inequality constraint function (usually a limit on the available material resource). $\boldsymbol{q}(\boldsymbol{x})$ and $\boldsymbol{r}$ respectively represent the finite element displacement and force vectors, while $\boldsymbol{K}(\boldsymbol{x})$ represents the global assembled finite element stiffness matrix (it is assumed that the loads $\boldsymbol{r}$ are design independent).

Since the discrete programming problem is intractable, problem (1) is often replaced by a relaxed problem in which the discrete variables $x_i \in [0,1]$ are replaced by $\boldsymbol{0} < \boldsymbol{x}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{1}$; it is now implicitly assumed that the resulting problem is combined with some (heuristic) method to arrive at an (approximate) discrete solution. We opt for the so-called solid isotropic material with penalization (SIMP) method.

We restrict ourselves to linear elastic materials. Hence, the constitutive relationship is assumed to be

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\sigma}$, $\boldsymbol{C}$ and $\boldsymbol{\epsilon}$ respectively are the stress, elasticity and strain tensors. For element i, the material density is introduced via the elasticity tensor $\boldsymbol{C}_i$, using

$$\bar{\boldsymbol{C}}_i(x_i) = \mu_{1_i}(x_i)\boldsymbol{C}_i. \tag{3}$$

Here, $\boldsymbol{C}_i$ is the elasticity tensor of the solid material, while $\bar{\boldsymbol{C}}_i(x_i)$ is the effective elasticity tensor. After Bruns (2005), we have now introduced the notion of the first density measure $\mu_{1_i}(x_i)$ for element i; the density 'scales' the material properties between 0 or void, and 1 or solid. We implicitly assume that $\mu_{1_i}(x_i)$ depends on element i only. In the standard SIMP method, we have

$$\mu_{1_i}(x_i) = x_i^p, \tag{4}$$

with the penalty $p \geq 1$. In the case of the inequality, p is the penalty parameter that drives the solution towards the bounds $\boldsymbol{x}_{\min}$ and $\boldsymbol{1}$, e.g. see Bendsøe (1995). $\boldsymbol{x}_{\min} > 0$ is introduced for the sake of numerical stability (it prevents disjointed regions and ensures that the stiffness matrix does not become singular). For the sake of simplicity, we will set $x_{i_{\min}} = \rho_{\min} \; \forall \, i$.

The principle of stationary potential energy may be used to demonstrate that the finite element stiffness matrices are expressed as

$$\boldsymbol{K}_i = \int_{v_i} \boldsymbol{B}_i^T \bar{\boldsymbol{C}}_i(x_i) \boldsymbol{B}_i \, dv, \tag{5}$$

where the $\boldsymbol{B}_i$ represent the elemental strain-displacement operators. Momentarily only considering minimum compliance problems, the compliance $f_0(\boldsymbol{x})$ is obtained as

$$f_0(\boldsymbol{x}) = \boldsymbol{q}^T\boldsymbol{r} = \boldsymbol{q}^T\boldsymbol{K}\boldsymbol{q} = \sum_{i=1}^{n} \mu_{1_i}(x_i)\boldsymbol{q}_i^T\boldsymbol{K}_i\boldsymbol{q}_i \tag{6}$$

where subscript i indicates elemental quantities and operators; there are n finite elements in the mesh.

For an elemental volume of v_i, the effective elemental material volume can be represented as

$$v_i^e = v_i \mu_{2_i}(x_i), \tag{7}$$

with $\mu_{2_i}(x_i)$ the second density measure. We then formulate the volume constraint

$$f_1(\boldsymbol{x}) = \frac{v(\boldsymbol{x})}{v_0} - \bar{v} = \frac{1}{v_0} \sum_{i=1}^{n} v_i \mu_{2_i}(x_i) - \bar{v} \leq 0, \tag{8}$$

where $v(\boldsymbol{x})$ represents the material or final structural volume, v_0 the total volume of the design domain Ω, and $0 < \bar{v} < 1$ a prescribed limit on the final volume fraction allowed. In the classical SIMP method, we have $d = 1$; in volumetric penalization methods, $0 < d \leq 1$.

