A PRELIMINARY FORMALIZATION OF SCALE IN TOPOLOGY OPTIMIZATION

BY

LOUIS VAUGHT

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Civil Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Professor Ahmed Elbanna
ABSTRACT

Topology Optimization often enforces a minimum feature size, also known as a scale, to avoid instability and inaccuracy in the optimized result. This is usually done by applying a discretized convolution kernel on either the density or the sensitivity values. It is not well-established, however, exactly how this filtering approach enforces scale numerically, nor is it evident that the enforced scale is consistent.

This paper introduces Linear Scale-Space Theory from the field of computer vision as a rigorous explanation for how a generic convolution enforces scale upon an optimization algorithm. This framework gives several guiding axioms for how to construct a filtering procedure in order to generate a consistent scale parameter across the entire problem domain.

When this theory is reapplied to linear filtering in Topology Optimization, it is shown that the standard filtering approach results in inconsistent enforcement of scale on the boundaries of a volume. This formulation error must be corrected by expansion of the boundaries to include virtual elements which have a finite sensitivity but can accept no volume, also known as passive elements. Additionally, it is shown that irregular meshes result in inconsistent scale across the volume, which should be corrected by local modification of the convolution kernel.

Numerical tests are performed for basic problems with expanded boundaries in both 2D and 3D. Significant changes in solution topology for these basic problems are noted and discussed. From this preliminary set of results, further avenues of research are suggested.
ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Dr. Ahmed Elbanna, for his support as I explored this field of research. The work has been highly speculative and somewhat unprecedented, and at times progress was slow and difficult. Without his patience, encouragement, and insight, I would never have been able to develop this theory to the point it is at now.

I would also like to thank the CUBIT and PLATO development teams at Sandia National Laboratories for introducing me to topology optimization from the perspective of a designer, and for giving me the tools to explore general behaviors of the method. Without that experience, I likely wouldn’t have noticed the high-level issues with mesh sensitivity that lead directly to this work.

Thank you Brett Sneed and Darin Peetz; both of you have taken time out of your schedules to correspond with me at length about this issue. Your help and criticism have been absolutely invaluable.

Finally, my thanks to the Department of Civil and Environmental Engineering at the University of Illinois at Urbana-Champaign, which has supported this work. Thanks as well to the National Science Foundation, which supported this work via grant NSF-CMMI 1435920.
# TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

CHAPTER 2: TOPOLOGY OPTIMIZATION

CHAPTER 3: SCALE-SPACE THEORY

CHAPTER 4: EXTENSION OF LINEAR SCALE-SPACE TO FILTERING

CHAPTER 5: CONSEQUENCES OF SCALE-SPACE ENFORCEMENT

CHAPTER 6: CONVOLUTION FILTERING MODIFICATION

CHAPTER 7: CONCLUSIONS

CHAPTER 8: FUTURE WORK

REFERENCES

APPENDIX A: MODIFIED TOP88 MATLAB CODE

APPENDIX B: MODIFIED TOP3D MATLAB CODE
Chapter 1: Introduction

Topology Optimization is the process of optimizing a distribution of material within a domain to minimize some physical objective function. These optimizations often rely on discretization of the underlying physics using the Finite Element Method. Since Finite Element discretization creates inaccuracy and instability at the element-level, it is advantageous to ensure that individual elements are smaller than the physical objects they model. Topology Optimization normally modifies density distributions on the element level; this ensures element-level topology for a given problem, no matter the size of the discretization.

Thus, to ensure accuracy and stability of the optimization, there exists a need to uniformly impose a general feature size, also known as a scale, onto a given problem. Although there are many approaches to this, the most common is to use a discretized convolution kernel on either the density or the sensitivity values (Sigmund, 2007). The application of this filtering method is not without controversy, however, as it is sometimes regarded as heuristic (Sigmund & Petersson, 1998) or unnecessary (Talischi et al., 2009).

From the perspective of an engineer using Topology Optimization to perform design work, the need to formalize the motivation behind filtering is particularly stark. With a filter applied, optimization results will often be globally sensitive to local irregularities in mesh topology, such as local asymmetries in an otherwise symmetric domain. Furthermore, solutions will often exhibit attachment to the boundaries of the optimization domain, and this attachment can often be prevented by redefining the domain such that the boundary is no longer close to the optimized solution.

Shown in Figure 1 is the effect of adding volume to a Michell Truss optimization problem without adding boundary conditions, modifying the starting volume distribution, or changing the absolute volume constraint. The optimized solution retreats from the removed boundary without meaningfully entering.
the added volume. This result was produced using the topology optimization package in the Albany Component-Based PDE Code (Salinger, 2013).

![Figure 1. 3D Michell Truss Optimization with Standard Problem Setup (a) and Unused Additional Volume (b)](image)

This behavior is clearly unwanted; both of these solutions exist within the same geometric boundaries, and both are candidate solutions to the optimization. Ideally, if a similar minimum exists in both problems and the starting conditions are identical, then the optimization method will always converge to that solution (Martínez, 2005). In addition, the method should be stable enough that if one of these solutions is not an artifact of errors in the problem setup, then it will always be chosen.

Visual inspection of the two solutions, however, shows that the topology in the smaller volume which intersects the boundary is particularly thin. This suggests that the filter is not properly enforcing scale on the boundaries. Since there is no detailed explanation of why a particular filtering operation creates a particular scale, there are no tools to evaluate how this potential error is generated.

The goal of this paper is therefore to put forward a consistent mathematical explanation of how a filtering operation enforces scale. From this explanation, it will be shown how enforcement of length-scale results in sensitivity of the converged topology to both optimization boundary location and local irregularities in the problem discretization.

In order to simplify problem setup, the compliance minimization problem with a volume constraint will be used specifically. Although this somewhat reduces the generality of the mathematics involved, density and sensitivity filters can be defined agnostically from the problem setup, and so these findings may be easily reapplied to other minimization problems.
Chapter 2: Topology Optimization

2.1 Problem Formulation

The optimization problem treated in this paper is the maximization of global stiffness across a finite domain, given a constraint of maximum volume allowed. Due to the intractable nature of maximization of stiffness, minimization of the inverse, compliance, is considered instead. The objective of the minimization is to find a distribution of material volume across the design domain that minimizes the compliance for a given set of loads and boundary conditions.

As both the derivation of compliance and the resultant minimization problem are intractable on a continuous domain, the problem is discretized using the Finite Element Method. Since the elements are discretized, each is associated with its own density, \( \rho_e \). The ideal problem admits only density values \( \rho_e \in \{0,1\} \), however this results in a discrete optimization which must be solved via a state-space search. Due to the size of the problem, this approach is intractable, and so the density representation is relaxed to \( 0 \leq \rho_e \leq 1 \). Using the Finite Element Method, the optimization problem is represented as:

\[
\min_{\rho} \quad c(\rho) = f^T u \\
\text{s.t.:} \quad K(\rho) u = f \\
V(\rho) = \sum_{e=0}^{N_{el}} \rho_e v_e \leq V_c \\
0 \leq \rho_e \leq 1
\]

Where \( c \) is compliance, \( u \) is the global displacement vector, \( f \) is the vector of applied nodal forces, \( K(\rho) \) is the global stiffness matrix, \( \rho \) is the vector of element densities, \( v \) is the vector of absolute element volumes, \( N_{el} \) is the number of elements (assuming continuous element numbering), \( V(\rho) \) is the volume of material used, and \( V_c \) is target volume specified.

Although the density has been relaxed to a continuous representation, only densities of 0 (fully void) and 1 (fully solid) have physical meaning. To induce the optimization to approach the desired values, the continuous density parameter is mapped onto the stiffness matrix via a convex function modifying the Young’s modulus:

\[
E_e(\rho_e) = E_{\min} + (\rho_e)^p (E_0 - E_{\min})
\]
Where $E_0$ is the stiffness of the fully dense material, $E_{min}$ is a small value chosen to ensure a positive-definite stiffness matrix, and $p > 1$ is a penalization factor (usually $p = 3$) to make the function convex. This approach is commonly referred to as Modified Solid Isotropic Material with Penalization, or Modified SIMP. As stiffness is linear with respect to Young’s modulus, this leads to the modified minimization problem:

$$\min_{\rho} \ c(\rho) = \sum_{e=0}^{N_{el}} E_e(\rho_e)u_e^T k_0 u_e$$

Where $u_e$ is the elemental displacement vector and $k_0$ is the elemental stiffness matrix for an element with $E = 1$.

The optimization problem can be solved via multiple methods, but for simplicity the Optimality Criteria method is considered. The sensitivity can be derived as:

$$\frac{\partial c}{\partial \rho_e} = -pp_e^{p-1}(E_0 - E_{min})u_e^T k_0 u_e$$

As the compliance minimization problem is self-adjoint, a simple update scheme can be used:

$$\rho_e = \begin{cases} 0 & \text{if } \rho_e B_e < 0 \\ 1 & \text{if } \rho_e B_e > 1 \\ \rho_e B_e^\alpha & \text{otherwise} \end{cases}$$

$$B_e = \frac{-\frac{\partial c}{\partial \rho_e}}{\lambda \frac{\partial V}{\partial \rho_e}}$$

Where the Lagrangian multiplier $\lambda$ is chosen via the bisection algorithm, and $\alpha$ is a heuristic stabilization parameter (typically $\alpha = 1/2$).

### 2.2 Filtering

Due to the initial relaxation of the continuous optimization problem into the discrete Finite Element space, a notable solution instability exists (Sigmund, 1994). Since quadrilateral finite elements may be connected diagonally by a single node, it is possible for the optimization to generate semi-dense structures with “checkerboard” patterns, as shown in Figure 2.
Additionally, finite element analysis exhibits inaccuracy at the element-level. As such, it is desirable to perform mesh convergence studies to ensure accuracy of a given solution. This is impossible given standard topology optimization; resolution of a finer mesh allows for finer features, and so under a mesh convergence study, the optimized solution will change significantly, and cannot be assured to asymptote to an accurate solution (Sigmund & Petersson, 1998). This is demonstrated in Figure 3.

