Revisiting approximate reanalysis in topology optimization: On the advantages of recycled preconditioning in a minimum weight procedure

Oded Amir
Faculty of Civil and Environmental Engineering, Technion - Israel Institute of Technology
Tel.: +972-4-8293041
odedamir@cv.technion.ac.il

This is an unformatted version of the paper, published online in SMO, May 2014.

1 Abstract

An efficient procedure for three-dimensional continuum structural topology optimization is proposed. The approach is based on recycled preconditioning, where multigrid preconditioners are generated only at selected design cycles and re-used in subsequent cycles. Building upon knowledge regarding approximate reanalysis, it is shown that integrating recycled preconditioning into a minimum weight problem formulation can lead to a more efficient procedure than the common minimum compliance approach. Implemented in MATLAB, the run time is roughly twice faster than that of standard minimum compliance procedures. Computational savings are achieved without any compromise on the quality of the results in terms of the compliance-to-weight trade-off achieved. This provides a step towards integrating interactive 3-D topology optimization procedures into CAD software and mobile applications. MATLAB codes complementing the article can be downloaded from the author's personal webpage.

Keywords : Topology optimization; Recycled preconditioning; Reanalysis of structures; Combined approximations

2 Introduction

This article addresses the efficient computational solution of a classical problem in structural topology optimization of continua: Find the material distribution that represents the best trade-off between stiffness and weight. The respective problem formulations typically involve a very large number of design variables and only a few constraints. For such cases the nested approach is often preferable, meaning that optimization is performed in the design variables only and that the equilibrium equations are treated as function calls and solved separately. Then the computational effort invested in optimization is often dominated by the effort involved in repeated structural analyses, i.e. solving a sequence of large systems of linear equations. For 2-D problems direct sparse solvers, typically based on *LU* or Cholesky decompositions, can be very efficient and are often preferred (e.g. Davis, 2006). However, for 3-D problems of medium and large scale, solution time and memory requirements of direct solvers increase significantly. Then iterative equation solvers hold an advantage, in particular if parallel computations are considered (Saad, 2003). The linear equation systems arising from linear elastic modeling are symmetric and positive-definite. Therefore from the family of iterative Krylov subspace methods, the preconditioned conjugate gradients method (PCG) is a natural choice for 3-D structural topology optimization.

The desire to speed up computational topology optimization procedures motivates several complementary research trajectories: Examining the application of multiple computational scales and resolutions, thus avoiding the inherent high cost of solving sequences of finite element analyses on a fine mesh; developing highly scalable computational procedures that can fully exploit the increasing availability of high performance parallel computers; and deriving effective procedures that can reduce the computational effort invested in sequential structural analysis. Contributions to each of these research efforts are reviewed in the following.

Multi-resolution multi-scale topology optimization where optimization is performed while progressively refining the grid was initially suggested by Kim and Yoon (2000). This work was later extended by Kim et al (2003) who applied adaptive wavelet-Galerkin analysis to multi-scale topology optimization. Stainko (2006) proposed a multilevel scheme where optimization is first performed on a coarse grid, which is then adaptively refined in the interface between solid and void. More recently, a multi-resolution topology optimization (MTOP) scheme was suggested, where the density

distribution is realized on a finer scale than the displacements (Nguyen et al, 2010, 2012). MTOP is in fact currently implemented in the TopOpt app for achieving high resolution designs at a low computational expense (Aage et al, 2013).

Early implementations of parallel topology optimization procedures focused on achieving parallel scalability - meaning that for a given problem size computational speedup is inversely proportional to the number of processors. Parallel implementation of minimum-compliance structural topology optimization was first accomplished by Borrvall and Petersson (2001). They used a domain decomposition strategy and PCG with Jacobi (diagonal) preconditioning, yielding 85%-95% parallel efficiency on up to 24 processors. A similar parallelization strategy was employed by Kim et al (2004), but for more demanding objectives involving maximization of eigenvalues. Vemaganti and Lawrence (2005) compared the parallel performance of three solvers: PCG with Jacobi preconditioning; PCG with incomplete LU (ILU) preconditioning; and a hybrid iterative-direct substructuring solver. They found the hybrid solver to be most effective, due to the difficulty of the Jacobi preconditioner to mitigate ill-conditioning of the equation system and due to the sequential nature of the ILU solver. However, they discuss only 2-D problems whereas for 3-D cases the conclusions may have been different. Another contribution based on Jacobi PCG was provided by Mahdavi et al (2006), who by utilizing parallel computations achieved a speedup of up to 6 with 32 processors. Again, only 2-D problems were discussed. Excellent parallel scalability was achieved by Aage and Lazarov (2013), who developed an object-oriented parallel framework for large-scale 3-D topology optimization. Linear speedup was reported on up to 800 processors, partly due to full parallelization of the optimization algorithm (MMA, (Svanberg, 1987)).

The contributions reviewed above did not address one of the main challenges in implementing parallel topology optimization procedures, that is achieving numerical scalability: This requires that the computational complexity will increase linearly with problem size. Using Jacobi and incomplete factorization preconditioners, the number of PCG iterations increases with mesh refinement, meaning the procedure is not scalable regardless of the parallel scalability. An important step towards numerical scalability was provided by Evgrafov et al (2008), who implemented a parallel domain decomposition solver (FETI-DP, (Farhat et al, 2001)). The number of CG iterations performed still increased with the number of DOF, but less significantly than in previous studies: A few hundred CG iterations were required in large-scale 3-D problems compared to a few thousands as reported by Borrvall and Petersson (2001). As pointed out also by Aage and Lazarov (2013), the path towards parallel and numerical scalability passes through the combination of an effective parallel framework with advanced linear solvers such as FETI or multigrid methods (e.g Trottenberg et al, 2001). It should be noted that from the perspective of parallel processing, utilization of massively parallel GPU systems is also very promising. Effort has been invested recently in efficient implementation of topology optimization in GPU systems, for example Wadbro and Berggren (2009), Schmidt and Schulz (2011), Suresh (2013) and Zegard and Paulino (2013).