3 The OC statement in topology optimization

Bendsøe and Sigmund (2003) have previously popularized the following heuristic updating scheme[1] for the design variables if a single volume constraint is present:

$$x_i^{\text{new}} = \begin{cases} x_i B_i^{\eta} & \text{if } \check{x}_i < x_i B_i^{\eta} < \hat{x}_i, \\ \check{x}_i & \text{if } x_i B_i^{\eta} \leq \check{x}_i, \\ \hat{x}_i & \text{if } x_i B_i^{\eta} \geq \hat{x}_i, \end{cases} \tag{9}$$

for $i = 1, 2, \cdots, n$, and

$$\check{x}_i \leftarrow \max\left(x_i - \delta, \rho_{\min}\right),$$

$$\hat{x}_i \leftarrow \min\left(x_i + \delta, 1\right), \tag{10}$$

with $\delta > 0$ a prescribed move limit, and $x_{i_{\min}} = \rho_{\min}\ \forall\, i$. Then, (9) is iteratively solved until some stopping condition is satisfied, or until a prescribed number of iterations have passed. In the 'fixed-point' updating scheme (9), the B_i are found from the optimality criterion (the stationary conditions) as (Bendsøe 1995)

$$B_i = -\left(\lambda \frac{\partial f_1}{\partial x_i}\right)^{-1} \left(\frac{\partial f_0}{\partial x_i}\right), \tag{11}$$

with $\lambda > 0$. In (9), we have $B_i \leftarrow B_i^{\eta}$, with η a heuristic numerical 'damping coefficient', largely introduced to overcome the oscillatory behavior observed when (9) is iteratively solved with $\eta = 1$. Seemingly, the choice $\eta = 1/2$ is invariably made in minimum compliance design.[2]

[1] The basic form of the criterion can be traced back to the early work by Olhoff et al. (1981) and Cheng and Olhoff (1982), while its use has become widely known through the work by Bendsøe and Kikuchi (1988).

[2] For an interpretation of η in a rigorous sequential approximate optimization (SAO) framework, the reader is referred to Groenwold and Etman (2008).

In iteratively applying (9), the Lagrangian multiplier λ may be found using a simple bi-sectioning strategy.

If the loads $\boldsymbol{r}$ are assumed to be independent of the design variables, the derivatives of compliance are obtained as

$$\frac{\partial f_0}{\partial x_i} = -p(x_i)^{p-1} \boldsymbol{q}_i^T \boldsymbol{K}_i \boldsymbol{q}_i. \tag{12}$$

In the SIMP method, we obtain for the constraint f_1

$$\frac{\partial f_1}{\partial x_i} = \frac{\partial v}{\partial x_i} = \frac{v_i}{v_0}. \tag{13}$$

4 A simple heuristic for gray-scale suppression in SIMP-based OC-like methods

We now propose a simple heuristic to effect black-and-white designs, using a standard SIMP-like approach to define the first and second density measures, and OC-like method (9): we modify updating rule (9) to become

$$x_i^{\text{new}} = \begin{cases} \Gamma(x_i B_i^{\eta}) & \text{if } \check{x}_i < \Gamma(x_i B_i^{\eta}) < \hat{x}_i, \\ \check{x}_i & \text{if } \Gamma(x_i B_i^{\eta}) \leq \check{x}_i, \\ \hat{x}_i & \text{if } \Gamma(x_i B_i^{\eta}) \geq \hat{x}_i, \end{cases} \tag{14}$$

where the operator $\Gamma(\cdot)$ biases individual gray-scale variables towards white or void designs during 'inner iterations' (i.e. when the Lagrangian multiplier λ is updated). The linear volume constraint is 'satisfied exactly' in each iteration (viz. the equality $f_1 = 0$ holds); black material is effected when $\Gamma(x_i B_i^{\eta}) \geq \hat{x}_i$. Hence, we propose to *underestimate the contribution of all gray-scale material, while preserving the volume constraint.*

In this note, we restrict ourselves to the two simple, typical gray-scale suppression operators depicted in Figs. 1 and 2. Figure 1 depicts the effects of the 'power-law' gray-scale suppression operator

$$\Gamma(x_i B_i^{\eta}) = (x_i B_i^{\eta})^q, \tag{15}$$

with $q \geq 1$, while Fig. 2 depicts the effects of the 'linear' gray-scale suppression operator

$$\Gamma(x_i B_i^{\eta}) = q(x_i B_i^{\eta}) - q + 1, \tag{16}$$

again with $q \geq 1$. The specific forms (15) and (16) were in part chosen to ensure that a value of unity for q results in a standard OC method. Using power-law gray-scale suppression (15) implies that updating rule (9) is replaced by

$$x_i^{\text{new}} = \begin{cases} (x_i B_i^{\eta})^q & \text{if } \check{x}_i < (x_i B_i^{\eta})^q < \hat{x}_i, \\ \check{x}_i & \text{if } (x_i B_i^{\eta})^q \leq \check{x}_i, \\ \hat{x}_i & \text{if } (x_i B_i^{\eta})^q \geq \hat{x}_i, \end{cases} \tag{17}$$