To simultaneously solve both of these issues, a filtering method is commonly employed. The optimization is reformulated on a new quantity generated via a convolution integral of the original finite element values. The filter thus modifies the calculation of the element sensitivities such that the optimization algorithm no longer has direct access to the element-level quantities. Sigmund first suggests this approach via a filtering of the element level sensitivities (Sigmund, 1994):

\[
\frac{\partial c'}{\partial \rho_e} = \frac{\sum_{i \in N_{el}} H_{el} \rho_i \frac{\partial c}{\partial \rho_i}}{\max(\rho_{min}, \rho_e) * \sum_{i \in N_{el}} H_{ei}}
\]
Where $N_{el}$ is a set of elements whose centerpoint distance from element $e$, $\delta x_{ei}$, is less than a given filter radius $r$, $\rho_{min}$ is a small positive number introduced to avoid division by zero, and $H_{ei}$ is a weighting function approximating a linear convolution integral:

$$H_{ei} = \max(0, r_{min} - \delta x_{ei})$$

A separate filtering approach has been suggested (Bruns & Tortorelli, 2001) where convolution is performed on the densities instead of the sensitivities and is then carried through to the convoluted sensitivity as well. The formulation for this filter is:

$$\rho'_e = \sum_{i \in N_{el}} \frac{H_{ei} \rho_i}{\sum_{i \in N_{el}} H_{ei}}$$

And the modified sensitivities are given by:

$$\frac{\partial c}{\partial \rho_i} = \frac{\partial c}{\partial \rho'_e} \frac{\partial \rho'_e}{\partial \rho_i}$$

Of these two approaches, density filtering is the most common, as it is considered more mathematically rigorous. It is noteworthy, however, that the sensitivity filtering method has a physical analogue in the Nonlocal Finite Element approach, while a similar justification does not exist for the density filter (Sigmund & Maute, 2001). Shown in Figure 4 are the results of applying the density and sensitivity filters to the typical Michell Beam (MBB) optimization problem. As both filtering approaches are used in the literature, both shall be considered here.

![Figure 4: 2D MBB Solution with Sensitivity Filtering (a) and Density Filtering (b)](image)
Chapter 3: Scale-Space Theory

3.1 Formalization of Scale

Although no formalization of scale exists for Topology Optimization, such a definition exists within the image processing community. To robustly process the information given by an image, it is important to construct a hierarchy of features that have been detected; this hierarchical representation is commonly referred to as the relative scale of features (Lindeberg, 1994a, pp.9-10). This scale information is used as a pre-processing step to generate a primitive representation of the features within an image, to which further image detection can be applied to identify those features, given the scale at which they reside. This computation significantly reduces the complexity of a generic image recognition algorithm, as it removes from consideration fine features that give irrelevant information.

To begin formalization, the image is abstracted as a generic signal. None of the features within the signal can be unambiguously identified as of a particular scale without a-priori knowledge or adoption of a set metric within the scale topology (Kuijper, Florack & Viergever, 2003). Thus, information on the relative scale of features must be constructed in order to determine the scale of a particular feature. This means that, in order to construct scale information for an image, there must exist some transformation that takes a scale input and suppresses information of either a lower or a higher scale. Although both approaches are valid, the image processing community adopts suppression of lower scale information as the standard (Lindeberg, 1994a, pp.10-11). The transformation is referred to as a scale-space smoothing operation, and the family of results constructed by this transformation are referred to as a multi-scale representation.

From the formal description of this transformation, three necessary conditions emerge to define an operation as belonging to the scale-space smoothing family (Koenderink, 1984):

1) **Causality** – as scale increases, the number of features within the signal should not increase.
2) **Homogeneity** – the transformation should not depend on the location in the signal.
3) **Isotropy** – the transformation operation should be independent of the original scale.

This idea of a multi-scale representation is directly compatible with the filtering operation used in topology optimization. Instead of extracting a range of scale information, however, the goal of filtering is to expose the optimization algorithm to potential features of a given scale or larger. As mentioned previously, this addition of scale allows for suppression of fine features, which removes instability and
inaccuracy introduced by the Finite Element approximation. It is clear, then, that the necessary conditions developed for scale-space smoothing of image features must also be satisfied by the filtering operation.

**3.2 Continuous Linear Scale-Space Theory**

Although the formalization of scale yields necessary conditions for a scale-space smoothing operation, these conditions have not been expressed mathematically; several assumptions must be made about the structure of the multi-scale representation and the scale-space smoothing operation in order to construct mathematical axioms (Witkin, 1983, Lindeberg, 1990, and Florack et al., 1992).

In linear scale-space theory, an image is considered specifically as a signal within the real function space, \( f: \mathbb{R}^2 \rightarrow \mathbb{R} \). This signal is used to construct a series of gradually smoother images, into which it is then imbedded. The series of images is then treated as time-evolving, such that the original image corresponds to a scale at \( t = 0 \), and as \( t \to \infty \), the scale of the smoothed image increases.

From the necessary conditions for a multi-scale representation, a series of axioms can now be expressed which can then be used to derive admissible scale-space transformations. Added to the necessary conditions described previously is the assumption that the scale-space transformation will behave linearly; this added assumption bounds the problem and allows for determination of a single valid smoothing function (Witkin, 1994). There are many different combinations of axiomatics that are valid for this case, but here the following will be used (Weickert et al., 1997):

- **Linear Integral Operator:**

  A scale-space transformation should be a linear transformation in order to preserve scale information. That is, as a pattern becomes uniformly more intense, the transformed pattern should also become uniformly more intense:

  \[
  \phi[A f(x^t), x_i, t] = A \phi[f(x^t), x_i, t]
  \]

  Every continuous linear functional may be written as an integral operator with an unknown functional. From this, the requirement of linearity can be reinterpreted as a requirement for a linear integral operator (Florack, Neissen & Nielsen, 1998). Thus, there exists a family of admissible kernel functions such that:
\[(T_t f)(x_i) = \int_{\mathbb{R}^N} k_t(x_i, x_i') f(x_i') \, dx_i'\]

- **Translation Invariance:**

Uniform translations of the underlying function should not affect the results of a scale-space operation; this implies that translation operations on the function should be order-invariant with the scale-space transformation. Let a uniform translation be defined by \((\tau_a f)(x) \equiv f(x - a)\). For a scale-space transformation, the following must be true:

\[(\tau_a T_t f)(x_i) = (T_t \tau_a f)(x_i), \quad \forall \ a \in \mathbb{R}^N, \quad \forall \ t > 0\]

Often, this axiom is combined with the Linear Integral Operator to form a single axiom, referred to as the convolution integral:

\[(T_t f)(x_i) = \int_{\mathbb{R}^N} k_t(x_i - x_i') f(x_i') \, dx_i'\]

- **Semigroup Property:**

An image within a scale-space should still be a discrete image, to which a scale-space operation can be further applied. Given that property, cascading applications of a scale-space transformation should still fall within the scale space of the original function. This implies that:

\[(T_{t+a} f)(x_i) = T_t (T_a f)(x_i), \quad \forall \ t, a \geq 0 , \forall f\]

- **Regularity:**

A scale-space transformation must be sufficiently smooth to create a continuous family of transformed images from an image. Additionally, from the definition of the method, at \(t = 0\), \((T_t f)(x_i) = f(x_i)\). This implies that:

\[As \ t \to 0^+ : \left\| (T_t f)(x_i) \right\|_{L_1} = \delta(x).\]

- **Causality:**

A scale-space transformation should not create new features as \(t \to \infty\). As features within computer vision are defined by maxima and minima, this implies that a time-evolving scale-space transformation should not generate new local extrema. This requirement can be shown to imply that the diffusion equation:

\[\partial_t u = \alpha(x, t) \Delta u\]
Must be satisfied at all of the extrema (Koenderink, 1984).

Since the transformation may be applied to all admissible images, or the entire function space, this condition may be generalized to imply that the transformation must universally satisfy the diffusion equation.

- **Isometry Invariance**

An admissible scale-space transformation should be directionally invariant, such that if the basis of the function space is swapped, the transformation yields the same result. Let \( R \in \mathbb{R}^N \) be a purely orthogonal transformation and define \( (Rf)(x) := f(R(x)) \); this restriction implies that:

\[
T_t(Rf)(x_i) = R(T_tf)(x_i), \quad \forall f, \forall t > 0
\]

Taken as a whole, these axioms require a convolution integral which is the Green’s function of the homogeneous linear diffusion equation (Lindeberg, 1994a, pp. 47-52). Since the Gaussian kernel is the unique solution for this case, it follows that the only admissible linear scale-space for the continuous case is:

\[
\mathcal{L}(x; t) = \int_{\xi \in \mathbb{R}^N} g(\xi; t)f(x - \xi)\, d\xi
\]

where \( g(\xi; t) \) is the Gaussian kernel:

\[
g(\xi; t) = \frac{1}{(2\pi t)^{\frac{N}{2}}} e^{-\frac{\xi^T \xi}{2t}}
\]

### 3.3 One-Dimensional Discrete Extension

The previous axiomatics, and the scale-space transformation derived from them, are only valid when assuming a continuous and infinite function space. Scale-space, however, must be applied to signals which are discretized and which have finite boundaries. Thus, the convolution integral must be modified into a kernel function. To begin, the one-dimensional case with infinite support is considered, where \( n \) is a discretization number:

\[
\mathcal{L}(x; t) = \sum_{n=-\infty}^{\infty} g(n; t)f(x - n)
\]

There are three potential approaches to this necessary discretization (Köthe, 2004):
1. Apply the continuously formulated axiomatics to the discrete space, and treat summations in
   the discrete space as integral approximations via the Riemann sum method.
2. Reconstruct an approximation of the continuous field represented by the discrete signal.
3. Reformulate the previous axiomatics within a discrete function space.