Apart from developing efficient parallel implementations, speedup of topology optimization can also be achieved by reducing the computational effort on the procedural level. An important contribution in this direction was provided by Wang et al (2007), who used an iterative MINRES solver for large-scale 3-D minimum compliance problems. Computational time was reduced by a combination of rescaling for avoiding ill-conditioning; preconditioning with incomplete factorization; and exploiting the small changes between subsequent linear systems by recycling Krylov subspaces. Another related approach is based on utilizing approximate reanalysis procedures, essentially performing a fully accurate structural analysis only at selected design cycles and faster approximate reanalysis otherwise. The full integration of such a method into minimum-compliance topology optimization procedures, following Kirsch's Combined Approximations (CA) approach (Kirsch, 1991), was initially presented by Amir et al (2009). An extension towards reanalysis-based topology optimization for free vibrations was formulated subsequently (Bogomolny, 2010). Kirsch's CA was integrated also into truss optimization with frequency constraints solved by genetic algorithms (Zuo et al, 2011).

The purpose of the current study is to revisit the integration of approximate reanalysis into topology optimization procedures and to offer significant improvements with respect to the original contribution. These build upon new insight coming from recent investigations on utilizing approximate reanalysis in robust topology optimization (Amir et al, 2012) and on multigrid preconditioned conjugate gradients (MGCG) solvers in 3-D topology optimization (Amir et al, 2013). The main difference between the proposed reanalysis-based approach and its predecessor is in adopting a minimum-weight formulation rather than a minimum-compliance formulation. The inherent computational advantage of the former is thoroughly discussed throughout the article. The current work is directed primarily towards facilitating efficient implementations on a single CPU so to serve the integration of 3-D topology optimization into CAD software, e.g. within the Grasshopper-Rhino TopOpt component released recently; as well as into applications for mobile devices such as the TopOpt app (Aage et al, 2013). Parallel procedures are not of primary concern in this work, but much of the conclusions arising are indeed applicable also to large-scale topology optimization problems. Most importantly, the MGCG solver was shown to exhibit numerical scalability in topology optimization problems: The number of MGCG iterations remains constant with mesh refinement (Amir et al, 2013).

Utilizing approximate reanalysis introduces certain errors in the design sensitivities of any functional involving displacements. Consistent sensitivity analysis that accounts for these errors was derived both in the context of traditional CA (Amir et al, 2009) as well as in the PCG framework (Amir et al, 2010, 2012). However, as pointed out in the abovementioned studies, accepting approximate sensitivities while controlling the accuracy of reanalysis is often a more effective

approach. Therefore in the current study only approximate design sensitivities are utilized and accuracy is ensured by monitoring one of the following measures: 1) The relative norm of residual forces; or 2) The error obtained in the design sensitivities, in particular using criterion (II) as defined by Amir et al (2013). Another possibility which may be explored in the future is to adopt the alternative convergence criterion for PCG which basically monitors the convergence of the required functional (Amir et al, 2010). For the particular case of compliance this resembles Arioli's stopping criterion for PCG (Arioli, 2004).

The central message communicated in this article is that once reanalysis is employed, it is advantageous to use a minimum weight problem formulation rather than the more popular minimum compliance formulation. The basis for this idea is described in Section 3 where we first review the general principles of reanalysis-based approaches and then present the motivation for adopting the minimum weight objective. The computational implementation is discussed in Section 4 where the proposed procedure is benchmarked versus existing approaches in 2-D. The benefits of the minimum weight formulation are demonstrated and the extension to 3-D using an MGCG framework is explained. Once the discussion is shifted towards the context of iterative solvers for 3-D problems, the concept of *reanalysis* is replaced by the similar idea of *recycled preconditioning*. Building upon the insight gained from studying reanalysis procedures in 2-D, significant savings are achieved in 3-D problems by recycling preconditioners within the MGCG framework. Several examples are presented in Section 5 and conclusions are drawn in Section 6.

3 Reanalysis-based approach to topology optimization

The article presents an efficient computational approach to the solution of a classical task in structural topology optimization: Find the material distribution that represents the best trade-off between stiffness and weight. This can be stated either as a minimization of compliance (maximization of stiffness) subject to a constraint on weight, or vice-versa - minimization of weight subject to a constraint on compliance. In principle, both approaches should lead to the same solution. In practice however, different solutions representing equivalent stiffness-to-weight trade-offs may be obtained. This is due to the non-convex nature of the topology optimization problem which possesses many local minima.

In an industrial setting, the choice between minimizing compliance or minimizing weight can depend on the specific purpose as well as on the information available to the design engineer. Based on existing designs and performance requirements, reasonable target values for constraining weight or compliance can be obtained. In the examples presented throughout the article, minimum compliance optimization is first performed, followed by a minimum weight optimization. This is in order to conduct a fair comparison of the resulting volume-to-compliance trade-offs obtained by the two formulations. For each test case, we first obtain a minimum compliance result that later serves as a benchmark. It is important to note that the relative advantages of the reanalysis-based approach are not related to the choice of executing compliance minimization first - this is merely in order to obtain a value for constraining the compliance, which in a true industrial setting will be available to the designer.