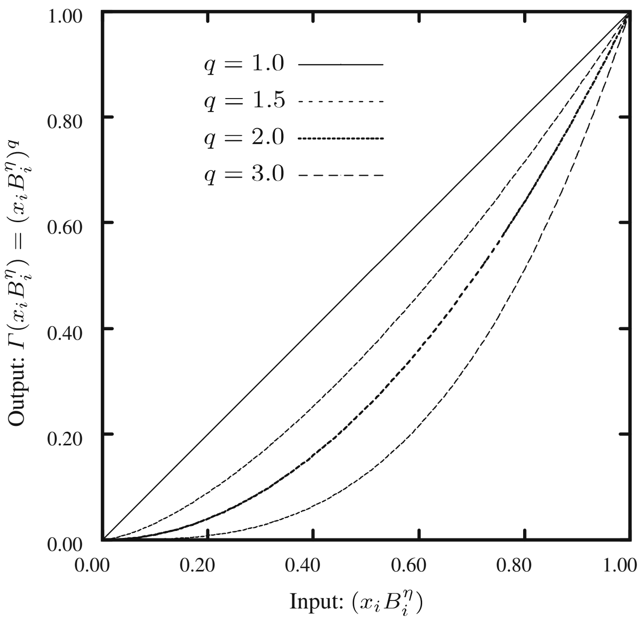


Fig. 1 Typical power-law gray-scale suppression

whereas the linear gray-scale suppression operator (16) results in

$$x_i^{\text{new}} = \begin{cases} q(x_i B_i^\eta) - q + 1 & \text{if } \check{x}_i < q(x_i B_i^\eta) - q + 1 < \hat{x}_i, \\ \check{x}_i & \text{if } q(x_i B_i^\eta) - q + 1 \leq \check{x}_i, \\ \hat{x}_i & \text{if } q(x_i B_i^\eta) - q + 1 \geq \hat{x}_i. \end{cases} \quad (18)$$

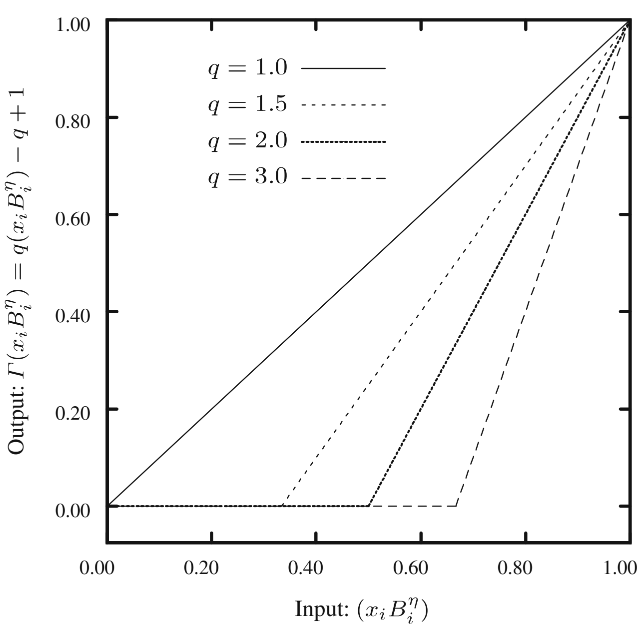


Fig. 2 Typical linear gray-scale suppression

Typical values for both the SIMP penalty parameter p and the gray-scale suppression parameter q are $p = 3$ and $q = 2$, although it is by far preferable to use continuation on both p and q. In addition, for $q > 1$, it may be sensible to use slightly lower values for p.

5 Numerical results

In presenting numerical results, we will use the linear mesh independence filter proposed by Sigmund (1994, 1997) to filter the sensitivities of the objective function. We will use both isoparametric 4-node Lagrangian finite elements and 8-node serendipity finite elements, respectively denoted Q4 and Q8.

We introduce $\phi_{B\&W}$, the elemental 'black-and-white fraction' as

$$\phi_{B\&W} = (n_{[0]} + n_{[1]})/n, \quad (19)$$

where $n_{[0]}$ and $n_{[1]}$ respectively indicate the number of elements on the lower and upper bounds. We will denote our method by OC with gray-scale suppression, or OC-GSS for short.