The first approach is common within the field of image processing and requires no further treatment of
the situation. However, a scale-space formulation that uses this assumption becomes increasingly
inaccurate as the scale nears the size of the discretization (Weiss, 1994). Thus, this approach is not well-
suited to Topology Optimization, which has a relatively coarse discretization. The second approach is
capable of easily allowing for scale-space filtering, however, topology optimization is currently
formulated for a discrete density distribution, and so use of this approach would require a
reformulation. For these reasons, use of the fully discrete scale-space theory is preferable.

As mentioned previously, a multi-scale representation is a function space which represents the original
function at varying levels of scale; assuming the underlying signal is discrete, the scale parameter, \( r \),
which maps the function space onto that signal may either be discrete \( r \in \mathbb{Z}_+ \) or continuous \( r \in \mathbb{R}_+ \), which yields two different formulations. In both cases, the following assumptions are made
(Lindeberg, 1990):

- The scale-space shall be constructed by a convolution integral as in the continuous formulation.
- An increase of the scale parameter shall correspond to an increase in the scale size of the
  features within the function. This implies the causality axiom as described in the continuous
  formulation.
- All admissible functions should be real-valued such that \( \mathbb{Z} \to \mathbb{R} \) is defined on an infinite grid.

Although derivation in N-dimensional space is intractable, derivation of a discrete kernel can easily be
performed in one-dimensional space. From the three assumptions given, the following discrete 1-D
axioms may be derived (Lindeberg, 1994a, pp. 65-67):

- **Positivity:** All coefficients of a discrete scale-space kernel must have the same sign
- **Unimodality:** The coefficients of a discrete kernel must be unimodal; that is, continuously
  ascending towards the center of the convolution
- **Semigroup Property:** As in continuous scale-space, if two discrete kernels \( K_a \) and \( K_b \) are within
  scale space, then the kernel \( K_a \ast K_b \) must also be within scale space
From these properties, it can be shown that for a discrete scale-space kernel, \( K \), the discrete Fourier transformation of the original signal:

\[
\psi_K(\theta) = \sum_{n=-\infty}^{\infty} K(n)e^{-in\theta}
\]

Must be non-negative and unimodal within the range \([-\pi, \pi]\) (Lindeberg, 1994a, pp.67-69). Through these properties, it can be further proven that a discrete kernel on a uniform discretization is a scale-space kernel iff its generating function \( \varphi_K(z) = \sum_{n=-\infty}^{\infty} K(n)z^n \) is of the form (Lindberg, 1990):

\[
\varphi_K(z) = cz^k e^{(q_1z^{-1}+q_2z)} \prod_{i=1}^{\infty} \frac{(1 + \alpha_i z)(1 + \delta_i z^{-1})}{(1 - \beta_i z)(1 - \gamma_i z^{-1})}
\]

\[
c > 0, \quad k \in \mathbb{Z}, \quad q_{-1}, q_1, \alpha_i, \beta_i, \gamma_i, \delta_i \geq 0
\]

\[
\beta_i, \gamma_i < 1, \quad \sum_{i=1}^{\infty} (\alpha_i + \beta_i + \gamma_i + \delta_i) < \infty
\]

By isolating components of this form, it can be shown that admissible discrete 1D scale-space kernels must only be a combination of any of the following primitive operations:

- **Two-point weighted average filtering**
  \[
f_{\text{out}}(x) = f_{\text{in}}(x) + \alpha_i f_{\text{in}}(x - 1), \quad (\alpha_i \geq 0)
  \]
  \[
f_{\text{out}}(x) = f_{\text{in}}(x) + \delta_i f_{\text{in}}(x + 1), \quad (\delta_i \geq 0)
  \]

- **First-order recursive filtering**
  \[
f_{\text{out}}(x) = f_{\text{in}}(x) + \beta_i f_{\text{out}}(x - 1), \quad (0 \leq \beta_i < 1)
  \]
  \[
f_{\text{out}}(x) = f_{\text{in}}(x) + \gamma_i f_{\text{out}}(x + 1), \quad (0 \leq \gamma_i < 1)
  \]

- **Diffusion smoothing**
  \[
f_{\text{out}}(x) = \sum_{n=-\infty}^{\infty} T(n, 0) f_{\text{in}}(x + n)
  \]

- **Rescaling**
  \[
f_{\text{out}}(x) = c \ f_{\text{in}}(x), \quad (0 < c)
  \]

- **Translation**
  \[
f_{\text{out}}(x) = f_{\text{in}}(x \pm 1)
  \]
Diffusion smoothing is performed as the discrete analogue to the Gaussian kernel, where $I_n$ are the modified Bessel functions of integer order, and are related to the ordinary Bessel functions $J_n$ via (Haddad & Akansu, 1991):

$$I_n(t) = I_{-n}(t) = (-i)^n J_n(it)$$

### 3.4 N-Dimensional Discrete Extension

The extension of the one-dimensional formulation to n-dimensional discrete space is not directly apparent, since a proof via counterexample exists that there are no non-trivial kernels on $\mathbb{R}^2$ or $\mathbb{Z}^2$ which cannot introduce new local extrema (Lifshitz & Pizer, 1987).

This requirement, however, was a sufficient condition derived from the causality axiom, and can be relaxed. Instead of direct creation of new local maxima and minima, causality can be maintained by ensuring that a transformation neither increases a local maximum nor decreases a local minimum. Although this will allow for the creation of plateaus, no new maxima or minima can be created from those plateaus, and so causality will be maintained. Note that this relaxation is only consistent for a continuous scale parameter, as the noncreation proof is limit-based, and so requires a differentiable space with respect to scale.

Given a point $x \in \mathbb{Z}^N$ within a uniform discretized grid of size 1, its connected neighbors are given via

$$N(x) = \left\{ \xi \in \mathbb{Z}^N : ||x - \xi||_\infty \leq 1 \land (\xi \neq x) \right\}$$

within this neighborhood, we define a weak local maximum of $g: \mathbb{Z}^N \rightarrow \mathbb{R}$ as satisfying

$$g(x) \geq g(\xi) \ \forall \ \xi \in N(x)$$

and weak local minimum as satisfying

$$g(x) \leq g(\xi) \ \forall \ \xi \in N(x)$$

Taking advantage of this relaxation, it can be proven that, in one, two, and three dimensions respectively, a scale-space transformation satisfies the differential equations (Lindeberg, 1994b):

$$\partial_t L = \alpha_1 \nabla_2^2 L$$

$$\partial_t L = \alpha_1 \nabla_2^2 L + \alpha_2 \nabla_{x_3}^2 L$$
\[
\partial_t L = \alpha_1 \nabla_1^2 L + \alpha_2 \nabla_2^2 L + \alpha_3 \nabla_3^2 L
\]

For some constants \( \alpha_1 \geq 0, \alpha_2 \geq 0, \) and \( \alpha_3 \geq 0. \)

The \( \nabla_i \) operators correspond to the discrete analogues to the Laplacian operator for various computational molecules. Using the simplifying discrete notation \( f_{i,j,k} = f(x_1 + i, x_2 + j, x_3 + k) \), these are, for 1D, 2D, and 3D respectively:

\[
(\nabla_1^2 f)_{0,0,0} = f_{-1,0,0} - 2f_0 + f_1
\]

\[
(\nabla_1^2 f)_{j,0,0} = f_{j-1,0,0} + f_{j+1,0,0} + f_{j,0,0} - 4f_0
\]

\[
(\nabla_1^2 f)_{0,0,j} = \frac{1}{2}(f_{-1,j-1} + f_{-1,j+1} + f_{1,j-1} + f_{1,j+1} - 4f_0)
\]

\[
(\nabla_1^2 f)_{0,0,0} = f_{-1,1,0} + f_{1,1,0} + f_{0,0,0} - 6f_0
\]

\[
(\nabla_1^2 f)_{0,0,0} = \frac{1}{4}(f_{-1,1,1} + f_{-1,1,-1} + f_{-1,-1,1} + f_{-1,-1,-1} + f_{1,1,1} + f_{1,1,-1} + f_{1,-1,1} + f_{1,-1,-1} - 8f_0)
\]

\[
(\nabla_1^2 f)_{0,0,0} = \frac{1}{4}(f_{-1,1,0} + f_{-1,-1,0} + f_{1,1,0} + f_{1,-1,0} + f_{-1,0,1} + f_{1,0,1} + f_{0,1,0} + f_{0,1,-1} + 12f_0)
\]

Considering the 2D case for simplicity, the differential equation may be discretized using Euler’s explicit timestep. This yields a linear convolution kernel with a computational molecule, or local convolution operation, of the form (Lindeberg, 1994a, pp. 114-117):

\[
\begin{pmatrix}
\frac{1}{4} \gamma \Delta t & \frac{1}{2} (1 - \gamma) \Delta t & \frac{1}{4} \gamma \Delta t \\
\frac{1}{2} (1 - \gamma) \Delta t & 1 - (2 - \gamma) \Delta t & \frac{1}{2} (1 - \gamma) \Delta t \\
\frac{1}{4} \gamma \Delta t & \frac{1}{2} (1 - \gamma) \Delta t & \frac{1}{4} \gamma \Delta t
\end{pmatrix}
\]

In the case of separable dimensions within the convolution kernel, a higher-dimensional scale-space transformation is equivalent to repeated applications of the one-dimensional scale-space concept along each coordinate direction. When the dimensions are regarded as separable, the following transformation represents the admissible discrete scale-space transformations:
3.5 Nonexistence of Circular Convolution

Spatial isotropy is a necessary property to define a scale-space transformation. In the continuous case, a necessary component of spatial isotropy is rotational invariance of the transformation. In discrete scale-space theory, however, radial symmetry no longer has useful meaning. Because the sampling points are not radially symmetric, the convolution kernel necessarily cannot preserve radial symmetry. It is unclear, then, that rotational invariance remains a useful condition for the enforcement of spatial isotropy (Lindeberg, 1994a, pp. 117-118).