3.1 Standard problem formulations with accurate analysis

The two problem formulations are hereby briefly reviewed for the sake of completeness. As mentioned above, the nested approach is followed, meaning that structural analysis is separated from optimization. For simplifying the presentation, standard ingredients of topology optimization procedures are utilized. This does not imply any loss of generality because various computational approaches to topology optimization share the same basic formulation. We follow the density-based approach (Bendsøe, 1989) and apply the modified SIMP interpolation scheme relating density to elastic stiffness (Sigmund and Torquato, 1997)

$$E(\rho) = E_{min} + (E_{max} - E_{min})\rho^p$$

where $E(\rho)$ is Young's modulus of the interpolated material; ρ is the material density; E_{min} and E_{max} correspond to the two materials that are distributed in the design domain; and p is the penalization power. This representation is more general than the classical SIMP (Bendsøe, 1989; Zhou and Rozvany, 1991) which can be deduced by setting $E_{min} = 0$. Mesh-independence and regularization are treated by either sensitivity filtering (Sigmund, 1997; Sigmund and Maute, 2012) or density filtering (Bruns and Tortorelli, 2001; Bourdin, 2001). For distributing a single material, weight can be substituted by volume so the minimum compliance optimization problem has the following form

$$\begin{aligned}
\min_{\rho} \quad & f_c = \mathbf{f}^{\mathsf{T}}\mathbf{u} \\
\text{s.t.:} \quad & g_v = \sum_{e=1}^{N} v_e \bar{\rho}_e - V^{\star} \leq 0 \\
& 0 \leq \rho_e \leq 1 \qquad e = 1,...,N \\
\text{with:} \quad & \mathbf{K}(\bar{\rho})\mathbf{u} = \mathbf{f}
\end{aligned} \tag{MinC}$$

where ρ represents the mathematical (non-filtered) density distribution; $\bar{\rho}$ represents the physical (filtered) density distribution, which differs from ρ if density filtering is applied; $\mathbf{K}(\bar{\rho})$ is the stiffness matrix; $\mathbf{f}$ is the load vector; $\mathbf{u}$ is the displacements vector; v_e is the element volume; $V^\star$ is the total available volume; and N is the number of finite elements, which coincides with the number of design variables if no design restrictions are imposed. Swapping the objective and constraint gives the following minimum volume problem formulation

$$\begin{aligned} \min_{\rho} \quad & f_v = \frac{1}{V}\sum_{e=1}^{N} v_e \bar{\rho}_e \\ \text{s.t.:} \quad & g_c = \mathbf{f}^{\mathsf{T}}\mathbf{u} - c^\star \leq 0 \\ & 0 \leq \rho_e \leq 1 \qquad e = 1,...,N \\ \text{with:} \quad & \mathbf{K}(\bar{\rho})\mathbf{u} = \mathbf{f} \end{aligned} \tag{MinV}$$

where V is the total volume of the design domain and $c^\star$ is the required compliance. Design sensitivities of the compliance functionals are computed by the adjoint method and are given by

$$\frac{\partial f_c}{\partial \bar{\rho}_e} = \frac{\partial g_c}{\partial \bar{\rho}_e} = -\mathbf{u}^{\mathsf{T}} \frac{\partial \mathbf{K}}{\partial \bar{\rho}_\mathbf{e}} \mathbf{u}. \tag{1}$$

In case density filtering is applied, the derivatives with respect to the mathematical variables are then obtained by the chain rule. Using the computed gradients of the objectives and of the constraints, both problems can be solved by either optimality criteria (OC) procedures or by more general nonlinear programming methods.

For solving the minimum compliance problem we follow the 88-line code by Andreassen et al (2011) implemented in MATLAB (MATLAB, 2013). Within the OC procedure, a bi-section scheme is used there to ensure the linear volume constraint is satisfied. Solving the minimum volume problem by a similar optimality criteria approach requires approximating the nonlinear constraint. This leads to a slightly more involved treatment compared to the minimum compliance formulation. The procedure will be described in Section 4.

3.2 Employing approximate reanalysis procedures

In the process of sequential optimization it is required to solve the sequence of linear systems

$$\mathbf{K}(\bar{\rho}_k)\mathbf{u}_k = \mathbf{f} \tag{2}$$

where the index k corresponds to the design cycle. It is assumed here that $\mathbf{f}$ is design-independent but this does not imply any loss of generality. The term *reanalysis* refers to replacing Eq. (2) with

$$\left(\mathbf{K}(\bar{\rho}_{k-l}) + \Delta\mathbf{K}\right)\mathbf{u}_k = \mathbf{f}$$

where $\Delta\mathbf{K} = \mathbf{K}(\bar{\rho}_k) - \mathbf{K}(\bar{\rho}_{k-l})$ and the index $k-l$ corresponds to a certain design cycle preceding k. Later in the paper the stiffness matrix $\mathbf{K}(\bar{\rho}_{k-l})$ will be referred to as the *reference* stiffness matrix - corresponding to a design cycle where accurate analysis is performed. *Approximate reanalysis* aims at finding an approximation $\widetilde{\mathbf{u}}$ that will be sufficiently accurate for the purpose of optimization,

$$\begin{aligned} \widetilde{\mathbf{u}}_k &\approx \mathbf{u}_k \\ \mathbf{K}(\bar{\rho}_k)\widetilde{\mathbf{u}}_k &\approx \mathbf{f}. \end{aligned}$$

In Kirsch's Combined Approximations (CA) approach to approximate reanalysis, the displacements are expressed by the series

$$\begin{aligned} \widetilde{\mathbf{u}}_k \approx \Bigg(& \mathbf{I} - \mathbf{K}\left(\bar{\rho}_{k-l}\right)^{-1}\Delta\mathbf{K} + \left(-\mathbf{K}\left(\bar{\rho}_{k-l}\right)^{-1}\Delta\mathbf{K}\right)^2 + \\ & \left(-\mathbf{K}\left(\bar{\rho}_{k-l}\right)^{-1}\Delta\mathbf{K}\right)^3 + ... \Bigg)\mathbf{u}_{k-l} \end{aligned} \tag{3}$$

where an accurate solve is performed at design cycle $k-l$ yielding $\mathbf{K}(\bar{\rho}_{k-l})\mathbf{u}_{k-l} = \mathbf{f}$. The main feature of CA is the utilization of only a few series terms from Eq. (3). In traditional implementations, these serve as basis vectors in a reduced basis solution. For a detailed discussion of the method and its applications the reader is referred to the monographs by Kirsch (2002, 2008). Consistent integration of CA into topology optimization procedures was first presented by Amir et al (2009) for minimum compliance problems. The approach was later extended also for objectives involving eigenfrequencies (Bogomolny, 2010).