We will occasionally use the following simple continuation strategy: we keep p and q at unity until $k = 20$ iterations; thereafter, we set

$p^{\{k+1\}} \leftarrow (1 + \alpha_p) p^{\{k\}}$, and/or

$q^{\{k+1\}} \leftarrow (1 + \alpha_q) q^{\{k\}}$,

with $\alpha_p, \alpha_q \geq 0$. We use[3] $\alpha_p = 0.02$ and $\alpha_q = 0.01$. In addition, we enforce $p = \min(p, 3.0)$. The aforementioned continuation strategies are implicated when we use the notations $p > 1$ respectively $q > 1$. Unless otherwise stated, implementations without GSS (i.e. classical SIMP) are terminated after 100 iterations; implementations with GSS on the other hand are terminated when $\|\boldsymbol{x}^{\text{new}} - \boldsymbol{x}\|_\infty < 10^{-6}$ (and a tighter tolerance in the inner loop of the optimality criterion update). Finally: unless otherwise stated, we use power-law based GSS.

5.1 The MBB beam

Consider the popular MBB beam depicted in Fig. 3. We use mesh independence filtering at 8% of the height. We start with a very simple implementation of gray-scale suppression, by keeping both the SIMP parameter $p = 3$ and the GSS parameter q constant, and we

[3]We have not done exhaustive experimentation to determine an optimal continuation strategy.

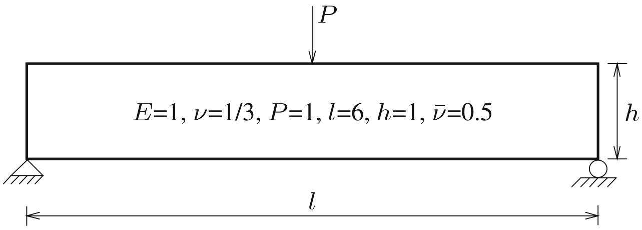
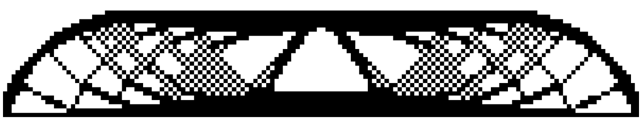


Fig. 3 The MBB beam (unit thickness; plane stress)

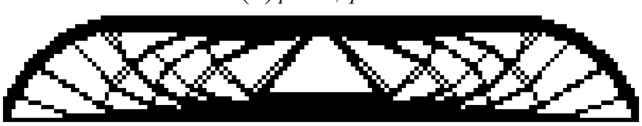
(a) $q = 1$ (i.e. classical SIMP): $f_0^* = 208.66, \phi^*_{B\&W} = 0.628$

(b) $q = 2$: $f_0^* = 193.01, \phi^*_{B\&W} = 0.975$

Fig. 4 MBB beam, Q4 elements, 150 × 50 mesh, $p = 3$, terminated after 100 iterations, power-law OC-GSS algorithm (**a**, **b**)

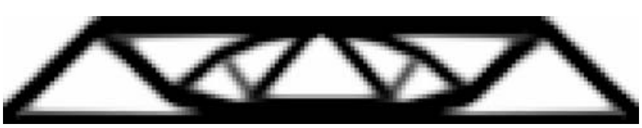


(a) $p > 1, q = 1$

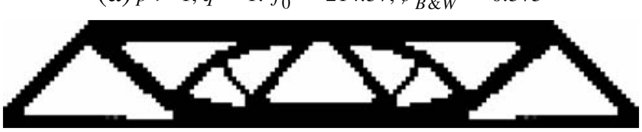


(b) $p = 1, q > 1$

Fig. 5 MBB beam, Q4 elements, 75 × 25 mesh, no mesh independence filtering whatsoever, power-law OC-GSS algorithm, terminated after 100 iterations (due to the checkerboarding, we do not report function values) (**a**, **b**)

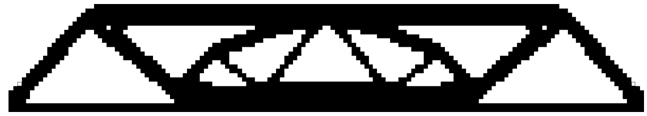


(a) $p > 1, q = 1$: $f_0^* = 214.57, \phi^*_{B\&W} = 0.575$



(b) $p = 1, q > 1$: $f_0^* = 195.90, \phi^*_{B\&W} = 1.000$

Fig. 6 MBB beam, Q8 elements, 75 × 25 mesh, power-law OC-GSS algorithm (**a**, **b**)