However, since a discrete Fourier transform produces a continuous function, a separable convolution kernel can be shifted into the Fourier domain, converted to radially symmetric coordinates, and the variation can be minimized. For a 2D signal, the generic convolution kernel generating function is of the form:

\[
C_T(z, w) = \sum_{(m,n) \in \mathbb{Z}^2} T(m,n; t) z^m w^n = e^{-(2-\gamma)t + \frac{1}{2} (1-\gamma)(z^{-1}+z+w^{-1}+w)} + \frac{1}{4} r (z^{-1}w^{-1}+z^{-1}w+zw^{-1}+zw)
\]

The discrete Fourier transformation of this kernel is:

\[
C_T(e^{-iu}, e^{-iv}) = e^{-(2-\gamma)t + (1-\gamma) (\cos u + \cos v)} t + (\gamma \cos u \cos v) t
\]

Converting this transformation to polar coordinates yields:

\[
C_T(\omega \cos \phi, \omega \sin \phi) = e^{h(\omega \cos \phi, \omega \sin \phi)}
\]

\[
h(...) = -(2-\gamma) + (1-\gamma)(\cos(\omega \cos \phi) + \cos(\omega \sin \phi)) + \gamma \cos(\omega \cos \phi) \cos(\omega \sin \phi)
\]

In order to determine the error in radial symmetry, the Taylor expansion of \(h\) for small values of \(\omega\) is calculated (Lindeberg, 1991, Appendix A.2.3):

\[
h(...) = -\frac{1}{2} \omega^2 + \frac{1}{24} (1 + (6\gamma - 2) \cos^2 \phi \sin^2 \phi) \omega^4 + O(\omega^6)
\]
Where \( O(\omega^6) \) is an error term that also depends on \( \phi \) and \( \gamma \). When \( \gamma = \frac{1}{3} \), variation in \( \Delta \) with respect to \( \phi \) is confined to the 6\(^{th}\)-order error term, and thus decreases two orders of magnitude more quickly than for any other choice of parameter. This analysis shows that there is a finite radial asymmetry generated by reformulation of a scale-space convolution in a discrete domain. This inaccuracy, however, does decrease as a function of the size of the discretization, and through careful selection of the discrete approximation to the convolution, it is possible to minimize the error introduced.

### 3.6 Limitations of Scale-Space for Image Processing

The treatment summarized here has been derived assuming a uniform discretization on a square grid within an infinite space. Scale-space theory has not been developed for additional cases, as image data is largely discretized into a uniform grid of pixels, so the need for further development of the theory past this very specific case has been unnecessary. The assumption of a square grid, for instance, can potentially be removed and the same derivation can be performed on a non-uniform spatial grid. The underlying mathematics, however, becomes significantly more complex, and it is not clear if a meaningful derivation of the discrete scale-space filter exists for the general unstructured case (Lindeberg, 1994a, pp. 121).

More importantly, scale-space theory assumes an infinite domain, when all practical problems have finite data available. The most conservative solution to this problem is to regard any kernel that falls outside of the boundaries as invalid, however this is intractable, as the most accurate kernels have infinite support. A genuinely finite derivation of the scale-space theory has not been formally proven or disproven, however the inaccuracy introduced by truncation of the kernel has been negligible for image processing purposes (Lindeberg, 1994a, pp. 120, Köthe, 2004 and Kuijper, 2002).

This assertion does not hold true, however, on the boundaries of an image, as a significant portion of the filter is truncated. To solve this issue of excessive filter truncation near the bounds of the image, the image frame is selected such that the objects of interest lie within the center, and scale-space violations on the edge of the volume are unimportant to the overall operation (Lindeberg, 1994a, pp. 121 and Kuijper, 2002).
Chapter 4: Extension of Linear Scale-Space to Filtering

4.1 Formulation

As noted previously, the filtering operation in Topology Optimization seeks to uniformly enforce a scale onto an optimization domain. As this scale can vary continuously depending on the problem setup, this requires that the filtering operation create a valid multi-scale representation. If the scale parameter is assumed to behave linearly, the requirement for a multi-scale representation means that a filtering operation must obey linear scale-space.

From the classical scale-space theory as applied to images, color levels in the image are treated as a signal, and then generalized as functions within the real space; \( f: \mathbb{Z}^N \rightarrow \mathbb{R} \). As noted previously, the transformation takes the form:

\[
L(x; t) = \sum_{n=-\infty}^{\infty} g(n; t)f(x - n)
\]

Considering the density filtering approach for the 2D Topology Optimization case, the relevant filtering equations are, again:

\[
\rho'_e = \frac{\sum_{i \in N_{el}} H_{ei} \rho_i}{\sum_{i \in N_{el}} H_{ei}}, \quad H_{ei} = \max(0, r_{min} - \delta x_{ei})
\]

The density field is a discrete function space that maps to a continuous value, thus: \( \rho_i(x_i): \mathbb{Z}^N \rightarrow \mathbb{R} \). To describe the transformation as a continuous function, the maximum operation on \( H_{ei} \) is dropped in favor of a kernel operation restricted to the local neighborhood, \( N \):

\[
\rho'_e(x_e, r_{min}) = \frac{\sum_{i \in N} (r_{min} - \delta x_{ei}) \rho_i(x_i)}{\sum_{i \in N} (r_{min} - \delta x_{ei})}
\]

This function involves both two-point weighted average filtering and rescaling, which are admissible scale-space transformations within the discrete space. Via the semi-group property of scale-space transformation, a transformation constructed from repeated scale-space transformations is itself a transformation (Lindeberg, 1994a, pp.41-42). Thus, density filtering can be established as a scale-space transformation applied to the density field of the topology optimization for the 1D case. This ensures that the filter operation is uniformly imposing a scale onto the optimization, but only for a perfectly radially symmetric or a separable kernelling operation.
Considering the sensitivity filtering approach, the relevant equations are:

\[
\frac{\partial c'}{\partial \rho_e} = \frac{\sum_{i \in \mathbb{E}} H_{ei} \rho_i \frac{\partial c}{\partial \rho_i}}{\max(\rho_{\min}, \rho_e) \times \sum_{i \in \mathbb{E}} H_{ei}}, \quad H_{ei} = \max(0, r_{\min} - \delta x_{ei})
\]

Removing the maximum operation \(\max(\rho_{\min}, \rho_e)\), which is used for numerical stabilization, and limiting the kernel to a local neighborhood, \(N\), allows the equation to be rearranged slightly:

\[
\frac{\partial c'}{\partial \rho_e} = \frac{\sum_{i \in \mathbb{E}} (r_{\min} - \delta x_{ei}) \rho_i(x_i) \frac{\partial c}{\partial \rho_i}(x_i)}{\rho_e(x_e) \times \sum_{i \in \mathbb{E}} (r_{\max} - \delta x_{ei})}
\]

For the continuous case, this sensitivity filter can be more clearly expressed using a nonlocal finite element formulation. Assuming a nonlocal strain energy density function \(\psi = \epsilon_{ijkl} \epsilon_{ijkl}\), the sensitivity filter takes the form (Sigmund & Maute, 2012):

\[
\frac{\partial c'}{\partial \rho_e} = \frac{\int_{\Omega} (r_{\min} - \delta x_{ei}) w(x_i) \, dV}{\rho_e(x_e) \int_{\Omega} (r_{\max} - \delta x_{ei}) \, dV}
\]

This operation is not a linear scale-space operation; the presence of the normalization term \(\rho_e(x_e)\) in the denominator of the convolution integral results in nonlinearity. Additionally, this convolution simultaneously maps two different function spaces into a single scale-space, and so it cannot be considered a standard linear scale-space transformation. This does not exclude sensitivity filtering from being a nonlinear scale-space operation, however a detailed analysis of this behavior is well beyond the scope of this work.

In summary, mapping both sensitivity and density filtering onto the scale-space method shows that, while sensitivity filtering does have a consistent physical explanation, it does not obey linear scale-space, and therefore it cannot be shown to enforce a multi-scale representation via existing scale-space theory. Thus, for the remainder of this work, only the density filtering approach will be considered.

### 4.2 Convolution Kernel Eigenvalue Analysis

For any generic signal which is being convolved, a multi-scale representation must preserve causality, which is the general property that representations of the signal at higher scale are less complex. For scale-space theory with a continuous scale parameter, this requires the noncreation of new local
maxima or minima with an increasing value of scale. Since the original signal is considered to be equal to the filtered result for a scale of $t = 0$, if it can be shown that a transformation at a particular scale creates new local maxima or minima relative to the original signal, that transformation violates causality, and is therefore not a scale-space transformation.

If a discrete transformation at a particular value of scale has a convolution matrix with negative eigenvalues with corresponding real eigenvectors, a component of the input signal will be inverted on output. A valid input signal may therefore be constructed as a combination of several eigenvectors which is made more complex on the output via inversion of this component (Lindeberg, 1994a, pp. 69-70). Therefore, if a convolution matrix has negative eigenvalues, then the corresponding transformation is not a scale-space operation.

As mentioned previously, assumption of radial symmetry on a discretely sampled mesh results in an inaccurate convolution that varies radially. This property, that a scale-space convolution may have no negative eigenvalues, allows for a straightforward verification of the issues arising from use of a circular convolution on a uniformly gridded discrete space.

To perform tests on a valid convolution matrix, a 20x20 grid of quadrilateral Finite Elements is constructed. This mesh will hold constant density values at each element, and the density field will be treated as discretized at the element centerpoints. The mesh and the corresponding centerpoints are shown in Figure 5.

Figure 5: Uniformly Gridded Mesh Topology (a) and the Corresponding Element Centroids (b)
A convolution matrix is constructed from this assuming a radially symmetric convolution with a tent function:

$$C_{ij} = \frac{r_{min} - r_{ij}}{\sum_{k \in N} (r_{min} - r_{ik})}, \quad r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Where $N$ is a set of element centers within the convolution radius. A second convolution matrix is constructed assuming a separable 2D convolution with a tent function:

$$C_{ij} = \frac{(r_{min} - |x_i - x_j|) \cdot (r_{min} - |y_i - y_j|)}{\sum_{k \in N} (r_{min} - |x_i - x_j|) \cdot (r_{min} - |y_i - y_j|)}, \quad r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Where $N$ is a set of element centers which obey the constraints:

$$|x_i - x_j| \leq r_{min}; \quad |y_i - y_j| < r_{min}$$

As shown by the histograms in Figure 6, for a regular grid, the radial convolution sampled on a 2D regular grid yields negative eigenvalues, while the separable convolution produces no negative eigenvalues. Also shown in Figure 7 are the histograms corresponding to a uniform mesh refinement; negative eigenvalues are still present, and have not decreased in either magnitude or frequency. This numerical test supports the assertion that radial symmetry of the convolution operation should not be enforced when the convoluted signal is discrete, as the convolution operation will not maintain radial symmetry. This enforcement may result in scale-space violations which persist under mesh refinement.