It was shown that the CA procedure is mathematically equivalent to a PCG procedure where the factors of $\mathbf{K}(\bar{\rho}_{k-l})$ serve as preconditioners (Kirsch et al, 2002). Assuming positive definiteness of both $\mathbf{K}(\bar{\rho}_{k-l})$ and $\mathbf{K}(\bar{\rho}_k)$, the Cholesky factorization $\mathbf{U}_{k-l}^{\mathsf{T}}\mathbf{U}_{k-l} = \mathbf{K}(\bar{\rho}_{k-l})$ is used to precondition the system of equations given in Eq. (2). Hence the concept of *reanalysis* can be seen as a specific form of the more general *recycled preconditioning*: The factorization $\mathbf{U}_{k-l}^{\mathsf{T}}\mathbf{U}_{k-l} = \mathbf{K}(\bar{\rho}_{k-l})$ corresponding to design cycle $k-l$ is re-used as a preconditioner in subsequent design cycles. Under certain conditions, such recycled preconditioning can yield an accurate solution, as is the case also with CA. In many cases however, the aim is to generate an approximation $\widetilde{\mathbf{u}}_k$ using only a small number of PCG iterations. This is equivalent to *approximate reanalysis* where only a few basis vectors are utilized. In the context of topology optimization, such a procedure was implemented using only one matrix factorization for the whole design process (Amir and Sigmund, 2011). The superior numerical robustness of the PCG implementation, in comparison with traditional CA implementations that utilize Gram-Schmidt orthogonalization, was recently demonstrated (Amir et al, 2012). The shift of terminology from *reanalysis* to *recycled preconditioning* is absolutely necessary once the extension to 3-D problems is discussed. This is because matrix factorizations become impractical and preconditioned iterative solvers are used throughout all design cycles. This point will be further elaborated in Section 4.3.

3.3 Advantages of stiff preconditioning

A successful application of approximate reanalysis relies to a large extent on the choice of the *reference* stiffness matrix which is the one decomposed by Cholesky factorization and then used to precondition subsequent systems. When applying approximate reanalysis for robust topology optimization it was shown that a very effective choice of a reference design is to use the stiffest design among the set to be evaluated - i.e. the most dilated in the context of robust topology optimization (Amir et al, 2012). This observation will be briefly reviewed here while more thorough derivations can be found in the referenced article. The effectiveness of using a stiffer reference design, or in other words *stiff preconditioning*, is related to some basic properties of the Combined Approximations approach.

Consider a demonstrative scenario where two different designs are to be evaluated - one accurately and the other by reanalysis. Their corresponding stiffness matrices are denoted $\mathbf{K}_1$ and $\mathbf{K}_2$ and let us assume that the first design is stiffer, i.e. $\mathbf{K}_1 \succ \mathbf{K}_2$. The main argument is that it is more effective to use $\mathbf{K}_1$ as the reference design rather than vice-versa. When $\mathbf{K}_1$ plays the role of the reference structure the series in Eq. (3) is guaranteed to converge, regardless of the magnitude of changes in stiffness $\Delta\mathbf{K} = \mathbf{K}_2 - \mathbf{K}_1$. In the opposite case where $\mathbf{K}_2$ is the reference design, divergence may occur if the difference in stiffness $\Delta\mathbf{K} = \mathbf{K}_1 - \mathbf{K}_2$ is large. Furthermore, for any given pair of designs convergence of the series in Eq. (3) is faster if the stiffer design plays the role of the reference design. This is because the convergence rate increases as the ratio between the norms of $\Delta\mathbf{K}$ and of the reference stiffness decreases. Consequently, also the PCG procedure converges faster if the preconditioner is stiffer than the evaluated design. For clarification, choosing an arbitrarily stiff reference design will not yield fast convergence, because the norm of $\Delta\mathbf{K}$ will be large. Maintaining small design changes between the reference and the evaluated design, thus small values in $\Delta\mathbf{K}$, is necessary for achieving good convergence. The advantage of stiff preconditioning merely suggests that for a given magnitude of $\Delta\mathbf{K}$, it is beneficial to use the *stiffer* design as the preconditioner.

The advantages of stiff preconditioning imply that significant improvements can be offered compared to the original integration of CA into topology optimization. Initially, compliance was minimized subject to a volume constraint. This means that as optimization progressed, the design in general stiffened by making better use of a constant available volume. Therefore in the context of reanalysis we had $\mathbf{K}(\bar{\rho}_{k-l}) \prec \mathbf{K}(\bar{\rho}_k)$, or in other words reanalysis was performed with a soft reference design because it utilized a factorization from an earlier design cycle. By switching to minimization of volume subject to a constraint on compliance, the benefits of stiff preconditioning can be exploited. Particularly in the early design cycles, material is removed until the constraint is active - meaning the design is in general softened as optimization progresses. As will be shown, also in later design cycles the slackness of the compliance constraint tends to decrease as optimization progresses. In other words, compliance in general increases meaning that reanalysis is very likely to involve a stiff preconditioner and a softer reanalyzed design, i.e. $\mathbf{K}(\bar{\rho}_{k-l}) \succ \mathbf{K}(\bar{\rho}_k)$. This leads to a very efficient minimum volume procedure which will be presented and demonstrated in the following section.