(a) 75 × 25 mesh: $f_0^* = 186.83, \phi^*_{B\&W} = 0.999$

(b) 600 × 200 mesh: $f_0^* = 191.18, \phi^*_{B\&W} = 1.000$

Fig. 7 MBB beam, Q4 elements, $p = 1$, $q > 1$, power-law OC-GSS algorithm (**a**, **b**)

terminate after 100 iterations. The result is depicted in Fig. 4, which illustrates how easily high black-and-white fractions $\phi^*_{B\&W}$ may be effected. The (few) changes required to the popular 99-line Matlab topology optimization code of Sigmund (2001) to generate the results depicted in Fig. 4 are presented in the Appendix.

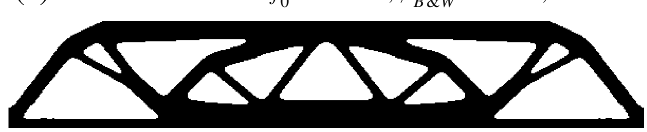
In Fig. 5, we show that the extent of checkerboarding with Q4 elements may greatly be reduced if GSS is performed without SIMP-like penalization. (Both figures were obtained after 100 iterations.) Figure 6 depicts the effects of GSS for Q8 elements, while Fig. 7 illustrates that GSS can result in mesh-independent designs. Finally, in Fig. 8, we compare linear GSS with power-law GSS. A short comment on this is in order: our experiments suggest that both power-law and linear GSS work well. However, linear GSS is more ‘aggressive’ than power-law GSS, and it is in general a good idea to reduce the value of the continuation parameter α_q when linear GSS is performed.

5.2 The loaded knee structure

Next, we study the loaded knee structure depicted in Fig. 9. The data are from the paper by Svanberg and Werme (2005).

In Fig. 10, we again demonstrate that the results are mesh independent. Not shown is that the results for

(a) Power-law OC-GSS: $f_0^* = 188.49, \phi^*_{B\&W} = 1.000, k^* = 129$



(b) Linear OC-GSS: $f_0^* = 188.65, \phi^*_{B\&W} = 1.000, k^* = 108$

Fig. 8 MBB beam, Q4 elements, 300 × 100 mesh, $p > 1$, $q > 1$ (**a**, **b**)

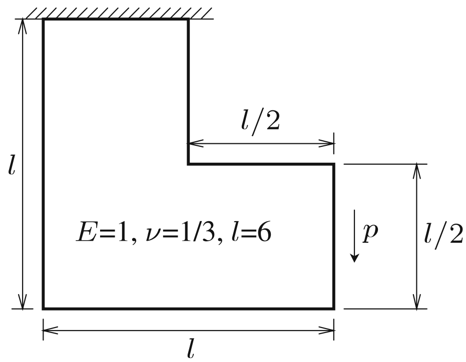


Fig. 9 The loaded knee structure (unit thickness; plane stress)

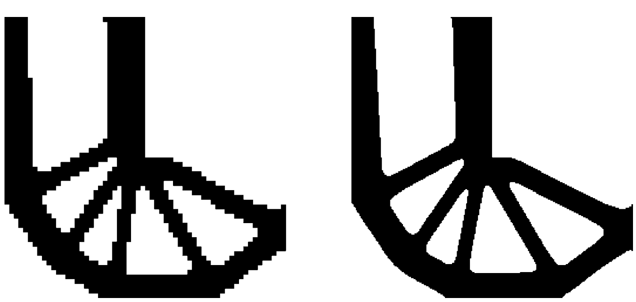
$p = 2$ are (marginally) superior to the results obtained with $p = 3$.

In Fig. 11, we compare our result with the pure black-and-white design published by Svanberg and Werme (2005), who used a hierarchical neighborhood search method. Note that the method of Svanberg and Werme does not satisfy an *a priori* specified maximum volume fraction $\bar{v}$. We have therefore used a prescribed volume fraction of $\bar{v} = 3064/6912$, being equal to the final volume fraction obtained by Svanberg and Werme (2005). It does not seem possible to use more refined meshes with the NHS method of Svanberg and Werme.

5.3 Compliant mechanism design

We now present a single result to illustrate that gray-scale suppression may also be applied to non-monotonic objective functions. These functions for example occur in compliant mechanism design problems, which are non-self-adjoint.