*Figure 6: Histogram of Eigenvalues of the Convolution Matrix for a Radially Symmetric Filter (a) and a Separable Filter (b)*
This finding can be further validated by performing eigenvalue analysis on an irregular mesh topology. An irregular polygonal mesh is generated using PolyMesher, a simple and robust Matlab code for polygonal mesh generation (Talischi et al., 2012). The mesh and its corresponding centroids are shown in Figure 8, and the histogram of eigenvalues for both the radially symmetric and separable filters are shown in Figure 9. As with the regular mesh topology, the unstructured polygonal mesh exhibits negative eigenvalues under circular convolution. This suggests that the negative eigenvalues of the convolution matrix are generated by the enforcement of circular convolution on a discretized domain, rather than a characteristic of the gridded mesh topology.
Figure 8: Fully Unstructured Mesh Topology (a) and the Corresponding Element Centroids (b)

Figure 9: Histogram of Eigenvalues of the Convolution Matrix for a Radially Symmetric Filter (a) and a Separable Filter (b)
Chapter 5: Consequences of Scale-Space Enforcement

5.1 Boundary Errors

Boundary errors are of particular concern within topology optimization. Current filtering methods do not account for the presence of a boundary except via regularization to enforce a consistent volume. A filter that intersects a boundary truncates and regularizes to change shape significantly. Any kind of truncated filter is an ad-hoc approach, and does not obey scale-space (Köthe, 2004).

The image processing community addresses this issue by ensuring that scale-space violations will not significantly impact the analysis. This is done by selecting an image frame which has objects of interest exclusively in the center. Unfortunately, this is not a viable solution for topology optimization. Given an ideal optimization problem, it is expected that the algorithm will search across the entire function space provided, including topologies which interact with the boundaries (Ranier & Price, 1997). In the case that a scale-space violation produces an erroneous minimum, an optimization procedure that has a globalization scheme will be expected to always find this solution, given an infinite optimization period (Rozvany, 1998). This behavior ensures that topology optimization will be significantly more sensitive to scale-space violations than image scale pre-processing algorithms.

It has been shown in both topology optimization (Lazarov & Sigmund, 2010) and scale-space theory (Lindeberg, 1994a, pp. 40) that kernels have the same characteristic length if their 2nd moments are equal. As a topology optimization density kernel approaches the boundary and is truncated, the increasing slope of the tent results in a decreased second moment, and therefore a decreased length scale enforced on the boundaries. This property violates the basic isotropy requirement of a scale-space transformation, and so the standard filtering approach cannot be said to consistently enforce a scale on the optimization.

Viewed from an optimization perspective, the scale enforcement is a constraint on the problem. If the scale decreases at the edges of the volume, this corresponds to a weaker constraint on the edges of the volume. This results in topologies that uniformly attach themselves to the boundary geometry, as those solutions are artificially more optimal. This behavior is highlighted on the MBB solution in Figure 10.
To develop a system of equations to correct the decreased length scale, the geometry of a 1D linear kernel can be parameterized and corrected to ensure equal 2nd moments under increasing boundary truncation. As an additional constraint, the 0th moment is taken to be 1 for all filtering kernels to enforce volume preservation (Sigmund, 1994). The 1st moment of a kernel is the location that it is integrating, which must remain the point of interest (Dose & Guiochon, 1990). Thus, the zeroth, first, and second moments for a normal filter and a truncated filter must be held equal to enforce the same characteristic length and also apply the same filtering operation.

The continuous case will be considered here, as the standard topology optimization density filter is attempting to approximate a radially symmetric continuous kernel. When the expressions for the moment of a continuous tent function are combined with the parameterized kernel, the following system of equations is produced:

\[
\rho(x_i) = A - \frac{(|x_i| + B)}{C \cdot r_{min}} \quad \text{s.t.} \quad \begin{align*}
\int_{x_0}^{x_1} \rho(x_i) \, dx &= 1 \\
\int_{x_0}^{x_1} \rho(x_i) \cdot x_i \, dx &= 0 \\
\int_{x_0}^{x_1} \rho(x_i) \cdot x_i^2 \, dx &= \frac{1}{6} r_{min}^2
\end{align*}
\]

Where A is the peak of the adjusted tent function, B is the minimum value before truncation, C is a scaling factor on the initial filter radius, and \( x_0 \) and \( x_1 \) are the lower and upper bounds of the integral, respectively. An encroaching boundary is simulated by defining the bounds of integration as:

\[
x_0 = \max \left[ -C \cdot r_{min} , x_{\text{bound}} \right] \quad ; \quad x_1 = C \cdot r_{min}
\]
This nonlinear system can then be solved numerically to determine the shape of the kernel. Shown in Figure 11 is required scaling of the filter radius as a function of the boundary location.

![Figure 11: Required Scaling of the Radius of Convolution as a Function of Boundary Truncation](image)

As the boundary encroaches, the required filter radius to have a consistent characteristic length asymptotes and enters the imaginary space, indicating that a physically relevant correction does not exist for the 1D case using the continuous kernel. As previously noted, the separable higher-dimensional scale-space approximation is equivalent to repeated applications of the 1D case, and so the separable 2D filter also cannot enforce a consistent characteristic length on a boundary.

## 5.2 Unstructured Mesh Behavior

Although proof that a given transformation is a scale-space transformation is reasonable for a uniform grid, nonuniform mesh topologies hold very undesirable properties that prevent such a proof. Most importantly; the lack of regularity means that the structure of the discretization is unknown, and so the discrete Fourier transformation cannot be performed (Henderson & Karniadakis, 1995). This presents intractable problems when trying to ensure unimodality of a particular transformation in the frequency space, which is a necessary condition to establish a kernel as belonging to scale-space (Lindberg, 1990). Thus, it is impossible to construct a convolution kernel for a generic mesh.
Structured meshes may be related to a uniform grid via simple transformations, and therefore a scale-space kernel may be constructed via coordinate transforms. Unstructured meshes, however, hold no such property; in most cases, a one-to-one mapping to a uniform grid is impossible (Owen, 1998). Therefore, a scale-space transformation for an unstructured finite element mesh must be constructed on a case-by-case basis (Lindberg, 1994a, pp.40). Unfortunately, discrete scale-space for nonuniform topology remains an outstanding problem due to a lack of interest in the image processing community (Köthe, 2004).

In addition to unimodality issues, a nonuniform discretization will result in nonuniform sampling of a uniform convolution kernel, which violates the required scale-space property of isotropy. As with the truncated kernel on the boundary, this will cause variation in the 2nd moment of the convolution across the optimization. For the mesh topology shown in Figure 8, a surface representing the 2nd moment of the discrete kernel is given in Figure 12. As shown by the boundary truncation analysis, it is possible to correct this variation numerically, although since no applicable discrete scale-space formulation exists, it is impossible to ensure that such a correction will preserve unimodality.

![Figure 12: Scatter Plot (a) and Surface Plot (b) of the Convolution Kernel 2nd Moment for the Mesh Given in Figure 8](image)
Chapter 6: Convolution Filtering Modification

6.1 Boundary Extension:

Of the undesirable filter behavior explained via scale space, the most ubiquitous issue is the inconsistent length-scale enforced on the boundaries of the design space, which strongly violates scale-space theory. As all relevant problems in Topology Optimization have a bounded geometry, this issue cannot be avoided, and potentially impacts the results of all solutions for which it is not mitigated.

Within the field of image processing, this issue is avoided by selecting an image frame such that the objects of interest lie within the center. Singular applications of a filter which violates scale-space will result in local violations which do not impact the overall topology, and thus the central image can be said to obey scale-space. However, the topology optimization procedure requires repeated applications of the filter. With repeated applications of a scale-space transformation, violations propagate inwards and become large (Lindeberg, 1991). Thus, departure from a scale-space solution on the edges of the domain results in scale-space violations which propagate across the entire solution.

However, if the quantity for which scale is being enforced is guaranteed to be zero on the boundaries, then it can be trivially shown that the solution does not depart from an ideal solution which uniformly enforces the same scale as in the center of the optimization. This behavior can be enforced for a topology optimization problem by adding passive elements to the boundaries which cannot accept material, extending the domain as illustrated in Figure 13. As scale-space violations cannot propagate inwards, the domain only needs to be extended by the filter radius, so that elements which have density cannot be affected by the scale space violations. This “boundary extension” technique was previously suggested by Sigmund, but at the time was deemed unnecessary without detailed analysis (Sigmund, 2007).
6.2 Numerical 2D Testing

To numerically test the theory of Boundary Expansion, Sigmund’s 88-line Matlab Topology Optimization code (Andreassen et al., 2011) is modified to enforce a passivated element boundary. This modification is relatively straightforward; as the code uses the Optimality Criteria method, the maximum density value is enforced via the line:

\[
\text{xnew} = \max(0, \max(\text{x-move}, \min(\text{high}, \min(\text{x+move}, x.*\sqrt{-\text{dc}./\text{dv}/\text{lmid}})))));
\]

The variable \text{high} can thus be modified from a single value into a matrix of element-level quantities, which allows it to act as a selective mask on the maximum density. This masking is achieved by adding the following lines of code to the problem setup:

\[
\text{clamped} = 0.3;
\]
\[
\text{high} = \text{padarray}([1, 1], [\text{nelx}, \text{nely}], [\text{pl}, \text{pl}], \text{clamped});
\]

A larger mesh is then specified by hand to account for the newly passivated elements, and the location of the boundary conditions is modified accordingly; the full modified code may be found in Appendix A. Note that the element is not made fully passive; initial testing determined that the standard starting condition, which is a uniform initial distribution of material, resulted in an initial move which placed large amounts of material in the passive region. This material was then immediately removed via
enforcement of the passive element maximum density, and so the initial optimization step was highly discontinuous and often failed to converge.