4 Reanalysis-based minimum volume formulation

This section is dedicated to the presentation and demonstration of the proposed reanalysis-based topology optimization procedure for minimizing volume subject to a compliance constraint. First, an adaptation of the common optimality criteria scheme is discussed. This is necessary for enabling efficient implementation in the relevant computational environments, namely Rhino-Grasshopper and mobile devices. Nevertheless the same reanalysis-based approach can be coupled with general nonlinear programming methods and the same computational benefits are expected. Accuracy and efficiency of the approach are discussed based on a demonstrative test case. The approach is compared against standard

procedures, i.e. involving full structural analysis, and against reanalysis-based procedures for minimizing compliance subject to a volume constraint.

4.1 Adapted optimality criteria

A very common approach for solving the minimum compliance problem presented in Eq. (MinC) is by applying the classical optimality criteria update scheme (e.g. Bendsøe and Sigmund (2003) and references therein). Within every design cycle, redistribution of material is driven by a constant strain energy density. The Lagrange multiplier corresponding to the linear volume constraint is then found by a simple bi-section scheme. The update of density distribution is consequently rescaled, followed by an update of the Lagrange multiplier. This is repeated until the volume constraint is satisfied up to a certain tolerance.

Minimizing volume subject to a compliance constraint introduces a certain difficulty due to the nonlinearity of the compliance functional. In principle, structural analysis is required after every update of the material distribution in order to evaluate the constraint correctly. This is clearly not preferable because it will increase the number of costly equilibrium analyses. Therefore it is suggested to apply the bi-section scheme on an approximate prediction of the constraint, based on first-order information in the current design cycle. This implies that in some design cycles the optimality criteria procedure may result in infeasible designs. Nevertheless, the approximate prediction becomes more accurate as optimization progresses because design changes decrease. This was also mentioned in the context of the COC algorithm (Rozvany and Zhou, 1992). Numerical experiments have been conducted with both linear and reciprocal approximations of the constraint, the latter inspired by the fundamental ideas behind CONLIN and MMA (Fleury and Braibant, 1986; Svanberg, 1987).

The implemented optimality criteria approach can be summarized as follows. Beginning with a full design domain, uniform reduction of density is performed in the first design cycles, until a linearized prediction of the constraint is violated. In all subsequent design cycles, a heuristic updating scheme is applied which is essentially the same as in common minimum compliance implementations (Sigmund, 2001; Andreassen et al, 2011)

$$\rho_e^{k+1}(\Lambda) = \begin{cases} \max(10^{-10}, \rho_e^k - m) & \text{if } \rho_e^k B_e^k(\Lambda) \leq \max(10^{-10}, \rho_e^k - m) \\ \min(1, \rho_e^k + m) & \text{if } \rho_e^k B_e^k(\Lambda) \geq \min(1, \rho_e^k + m) \\ \rho_e^k B_e^k & \text{otherwise} \end{cases}$$

with

$$B_e^k(\Lambda) = \left(-\frac{\Lambda}{\frac{\partial f_v}{\partial \rho_e}} \frac{\partial g_c}{\partial \rho_e}\bigg|_{\rho_e^k} \right)^{\frac{1}{2}}.$$

Here, Λ is the Lagrange multiplier corresponding to the compliance constraint; m is a user-defined move limit; and the superscripts k, $k+1$ represent design cycles. Using an iterative bi-section scheme, the value of Λ is determined such that an approximation of the compliance constraint is satisfied up to some tolerance. As mentioned above, it is suggested to utilize either linear or reciprocal approximations given by

$$\widetilde{g}_c(\Lambda) = g_c^k + \sum_{e=1}^{N} \frac{\partial g_c}{\partial \rho_e}\bigg|_{\rho_e^k} \left(\rho_e^{k+1}(\Lambda) - \rho_e^k \right) \approx 0$$

and

$$\widetilde{g}_c(\Lambda) = g_c^k + \sum_{e=1}^{N} \frac{\partial g_c}{\partial \rho_e}\bigg|_{\rho_e^k} \frac{\rho_e^k}{\rho_e^{k+1}(\Lambda)} \left(\rho_e^{k+1}(\Lambda) - \rho_e^k \right) \approx 0 \tag{4}$$

respectively, where g_c^k is the value of the compliance constraint at the k-th design cycle. In the numerical investigations, the latter provided slightly higher accuracy and exhibited more conservativeness with respect to violation of the true constraint. Note that a non-zero lower bound is imposed on ρ_e in order to allow division by ρ_e^{k+1}. The bi-section scheme is applicable for solving the minimum volume formulation just as it is for solving the minimum compliance formulation for the following reasons: 1) The constraint is a monotonously decreasing function of Λ because increasing any density value decreases compliance; and 2) At optimum, the constraint should be active, otherwise more material can be removed.

4.2 Accuracy and efficiency: A demonstrative test case

For demonstrating the benefits of the minimum volume formulation over the minimum compliance formulation, the following test case is examined. We seek the optimal topology of a two-dimensional cantilever beam with two point loads at the bottom of the free face, see Figure 1(a). The finite element (FE) mesh resolution is 200×100; the material

properties are $E_{min} = 10^{-9}, E_{max} = 1, \nu = 0.3$; the density filter radius is 2.0; and the modified SIMP penalty is 3.0. All finite elements are squares with side length of 1, meaning $v_e = 1$. Both problem formulations are solved using the optimality criteria approach with an iterative move limit of 0.05. In all the presented experiments, a fixed number of 200 design iterations was performed - the stopping criteria (maximum change in design smaller than 10^{-3}) was not satisfied.