We consider the popular force inverter compliant mechanism design problem depicted in Fig. 12. The problem and its formulation are described in detail in



(a) 60×60 mesh, $r_{\text{mesh}} = 2$ ($\phi^*_{B\&W} = 1$) (b) 240×240 mesh, $r_{\text{mesh}} = 8$ ($\phi^*_{B\&W} = 1$)

Fig. 10 The loaded knee structure, power-law gray-scale suppression, $\bar{v} = 13/27$, $p = 2$, $\rho_{\min} = 10^{-9}$, $q > 1$

(a) Svanberg: $f^* = 78.563$ (b) This study: $f^* = 77.629$

Fig. 11 Comparison with the result obtained by Svanberg and Werme for the loaded knee structure using a 96×96 mesh, $p = 2$, $q > 1$, power-law gray-scale suppression (**a**, **b**)

the text by Bendsøe and Sigmund (2003) (see Section 5.1.5 in this text, which lists a 104-line Matlab program for optimizing the force inverter).

The results are generated for an 80×40 mesh (due to symmetry, this is for half the force inverter). The filter radius in Sigmund's linear mesh independence filter is fixed at 1% of the dimensions of the square design domain.

We use settings in the OC-method popularized by Bendsøe and Sigmund. That is, for the move limit we use $\delta = 0.1$, and for the 'numerical damping parameter' η, we use $\eta = 0.3$.

As is usual in the OC-method, we use the inconsistent modification

$$\frac{\partial f_0^{\{k\}}}{\partial x_i} = \min\left(-\epsilon_c, \frac{\partial f_0^{\{k\}}}{\partial x_i}\right); \ \epsilon_c > 0, \qquad (20)$$

proposed by Bendsøe and Sigmund (2003), to 'monotonize' the objective f_0. Herein, we will use $\epsilon_c = 1 \times 10^{-10}$.

Numerical results are presented in Fig. 13. The iterations are terminated when the 2-norm $\|\boldsymbol{x}^{\{k+1\}} -$

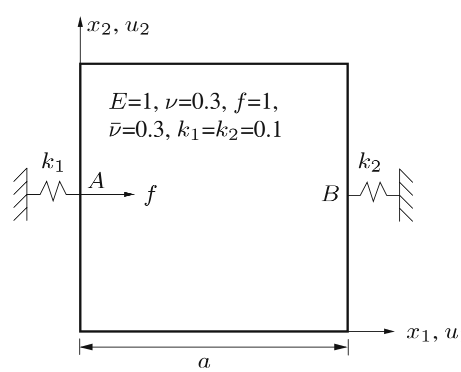
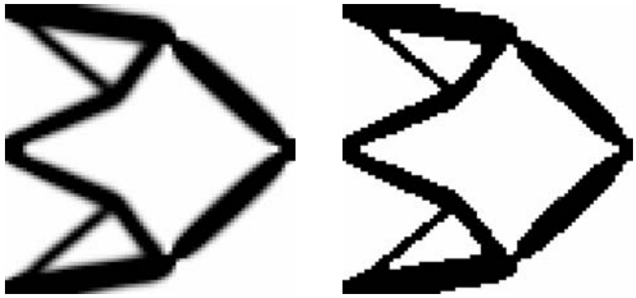


Fig. 12 The force inverter



(a) $q = 1$: $f_0^* = -1.044$, $\phi^*_{B\&W} = 0.711$, $k^* = 147$ (b) $q > 1$: $f_0^* = -1.122$, $\phi^*_{B\&W} = 1.000$, $k^* = 105$

Fig. 13 Optimal topologies found for the force inverter, 80×40 mesh, power-law OC-GSS algorithm (**a**, **b**)

$\boldsymbol{x}^{\{k}\| < \epsilon_x^2$, with $\epsilon_x = 1 \times 10^{-2}$. (Tighter tolerances have little influence on algorithm OC-GSS, since the solutions are predominantly black and white to start with, but greatly increase the computational effort when gray-scale suppression is not performed.) Finally, we enforce $p = \min(p, 2)$ if gray-scale suppression is performed, and $p = \min(p, 3)$ if gray-scale suppression is not performed.

Figure 13 reveals that gray-scale suppression increases the magnitude of the objective f_0, while simultaneously increasing the black-and-white fraction $\phi_{B\&W}$. However, gray-scale suppression even reduces the computational effort, as indicated by the required number of iterations k^*. The reduction of computational effort becomes even more pronounced if the stopping tolerance is tightened.