To accommodate the straightforward method of boundary expansion, and also to simplify calculation of the boundary conditions, the standard half-MBB problem is reflected into the full-MBB problem. This new problem is then run via the commands:

\[
\text{top88 \ (120,20,0.5,3,2,2);}
\]

\[
\text{top88\_Expansion \ (120,20,0.5,3,2,2);}
\]

Although this problem is relatively simple compared to the contemporary problems in the literature (Sigmund & Maute, 2013), it was considered to be a good test of the method due to its ubiquity. Shown in Figure 14 is the resultant change in topology from performing boundary expansion on the MBB problem. The topology shifts significantly away from both boundary attachment and uniform gradation along the boundaries, both of which were hypothesized to be artifacts of a relaxed scale on the boundary. Although a heuristic observation, this behavior is consistent with the predictions of scale-space regarding boundary violations, which suggests that the boundary expansion method is performing as expected.

It is important to note that, since the standard problem formulation represents a relaxed set of constraints on the boundaries, comparison of the compliance between the two solutions is not meaningful. Ideally, the compliance should be lower for the standard problem, and this was observed for this example. However, this implementation does not include a globalization scheme, and so there is no guarantee that a particular optimization problem will not become stuck in a local minimum (Rozvany, 1998). This means that any individual problem cannot be guaranteed to have an expected behavior under boundary expansion.
Although there is volume present outside of the domain of interest, partial-passivation of the boundary elements combined with the material penalization model ensures that boundary elements will usually not contribute meaningfully to the compliance of the volume. This assumption can be ensured universally via continuation of the penalty parameter, which will increase the convexity of the density-stiffness curve and thereby decrease the contribution of the passive elements to the compliance (Bendsøe M, et al, 2004).

As an alternative approach that takes advantage of the modifications that have already been performed, the boundary elements can instead be progressively passivated. This will ensure that the initial convergence will be stable, while also guaranteeing that the final solution is accurate. The following code is added to the problem setup:

```matlab
contsteps=30;
dhigh = padarray(zeros(nely,nelx),[pl,pl],clamped/contsteps);
```

And then the following modification is performed to the optimization move limit after each iteration of the optimization:

```matlab
if (clampcont & loop<=contsteps) high = high - dhigh; end
```

Figure 15 shows the shift in topology generated by adding progressive passivation to the optimization.
As discussed in Section 3.5 and further explored in Section 5.2, scale-space theory also predicts that enforcement of a rotationally invariant filter on a gridded mesh will result in violation of scale-space and imprinting of the mesh on the final solution. For visualization of the overall topology, this instability will result in a loss of sharpness at transitions from solid to void material, as the noise generated by the filter will distort the underlying solution. To test this prediction, the filter calculation is modified with separable dimensions:

\[ s_H(k) = \max(0, r_{\text{min}} - \text{abs}(i_1-i_2)) \times \max(0, r_{\text{min}} - \text{abs}(j_1-j_2)) \]

Note that this filter must be corrected to impose the same characteristic length, as the second moment is not the same. Equating the 2\textsuperscript{nd} moments:

\[
\int_0^{r_1} \int_0^{2\pi} \frac{(r_1 - r)}{3\pi r_1^2} \times r^3 \, d\theta \, dr = \int_{-r_2}^{r_2} \int_{-r_2}^{r_2} |r_2 - x| \times |r_2 - y| \times (x^2 + y^2) \, dy \, dx
\]

\[
r_2 = \frac{3}{2\sqrt{5}} r_1^2
\]

This corrected filter radius is then applied to the separable filter in order to produce comparable results. Figure 16 shows the shift in topology for both the traditional and the boundary-expanded solution when the separable filter is employed.
Although discrete scale-space is formulated for a Gaussian convolution kernel, the findings from that analysis suggest that, as larger values of scale are used, uncorrected kernels will depart from a kernel that samples a radially symmetric convolution. As such, a uniform mesh refinement is performed on the boundary-expanded solution to determine if a coarser scale parameter yields departure in the solution topology. The solutions are shown in Figure 17.

Unlike the coarser problem, the two solutions are no longer identical. However, the solutions exhibit no large shift in topology. Although unexpected, this result is still reasonable under scale-space; the negative eigenvalues exhibited in the filters analyzed in Section 4.2 were small, which indicates they are minor components of the output signal. As scale-space theory is formulated for a generic signal, a kernel which violates scale-space may potentially still enforce a valid scale parameter for a specific signal (Lindeberg, 1991).

An important conclusion, however, can be drawn from this test; the compatibility of the radially symmetric approximation and the separable kernel allows them to be used interchangeably. This supports the assertion from scale-space theory that discretization of the domain results in a breakdown of the usefulness of rotational isotropy. It is suggested, then, that the separable filter may be used interchangeably with the radially symmetric approximation on a discretized domain, which remains useful for reducing the computational complexity of filter assembly.
6.3 Numerical 3D Testing

As characteristic length is defined by the 2\textsuperscript{nd} moment of area, an increase in the number of dimensions will result in a 2\textsuperscript{nd}-order increase in the severity of the boundary truncation error. Thus, it is expected that the effect of boundary truncation will be much greater for a 3D optimization than for a corresponding 2D problem.

To test this behavior, boundary extension is added to the Top3d Matlab code, which is a 3D extension of the 88-line code used in the 2D analysis (Liu & Tovar, 2014). As the code architecture is very similar to the 88-line code, the modifications for boundary extension can be easily reapplied. The modified Top3d code is included in Appendix B.

To test the ubiquity of this boundary truncation issue and demonstrate the applicability of the boundary expansion solution, the default problem for the code is run, which is a thin Michell Beam optimization. The specific problem shown here was run using the commands:

```
top3d (60,20,4,0.3,3,1.5);
top3d_Expansion (60,20,4,0.3,3,1.5);
```

The standard and boundary-extended results are shown in Figure 18 and Figure 19 respectively. As predicted, the solution topology changed significantly under boundary expansion. The standard optimization produces a topology which is mostly uniform through the thinnest dimension. Where the topology is nonuniform, material is separated onto the boundaries of the optimization volume. A clear solution instability exists as well on the portion of the optimization nearest the loading condition. An asymmetric feature with poor connectivity has been generated on a symmetric problem; this indicates some sort of weak dual-minima in the solution space, one of which has been selected by computational inaccuracy.
By comparison, the boundary-expanded solution lumps material into larger elements in the center of the volume, which is expected behavior under an enforcement of scale. The result of the optimization shows more feature complexity in the X-Y plane and more closely mirrors the expected Michell Truss topology (Hegemier & Prager, 1969). Most importantly, the solution is now resolving significant features through the thin dimension, including a progressive taper of the structure as it moves away from the fixed boundary conditions.
Chapter 7: Conclusions

This goal of this research was to determine the cause of geometry and mesh dependency that has been noted when applying Topology Optimization to general problems outside of academic research. This behavior was initially hypothesized to be an artifact of the filtering operation, as the shifts in geometry were, global, and not local. As the filter is intended to enforce a scale on the optimization domain, a mathematically consistent explanation of this scale was sought in order to determine if there were inconsistencies in enforcement.

Formalization of the general concept of scale was performed using the scale-space theory from computer vision, which was discussed at length. From this theoretical treatment, it was hypothesized that mesh irregularity and filter truncation at the boundaries of the design volume resulted in an inconsistent enforcement of scale across the optimization domain, which lead to instabilities in the solution. As the domain and discretization changed, the irregularities changed as well, and thus a given optimization solution was found to be tightly coupled to the design volume and the mesh topology.

Of the two issues highlighted, the most ubiquitous and most pressing was found to be the boundary sensitivity, and so it was the focus of the remaining investigation. Based on an original hypothesis by Sigmund, the design volume was padded with additional elements which were passive, or not allowed to accept volume. This would ensure that inaccuracies created by a truncated filter would have no impact on the convergence of the overall solution, and thus the optimization would not depart from a theoretical result for which the boundaries are treated consistently.

This procedure was implemented and tested on standard problems in both the 2-dimensional Top88 code and the 3-dimensional Top3d code. The problems tested produced strong shifts in converged topology. No meaningful quantitative comparisons could be made between the standard solutions and the boundary-expanded solutions, as the inconsistent boundary filtering represented a targeted relaxation of the problem constraints which would ideally result in more rapid convergence to a lower minimum. Qualitatively, however, the optimization produced results with more complex topology that more closely mirrored the ideal solutions found via other mathematical treatment of the problems.

From the results obtained by boundary expansion, it is evident that truncation of the filter on the boundaries represents an intractable constraint on the uncorrected optimization. The inconsistent enforcement of scale on the boundaries of the volume artificially relaxes the length-scale constraint and...
results in structures which preferentially attach themselves to those boundaries. Once volume expansion has been performed, solutions that were a result of relaxation may experience strong topology shifts. These findings call into question the validity of any uncorrected optimization which has significant interaction with its boundaries.
Chapter 8: Future Work

For this preliminary treatment of scale, only the standard, linear Topology Optimization filter was considered. The purpose of this decision was to simplify and focus the analysis, and to ensure that the conclusions generated would be widely applicable to existing Topology Optimization codes. After significant preliminary analysis, it was determined that the most critical scale-space violation to correct was the boundary truncation issue, as preliminary efforts to remedy the issue resulted in strong shifts in global topology. Due to this specific focus on boundary issues, several theoretical issues have been briefly discussed which warrant further treatment.

8.1 Irregular Convolution Correction

In Section 5.2, it is mentioned that unstructured meshes which are discretely sampled violate both the isotropy and the unimodality requirements of scale-space, via a nonuniform 2nd moment of the convolution and the nonexistence of a general unimodality proof. As illustrated in Figure 11, a highly regular mesh topology or a mesh topology which is finely discretized will have very small spatial variance relative to the boundary truncation issue, and likely can be ignored. Mesh irregularity can therefore be regarded as a problem to be solved at the mesh-level, rather than via modification of the algorithm.