First we establish reference results corresponding to standard topology optimization procedures where a sparse Cholesky factorization was utilized for solving the FE analysis equations. A volume fraction limit of 0.35 is imposed, meaning that the total volume available is $V^\star = 0.35 \times N$ where N is the number of finite elements. The minimum compliance procedure attained an objective value of $f_c = 136.1$. Executing a slightly modified version of the 88-line code (Andreassen et al, 2011), avoiding unnecessary matrix-vector multiplications within the OC procedure, the run time in MATLAB was 71.1 seconds. Minimizing volume with the achieved compliance of 136.1 imposed as a constraint resulted in an optimized volume fraction of $f_v = 0.35$. Within the bi-section scheme, the reciprocal approximation of the constraint as in Eq. (4) was employed. The actual compliance constraint was violated only once throughout 200 design iterations. As expected, the run time of the minimum volume program was similar to that of the minimum compliance code: 71.8 seconds. Thus the equivalence of both formulations in terms of computational effort is verified, for the case of accurate FE analyses. Nevertheless, slightly different optimized layouts are obtained, see Figure 1. This demonstrates the abundance of local minima in the vicinity of the global optimum. The results are conveniently summarized in the top two rows of Table 1. For clarification, the choice of OC procedures rather than more rigorous nonlinear programming does not result in any compromise with respect to the quality of the results. With the latest MATLAB implementation of MMA (Svanberg, 1987), the minimum compliance procedure attained an objective value of $f_c = 136.0$ - just marginally better than OC. The run time was 147.2 seconds - roughly double than with OC. The MMA-based minimum volume procedure with a compliance of 136.1 imposed as a constraint resulted in an optimized volume fraction of $f_v = 0.353$ - slightly worse than with OC. Again, the run time was significantly longer than with OC - 146.6 seconds.

Table 1: Summary of results for the demonstrative test case. The minimum volume formulation holds a significant computational advantage when approximate reanalysis is employed - as can be inferred from the number of matrix factorizations and PCG iterations.

Formulation	Objective	Normalized constraint	Matrix factorizations	PCG iterations
Min. compliance, standard	$f_c = 136.1$	$g_v = 3.16 \cdot 10^{-8}$ $V^\star = 0.35 \times N$	200	—
Min. volume, standard	$f_v = 0.35$	$g_c = -2.65 \cdot 10^{-6}$ $c^\star = 136.1$	200	—
Min. compliance, reanalysis	$f_c = 136.0$	$g_v = 7.21 \cdot 10^{-7}$ $V^\star = 0.35 \times N$	25	565
Min. volume, reanalysis	$f_v = 0.35$	$g_c = -3.11 \cdot 10^{-6}$ $c^\star = 136.1$	22	369

We now turn to show that when approximate reanalysis is employed, the minimum volume formulation holds a significant computational advantage. For both formulations, we set the following controls over the reanalysis procedure:

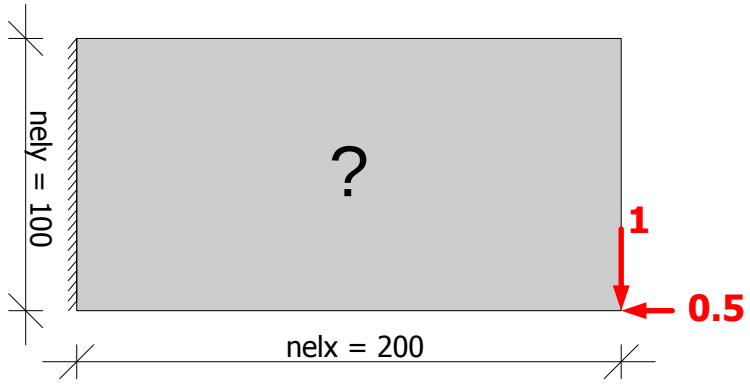
1. A standard Cholesky decomposition is performed every 10th design cycle, or if the approximation error (referring to the relative norm of residual forces $\frac{\|\mathbf{f}-\mathbf{K}\tilde{\mathbf{u}}\|}{\|\mathbf{f}\|}$) in the latest design cycle exceeded 10^{-2};

2. A maximum number of 5 PCG / reanalysis iterations is allowed;

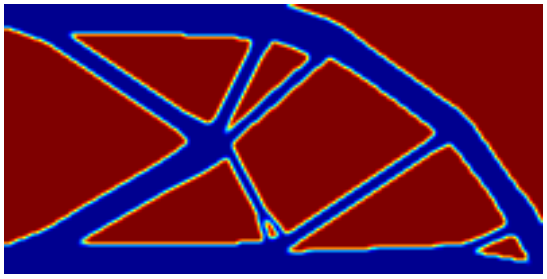
3. The PCG / reanalysis convergence tolerance is set to 10^{-3}, again referring to the relative norm of residual forces.

Imposing again $V^\star = 0.35 \times N$, the minimum compliance reanalysis-based procedure attained an objective value of $f_c = 136.0$. Throughout the 200 design cycles, 25 Cholesky decompositions were performed. A total number of 565 PCG / reanalysis iterations was performed within the 175 design cycles in which reanalysis was utilized, i.e. 3.23 PCG iterations per cycle in average. With a compliance constraint of 136.1, the minimum volume reanalysis-based procedure attained an optimized volume fraction of $f_v = 0.35$. The actual compliance constraint was marginally violated only a few times. Throughout the 200 design cycles, 22 Cholesky decompositions were performed. A total number of 369 PCG / reanalysis iterations was performed within the 178 design cycles in which reanalysis was utilized, i.e. 2.07 PCG iterations per cycle in average. These results are presented in the bottom two rows of Table 1.