5.4 Heat conduction

Finally, we implement the 91 line MATLAB code for heat conduction problems in the book by Bendsøe and Sigmund (2003) (see Fig. 5.6 therein, pg. 271).

Table 1 Optimal results for the heat conduction problem, 80×80 mesh

Method	f_0^*	f_1^*	$\phi^*_{B\&W}$
SIMP	5159.56	3.954e-11	0.216
SIMP-R	7535.77	2.878e-04	1.000
OC-GSS	6503.80	−4.219e-11	1.000

This time, we compare our OC-GSS method with an OC-like rounding method, proposed by one of the reviewers, for which the MATLAB implementation is as follows:

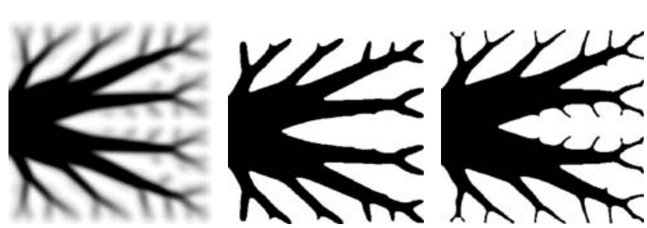
```
function [xnew]=ROUNDX(nelx,nely,
                       x,volfrac)
xlow = 0; xupp = 1;
while (xupp-xlow > 1e-6)
  xmid = 0.5*(xupp+xlow);
  xnew=max(0.001,ceil(x-xmid));
  if sum(sum(xnew)) - volfrac*nelx
                      *nely > 0;
    xlow = xmid;
  else
    xupp = xmid;
  end
end
```

For an 80×80 mesh, results are presented in Fig. 14 and Table 1. The SIMP implementation is terminated after 100 iterations, whereafter rounding commences. The GSS solution is clearly superior to the rounded solution, indicated by 'SIMP-R'.

Next, we repeat the comparison using a refined 320×320 mesh, see Fig. 15. This time, we terminate all the algorithms once $\|\boldsymbol{x}^{\text{new}} - \boldsymbol{x}\|_\infty < 10^{-6}$ is achieved. The implications for computational effort are clear from Table 2, while the GSS solution is still superior.

(a) $q = 1$ (SIMP) (b) SIMP-R (c) $q > 1$ (OC-GSS)

Fig. 14 Optimal topologies found for the heat conduction problem, 80×80 mesh, $p = 3$, mesh independence filter radius = 2.4 (**a**–**c**)



(a) $q = 1$ (SIMP) (b) SIMP-R (c) $q > 1$ (OC-GSS)

Fig. 15 Optimal topologies found for the heat conduction problem, 320×320 mesh), $p = 3$, mesh independence filter radius = 9.6 (**a**–**c**)

Table 2 Optimal results for the heat conduction problem, 320×320 mesh

Method	f_0^*	f_1^*	$\phi^*_{B\&W}$	Iterations
SIMP	1274592.14	2.67e-12	0.215	432
SIMP-R	1868104.83	−4.86e-06	1.000	432(+1)
OC-GSS	1639486.77	8.75e-14	1.000	149

6 Conclusions

We have proposed a very simple method for generating predominantly or pure black-and-white optimal topologies in OC-like methods, denoted gray-scale suppression (GSS). Gray-scale suppression may be used in conjunction with the SIMP material description, although gray-scale suppression in itself seems to be adequate to effect predominantly black-and-white designs. It is nevertheless suggested that GSS is used in combination with SIMP, albeit possibly with lower values of the SIMP penalty parameter p.

Salient features of the GSS method are simplicity, ease of implementation, computational efficiency, and suitability for problems of very high dimensionality.

Appendix: Matlab implementation

In this section, we list the required changes when incorporating power-law gray-scale suppression (15) into the popular 99-line Matlab topology optimization code of Sigmund (2001).[4]

The required changes are minimal. In fact, only 5 lines need to be changed; of these, the first is descriptive only, while the next 3 changes are merely required to pass the parameter q from the command line to the subroutine that implements the heuristic 'optimality criterion' (we have now also renamed subroutine OC to subroutine OC_GSS). The changes are listed as follows:

```
1 % 99 LINE TOP OPT CODE BY SIGMUND
                  WITH POWER-LAW
                  GRAY-SCALE SUPPRESSION
2  function topBW(nelx,nely,volfrac,
                    penal,q,rmin);
28 [x] = OC_GSS(nelx,nely,x,
                  volfrac,dc,q);
38 function [xnew] = OC_GSS(nelx,nely,x,
                            volfrac,dc,q)
```

```
42 xnew = max(0.001,max(x-move,min(1.,
     min(x+move,(x.*sqrt(-dc./lmid))
     .^q))));
```