That being said, close coupling of the optimization to the mesh topology occasionally presents a challenge for practitioners. In the generic, non-research application, design domains are often intractable shapes for which a regular mesh cannot be produced. Features which are locally asymmetric, such as a required access hole in a casing, may imprint themselves onto the mesh topology and cause highly asymmetric results in a low-resolution optimization. Thus, it may be advantageous to derive a discrete scale-space theory for the unstructured case and use this to construct a convolution which properly enforces scale.

8.2 Continuous Reconstruction

As discussed in Section 4.1, one approach to scale-space theory is to reconstruct a continuous signal from an underlying discrete signal space and perform continuous convolution. This approach holds promise for correcting errors introduced by irregular meshing, and it also allows for use of truly rotationally symmetric convolution operations (Köthe, 2004 and Lindberg, 1990). To make full use of a
continuous density field, however, the discrete problem must be reformulated in the continuous space, or an advantageous sampling method must be developed.

### 8.3 Nonlinear Scale-Space

The scale-space theory discussed within this paper is constructed on the assumption of a linear scale parameter. The assumption is not required for a convolution to be a scale-space transformation, and nonlinear diffusion schemes have been investigated for use in a nonlinear scale-space theory (Perona & Malik, 1990). These nonlinear enforcements of scale may have properties which are beneficial for Topology Optimization, most notably preservation of sharp gradients, or edges, within a filtered topology. This will allow enforcement of scale via diffusion of the high-level topology without the coupled issue of loss of distinct boundaries on the low-level topology.
References


Appendix A: Modified Top88 Matlab Code

function top88_Expansion(nelx,nely,volfrac,penal,rmin,ft)

%% MATERIAL PROPERTIES
E0 = 1;
Emin = 1e-5;
u = 0.3;

%% Parameters to add pillowing
pl = ceil(rmin); %Determines the thickness of the pillow
clamped = 0.3; %Determines the starting value of clamping
realintegration = 1; %Performs volume integration on only the real volume, not the border
realcompliance = 1; %Masks the compliance of the pillow volume

%Activate the separable filter
separableFilter = 0;

%Parameters to control continuation on the penalty or the clamping:
penalcont=0;
clampcont=0; contsteps=20;

npelx=nelx+pl*2;
npely=nely+pl*2;
norig=nelx*nely;

%% PREPARE FINITE ELEMENT ANALYSIS
A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
A12 = [-6 -3 0 3; -3 -6 -3 -6 0 -3 -6 3; 3 -6 3 -6];
B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 4 -3; 9 4 -3 -4];
B12 = [2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
KE = 1/(1-nu^2)/24*(A11 A12;A12' A11)+nu*[B11 B12;B12' B11];
nodenrs = reshape(1:(1+npelx)*(1+npely),1+npely,1+npelx);
edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,npelx*npely,1);
edofMat = repmat(edofVec,1,8)+repmat([0 1 2*npely+2 3 0 1 -2 -1],npelx*npely,1);
iK = reshape(kron(edofMat,ones(8,1))',64*npelix*npely,1);
jK = reshape(kron(edofMat,ones(1,8))',64*npelix*npely,1);

% Calculate initial DOF offset on the left side due to pillowing
regoff=2*(pl*(npely+1)+pl);

% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
F = sparse(regoff+2*(npely+1)*((nelx/2)+2,1,-1,2*(npely+1)*(npelix+1),1);
U = zeros(2*(npely+1)*(npelix+1),1);
fixeddofs = [regoff+2*(nely+1)-1;regoff+2*(nely+1);2*(npely+1)*(npelix+1)-
2*(npely+1)*pl-2*pl];
freedofs = setdiff(alldofs,fixeddofs);

% PREPARE FILTER
iH = ones(npelix*npely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for il = 1:nelix
    for jl = 1:nply
        e1 = (il-1)*nplex+jl;
        for i2 = max(il-(ceil(rmin)-1),1):min(il+(ceil(rmin)-1),npelix)
            for j2 = max(jl-(ceil(rmin)-1),1):min(jl+(ceil(rmin)-1),npely)
                e2 = (i2-1)*nplex+j2;
                k = k+1;
                iH(k) = e1;
                jH(k) = e2;
                if separableFilter
                    sH(k) = max(0,rmin-abs(il-i2))*max(0,rmin-abs(jl-j2));
                else
                    sH(k) = max(0,rmin-sqrt((il-i2)^2+(jl-j2)^2));
                end
            end
        end
    end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);

% Assemble the matrix of element maxima
high = padarray(ones(nely,nelx),[pl,pl],clamped);
dhigh = padarray(zeros(nely,nelx),[pl,pl],clamped/contsteps);

%Filters for selectively operating on volumes:
realV = padarray(ones(nely,nelx),[pl,pl],0);
nrv = sum(realV(:));
wasteV = padarray(zeros(nely,nelx),[pl,pl],1);

%% INITIALIZE ITERATION
x = padarray(repmat(volfrac,nely,nelx),[pl,pl],1e-5);
beta = 1;
if ft == 1 || ft == 2
   xPhys = x;
elseif ft == 3
   xTilde = x;
   xPhys = 1-exp(-beta*xTilde)+xTilde*exp(-beta);
end
xPhys = x;
loop = 0;
loopbeta = 0;
change = 1;

%% START ITERATION
while change > 0.01
   loop = loop + 1;
   loopbeta = loopbeta + 1;
   % FE-ANALYSIS
   sK = reshape(KE(:)*(Emin+xPhys(:).^penal*(E0-Emin)),64*npelx*npely,1);
   K = sparse(iK,jK,sK); K = (K+K')/2;
   U(freedofs) = K(freedofs,freedofs)\F(freedofs);
   % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
   if realcompliance
      ce = realV.*reshape(sum((U(edofMat)*KE).*U(edofMat),2),npely,npelx);
   else
      ce = reshape(sum((U(edofMat)*KE).*U(edofMat),2),npely,npelx);
   end
   c = sum(sum((Emin+xPhys.^penal*(E0-Emin)).*ce));
   dc = -penal*(E0-Emin)*xPhys.^(penal-1).*ce;
   dv = ones(npely,npelx);
%% FILTERING/MODIFICATION OF SENSITIVITIES
if ft == 1
dc(:) = H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
elseif ft == 2
dc(:) = H*(dc(:)/Hs);
dv(:) = H*(dv(:)/Hs);
elseif ft == 3
dx = beta*exp(-beta*xTilde)+exp(-beta);
dc(:) = H*(dc(:).*dx(:)/Hs);
dv(:) = H*(dv(:).*dx(:)/Hs);
end
%% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES AND PHYSICAL DENSITIES
l1 = 0; l2 = 1e9; move = 0.25;
while (l2-l1)/(l1+l2) > 1e-3
lmid = 0.5*(l2+l1);
xnew = max(0,max(x-move,min(high,min(x+move,x.*sqrt(-dc./dv/lmid)))));
if ft == 1
xPhys = xnew;
elseif ft == 2
xPhys(:) = (H*xnew(:))/Hs;
elseif ft == 3
xTilde(:) = (H*xnew(:))/Hs;
xPhys = 1-exp(-beta*xTilde)+xTilde*exp(-beta);
end
if realintegration
if sum(realV(:).*xPhys(:)) > volfrac*norig, l1 = lmid; else l2 = lmid;
end
else
if sum(xPhys(:)) > volfrac*norig, l1 = lmid; else l2 = lmid; end
end
change = max(abs(xnew(:)-x(:)));
x = xnew;
if ft == 3 && beta < 512 && (loopbeta >= 50 || change <= 0.01)
beta = 2*beta;
loopbeta = 0;
change = 1;
fprintf('Parameter beta increased to %g \n',beta);
end
if (clampcont & loop<=contsteps)
    high = high - dhigh;
end
%
PRINT RESULTS.
fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f Clamp:%7.3f ch.:%7.3f\n', ... 
    loop,c,dot(xPhys(:),realV(:))/nrv,high(1),change);
%
PLOT DENSITIES
colormap(gray); imagesc(1-xPhys); caxis([0 1]); axis equal; axis off;
drawnow;
end
Appendix B: Modified Top3D Matlab Code

function top3d_Expansion(nelx,nely,nelz,volfrac,pmax,rmin)
%Parameters to add volume expansion
pl = ceil(rmin); %Determines the thickness of the expansion
clamped = 0.3; %Determines the starting value of clamping
realintegration = 1; %Performs integration on only the original volume
realcompliance = 1; %Masks the compliance of the expanded volume

%Parameter to control continuation on the penalty or the clamping:
penalcont=0;
clampcont=0; contsteps=20;

npelx=nelx+pl*2;
npely=nely+pl*2;
npelz=nelz+pl*2;
norig=nelx*nely*nelz;

% USER-DEFINED LOOP PARAMETERS
maxloop = 200;    % Maximum number of iterations
tolx = 0.01;      % Terminarion criterion
displayflag = 0;  % Display structure flag

% USER-DEFINED MATERIAL PROPERTIES
E0 = 1;           % Young's modulus of solid material
Emin = 1e-5;      % Young's modulus of void-like material
nu = 0.3;         % Poisson's ratio

% USER-DEFINED LOAD DOFs
[iil,jl,kl] = meshgrid(nelx+pl, 0, pl:nelz+pl);              % Coordinates
loadnid = kl*(npelx+1)*(npely+1)+il*(npely+1)+(npely+1-jl); % Node IDs
loaddof = 3*loadnid(:) - 1;                                 % DOFs

% USER-DEFINED SUPPORT FIXED DOFs
[iif,jf,kf] = meshgrid(0,pl:nely+pl,pl:nelz+pl);              % Coordinates
fixednid = kf*(npelx+1)*(npely+1)+iif*(npely+1)+(npely+1-jf); % Node IDs
fixeddof = [3*fixednid(:); 3*fixednid(:)-1; 3*fixednid(:)-2]; % DOFs