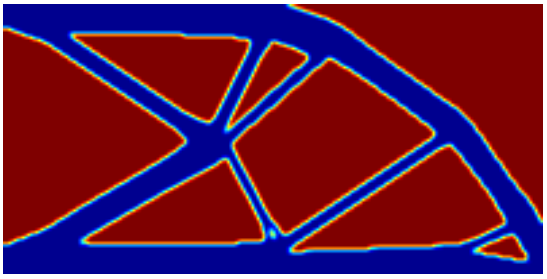
Several important observations arise from the experiments presented above. The first is that both reanalysis procedures succeed in reproducing the results of the fully accurate procedures. The actual accuracy-efficiency trade-off is of course related to the control measures employed, primarily the PCG stopping criterion. For further discussion regarding suggested stopping criteria and accuracy monitoring, the reader is referred to Arioli (2004), Amir et al (2010) and Amir



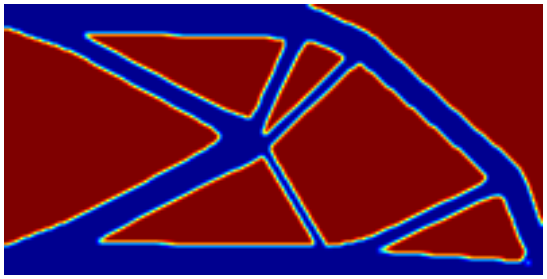
(a) Problem setting



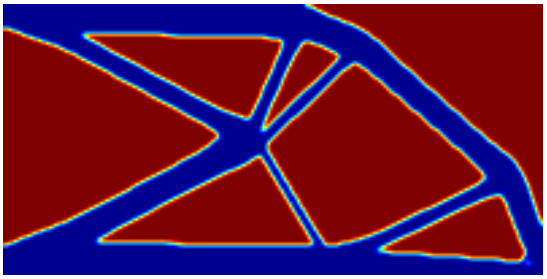
(b) Minimum compliance, standard analysis



(c) Minimum compliance, approximate reanalysis

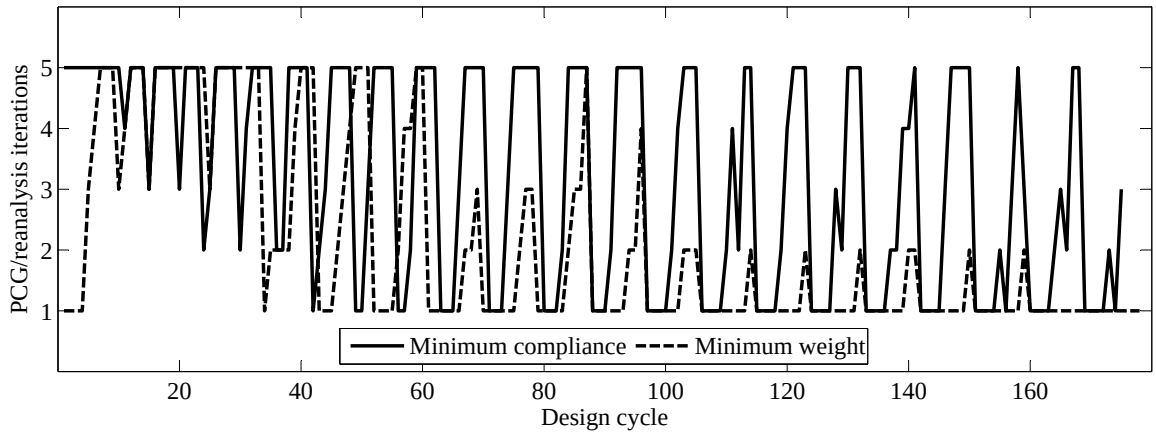


(d) Minimum volume, standard analysis

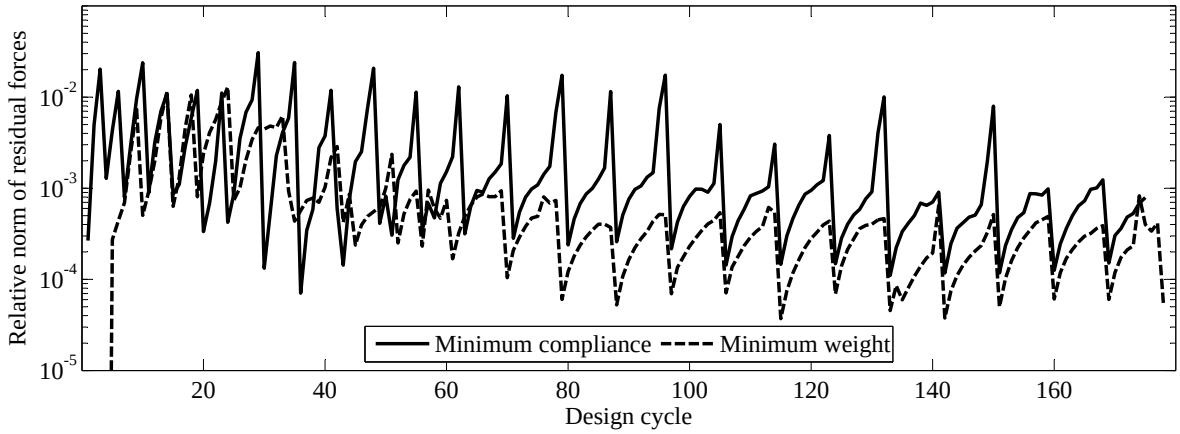


(e) Minimum volume, approximate reanalysis

Figure 1: Topology optimization of a cantilever beam with two point loads. A fixed number of 200 design iterations is performed. Objective values and layouts obtained by approximate procedures (**c,e**) are practically identical to those obtained by standard procedures (**b,d**), but the minimum volume reanalysis approach promises more significant savings in computational cost.