Note the two extra parentheses; the complete expression `x.*sqrt(-dc./lmid)` is raised to the power q. In addition, we make an *optional* change to also output the black-and-white volume fraction to the output device, by inserting the following line between the original lines 30 and 31:

```
30a    bw=(sum(sum(lt(x,0.001+1e-6)))+
     sum(sum(gt(x,1.0-1e-6))))/
         (nelx*nely);
```

This optional change require that we also modify the *original* line 33:

```
33      ' ch.: ' sprintf('%9.6f',change)
             ' B&W: ' sprintf('%6.3f',
                                 bw) ])
```

Finally, we prefer to change the termination criteria in line 8 (also optional):

```
8  while change > 1e-4 & loop < 100
```

A.1 Running the code

– To execute a standard SIMP method (with $p = 3.0$), the code is called with

  `topBW(150,50,0.5,3.0,1.0,4.0)`

  The resulting optimal topology is depicted in Fig. 4a. Note that we have generated our results using $\nu = 1/3$ instead of $\nu = 0.3$, and also with a tighter inner loop tolerance in the optimality criterion update.

– To activate gray-scale suppression (with $q = 2.0$), while retaining SIMP-like penalization (with $p = 3.0$), the code is called with

  `topBW(150,50,0.5,3.0,2.0,4.0)`

  This time, the resulting optimal topology is depicted in Fig. 4b.

A.2 Continuation

It is strongly recommended to use continuation on both p (`penal`) and q. For q, this may be effected by (for example) replacing line 9 by

```
9 loop = loop+1; if loop < 21 q=1;
                         else q=q*1.01;
                         end
```

[4]Our changes refer to the version published in Sigmund (2001). Note that the 99-line Matlab code currently downloadable from http://www.topopt.dtu.dk/ has one extra comment line in the header.

For p, we suggest to use a continuation factor larger than 1.01, e.g. 1.02. (For p it is worthwhile to enforce an upper limit, e.g. see Section 5).

References

Beckers M (1996) Topology optimization using a dual method with discrete variables. Struct Optim 11:102–112

Bendsøe M (1989) Optimal shape design as a material distribution problem. Struct Optim 1:193–202

Bendsøe M (1995) Optimization of structural topology, shape and material. Springer, Berlin

Bendsøe M, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71(2):197–224

Bendsøe M, Sigmund O (2003) Topology optimization: theory, methods and applications. Springer, Berlin

Bruns T (2005) A reevaluation of the SIMP method with filtering and an alternative formulation for solid-void topology optimization. Struct Multidisc Optim 30:428–436

Cheng KT, Olhoff N (1982) Regularized formulation for optimal design of axisymmetric plates. Int J Solids Struct 18: 153–169

Groenwold A, Etman L (2008) On the equivalence of optimality criterion methods and sequential approximate optimization in the classical topology layout problem. Int J Numer Methods Eng 73:297–316

Olhoff N, Lurie K, Cherkaev A, Fedorov A (1981) Sliding regimes and anisotropy in optimal design of vibrating axisymmetric plates. Int J Solids Struct 17:931–948

Rozvany G, Zhou M (1991) Applications of COC method in layout optimization. In: Eschenauer H, Mattheck C, Olhoff N (eds) Proc. engineering optimization in design processes. Springer, Berlin, pp 59–70

Sigmund O (1994) Design of material structures using topology optimization. PhD thesis, Technical University of Denmark, Department of Solid Mechanics

Sigmund O (1997) On the design of compliant mechanisms using topology optimization. Mech Struct Machines 25:495–526

Sigmund O (2001) A 99 line topology optimization code written in Matlab. Struct Multidisc Optim 21:120–127

Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidisc Optim 33:401–424

Svanberg K, Werme M (2005) A hierarchical neighborhood search method for topology optimization. Struct Multidisc Optim 29:325–340

Wang M, Zhou S, Ding H (2004) Nonlinear diffusions in topology optimization. Struct Multidisc Optim 28:262–276

Zhou M, Rozvany G (1991) The COC method. Part II. Topological, geometrical and generalized shape optimization. Comput Methods Appl Mech Eng 40:1–26

Zhou M, Shyy Y, Thomas H (2001) Checkerboard and minimum member size control in topology optimization. Struct Multidisc Optim 21:152–158