% PREPARE FINITE ELEMENT ANALYSIS
nele = npelx*npely*npelz;
ndof = 3*(npelx+1)*(npely+1)*(npelz+1);
F = sparse(loaddof,1,-1,ndof,1);
U = zeros(ndof,1);
freedofs = setdiff(1:ndof,fixeddof);
KE = lk_H8(nu);
nodegrd = reshape(1:(npely+1)*(npelx+1),npely+1,npelx+1);
nodeids = reshape(nodegrd(1:end-1,1:end-1),npely*npelx,1);
nodeidz = 0:(npely+1)*(npelx+1):(npelz-1)*(npely+1)*(npelx+1);
nodeids = repmat(nodeids,size(nodeidz))+repmat(nodeidz,size(nodeids));
edofVec = 3*nodeids(:)+1;
edofMat = repmat(edofVec,1,24)+ ...
   repmat([0 1 2 3*npely + [3 4 5 0 1 2] -3 -2 -1 ...
           3*(npely+1)*(npelx+1) + [0 1 2 3*npely + [3 4 5 0 1 2] -3 -2 -1]],nele,1);
iK = reshape(kron(edofMat,ones(24,1))',24*24*nele,1);
jK = reshape(kron(edofMat,ones(1,24))',24*24*nele,1);
% PREPARE FILTER
iH = ones(nele*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for k1 = 1:npelz
    for i1 = 1:npelx
        for j1 = 1:npely
            e1 = (k1-1)*npelx*npely + (i1-1)*npely+j1;
            for k2 = max(k1-(ceil(rmin)-1),1):min(k1+(ceil(rmin)-1),npelz)
                for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-
                    1),npelx)
                    for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-
                        1),npely)
                        e2 = (k2-1)*npelx*npely + (i2-1)*npely+j2;
                        k = k+1;
                        iH(k) = e1;
                        jH(k) = e2;
                        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2+(k1-
                            k2)^2));
                    end
                end
            end
        end
    end
end
end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);

% Prepare the clamping for the expanded volume
high = padarray(ones(nely,nelx,nelz),[pl,pl,pl],clamped);
dhigh = padarray(zeros(nely,nelx),[pl,pl],clamped/contsteps);

% Filters for calculating volume fractions:
realV = padarray(ones(nely,nelx,nelz),[pl,pl,pl],0);
nrv = sum(realV(:));
wasteV = padarray(zeros(nely,nelx,nelz),[pl,pl,pl],1);

% INITIALIZE ITERATION
x = padarray(repmat(volfrac,[nely,nelx,nelz]),[pl,pl,pl],0.0001);

xPhys = x;
loop = 0;
change = 1;

% START ITERATION
while change > tolx && loop < maxloop
    loop = loop+1;
    if penalcont
        if loop <= 20, penal = 1; else penal = min(pmax,1.02*penal); end
    else
        penal = pmax;
    end
    % Make the descending part of the clamping start at 10% of max iterations
    if clampcont
        if loop >= 0.1*maxloop, high = max(high - increment*wasteV,clamped); end
    else
        penal = pmax;
    end
    % FE-ANALYSIS
    sK = reshape(KE(:)*(Emin+xPhys(:)'.^penal*(E0-Emin)),24*24*nele,1);
    K = sparse(iK,jK,sK); K = (K+K')/2;
    U(freedofs,:) = K(freedofs,freedofs)\F(freedofs,:);

% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
ce = reshape(sum((U(edofMat)*KE).*U(edofMat),2), [npely, npelx, npelz]);
c = sum(sum(sum((Emin+xPhys.^penal*(E0-Emin)).*ce)));
dc = -penal*(E0-Emin)*xPhys.^(penal-1).*ce;
dv = ones(npely, npelx, npelz);

% FILTERING AND MODIFICATION OF SENSITIVITIES
dc(:) = H*(dc(:)/Hs);
dv(:) = H*(dv(:)/Hs);

% OPTIMALITY CRITERIA UPDATE
l1 = 0; l2 = 1e9; move = 0.25;
while (l2-l1)/(l1+l2) > 1e-3
    lmid = 0.5*(l2+l1);
    xnew = max(0.0001, max(x-move, min(high, min(x+move, x.*sqrt(-
dc./dv/lmid)))))
    xPhys(:) = (H*xnew(:))/Hs;
    if sum(xPhys(:)) > volfrac*norig, l1 = lmid; else l2 = lmid; end
end
change = max(abs(xnew(:)-x(:)));

% PRINT RESULTS.
fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f Clamp:%7.3f
 ch.:%7.3f
', loop, c, dot(xPhys(:), realV(:))/nrv, high(1), change);
fflush(stdout);

% PLOT DENSITIES
if displayflag, clf; display_3D(xPhys); end %#ok<UNRCH>
end
clf; display_3D(xPhys);
end

% === GENERATE ELEMENT STIFFNESS MATRIX ===
function [KE] = lk_H8(nu)
A = [32 6 -8 6 -6 4 3 -6 -10 3 -3 -3 -4 -8;
    -48 0 0 -24 24 0 0 0 12 -12 0 12 12 12];
k = 1/144*A'*[1; nu];
\[
\begin{align*}
K_1 &= \begin{bmatrix} k(1) & k(2) & k(3) & k(5) & k(5); \\
k(2) & k(1) & k(4) & k(6) & k(7); \\
k(2) & k(2) & k(1) & k(4) & k(7) & k(6); \\
k(3) & k(4) & k(4) & k(1) & k(8) & k(8); \\
k(5) & k(6) & k(7) & k(8) & k(1) & k(2); \\
k(5) & k(7) & k(6) & k(8) & k(2) & k(1) \end{bmatrix}; \\
K_2 &= \begin{bmatrix} k(9) & k(8) & k(12) & k(6) & k(4) & k(7); \\
k(8) & k(9) & k(12) & k(5) & k(3) & k(5); \\
k(10) & k(10) & k(13) & k(7) & k(4) & k(6); \\
k(6) & k(5) & k(11) & k(9) & k(2) & k(10); \\
k(4) & k(3) & k(5) & k(2) & k(9) & k(12); \\
k(11) & k(4) & k(6) & k(12) & k(10) & k(13) \end{bmatrix}; \\
K_3 &= \begin{bmatrix} k(6) & k(7) & k(4) & k(9) & k(12) & k(8); \\
k(7) & k(6) & k(4) & k(10) & k(13) & k(10); \\
k(5) & k(5) & k(3) & k(8) & k(12) & k(9); \\
k(9) & k(10) & k(2) & k(6) & k(11) & k(5); \\
k(12) & k(13) & k(10) & k(11) & k(6) & k(4); \\
k(2) & k(12) & k(9) & k(4) & k(5) & k(3) \end{bmatrix}; \\
K_4 &= \begin{bmatrix} k(14) & k(11) & k(11) & k(13) & k(10) & k(10); \\
k(11) & k(14) & k(11) & k(12) & k(9) & k(8); \\
k(11) & k(11) & k(14) & k(12) & k(8) & k(9); \\
k(13) & k(12) & k(12) & k(14) & k(7) & k(7); \\
k(10) & k(9) & k(8) & k(7) & k(14) & k(11); \\
k(10) & k(8) & k(9) & k(7) & k(11) & k(14) \end{bmatrix}; \\
K_5 &= \begin{bmatrix} k(1) & k(2) & k(3) & k(5) & k(4); \\
k(2) & k(1) & k(8) & k(4) & k(6) & k(11); \\
k(8) & k(8) & k(1) & k(5) & k(11) & k(6); \\
k(3) & k(4) & k(5) & k(1) & k(8) & k(2); \\
k(5) & k(6) & k(11) & k(8) & k(1) & k(8); \\
k(4) & k(11) & k(6) & k(2) & k(8) & k(1) \end{bmatrix}; \\
K_6 &= \begin{bmatrix} k(14) & k(11) & k(7) & k(13) & k(10) & k(12); \\
k(11) & k(14) & k(7) & k(12) & k(9) & k(2); \\
k(7) & k(7) & k(14) & k(10) & k(2) & k(9); \\
k(13) & k(12) & k(10) & k(14) & k(7) & k(11); \\
k(10) & k(9) & k(2) & k(7) & k(14) & k(7); \\
k(12) & k(2) & k(9) & k(11) & k(7) & k(14) \end{bmatrix}; \\
KE &= 1/((\nu+1)*(1-2*\nu)) * \ldots \begin{bmatrix} K_1 & K_2 & K_3 & K_4 \end{bmatrix};
\end{align*}

K2'  K5  K6  K3';
K3'  K6  K5'  K2';
K4  K3  K2  K1');
end

% === DISPLAY 3D TOPOLOGY (ISO-VIEW) ===
function display_3D(rho)

[npely,npelx,npelz] = size(rho);

hx = 1; hy = 1; hz = 1; % User-defined unit element size
face = [1 2 3 4; 2 6 7 3; 4 3 7 8; 1 5 8 4; 1 2 6 5; 5 6 7 8];

set(gcf,'Name','ISO display','NumberTitle','off');
for k = 1:npelz
    z = (k-1)*hz;
    for i = 1:npelx
        x = (i-1)*hx;
        for j = 1:npely
            y = npely*hy - (j-1)*hy;
            if (rho(j,i,k) > 0.5) % User-defined display density threshold
                vert = [x y z; x y-hx z; x+hx y-hx z; x+hx y z; x y z+hx; x y-
                        hx z+hx; x+hx y-hx z+hx; x+hx y z+hx];
                vert(:,[2 3]) = vert(:,[3 2]); vert(:,2,:) = -vert(:,2,:);
                patch('Faces',face,'Vertices',vert,'FaceColor',[0.2+0.8*(1-
                        rho(j,i,k)),0.2+0.8*(1-rho(j,i,k)),0.2+0.8*(1-rho(j,i,k))]);
            end
        end
    end
end

axis equal; axis tight; axis off; box on; view([30,30]); pause(1e-6);
end