(a)

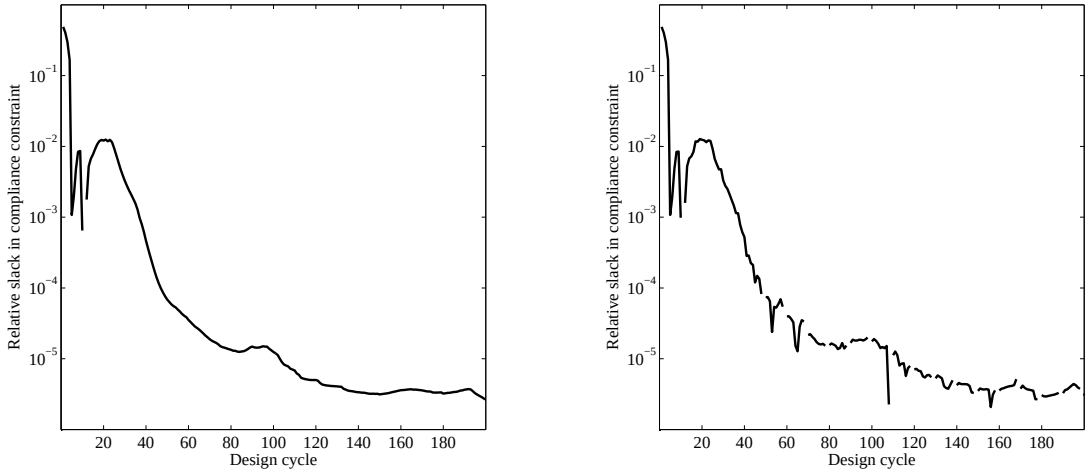


(b)

Figure 2: Data from the optimization of a cantilever beam with two point loads, using both minimum compliance and minimum volume formulations. **a** Number of PCG / reanalysis iterations; **b** Approximation errors in terms of residual forces. Design cycles in which standard direct solves were performed were omitted. The minimum volume reanalysis approach requires much fewer PCG iterations (in total, 369 compared to 565) and provides more accurate approximations in terms of residual forces. Note that a maximum number of 5 PCG iterations was imposed, and the stopping criterion was set to a relative error of 10^{-3}.

et al (2013). Some of the ideas suggested in this context will be examined in Section 5. The second observation is that when reanalysis is utilized, the minimum volume formulation requires fewer matrix decompositions and fewer PCG iterations in comparison to the minimum compliance formulation (see also Figure 2(a)). Furthermore, the accuracy of the approximations is improved, as can be seen in Figure 2(b). This is a direct result of the *stiff preconditioning* concept: In a minimum compliance optimization process, the design is in general stiffened as optimization progresses. Thus reanalysis is typically performed with the factors of $\mathbf{K}(\bar{\rho}_{k-l})$ which is less stiff than $\mathbf{K}(\bar{\rho}_k)$. The opposite happens in a minimum volume optimization process, particularly in the early design cycles: Material is gradually removed so reanalysis is typically performed with the factors of $\mathbf{K}(\bar{\rho}_{k-l})$ which is stiffer than $\mathbf{K}(\bar{\rho}_k)$. In Figure 3 the slackness in the compliance constraint is plotted for both accurate and approximate minimum volume procedures. Clearly the slackness tends to decrease as optimization progresses, meaning that stiffness is gradually reduced towards the target value.

The efficiency of the minimum volume procedure is reflected also in MATLAB run time: 70.7 seconds compared to 79.8 seconds. The difference is expected to increase in larger 2-D as well as in 3-D problems. For this relatively small 2-D problem, reanalysis is not expected to be faster than standard procedures, especially not in MATLAB where a very efficient sparse Cholesky factorization is implemented (Chen et al, 2008). Nevertheless we see that the minimum volume reanalysis-based code is competitive even for this test case and more significant savings are reported in the following, as well as in Section 5.



(a) (b)

Figure 3: The slackness in the compliance constraint plotted versus design cycles. **a** Standard minimum volume optimization; **b** Reanalysis-based minimum volume optimization. Clearly the slackness tends to decrease as optimization progresses, meaning that stiffness is gradually reduced towards the target value. Reanalysis is likely to involve stiff preconditioning thus its accuracy and efficiency are improved compared to reanalysis-based minimum compliance procedures. Missing values correspond to slight violations of the constraint.

4.3 Increased efficiency by matrix-free reanalysis and extension to 3-D

In the implementation of the reanalysis procedure presented above, the stiffness matrix was assembled during every design cycle - whether for the purpose of factorization or for performing matrix-vector products within PCG. In some cases it can be beneficial to avoid assembly during reanalysis design cycles, hence matrix-vector products are performed in a *matrix-free* manner. The difference in computational time compared to an assembly-based procedure depends on various factors such as: The properties of the stiffness matrix; the number of PCG iterations; the computational platform; and programming proficiency. When the number of PCG iterations is rather limited, the matrix-free implementation may perform better. Even for the small test case utilized above, running time measured in MATLAB was further reduced by avoiding matrix assembly in reanalysis-based design cycles: The minimum volume result reported in the bottom of Table 1 was reproduced in only 64 seconds, thus faster than both the assembly-based implementation as well as the minimum compliance 88-line code. Further comparisons of running times are reported in Section 5.

Matrix factorizations and direct solvers are in general not suitable for effectively tackling three-dimensional problems. Iterative equation solvers, primarily the family of Krylov subspace solvers, are then employed (see for example Saad

(2003) for an introduction to the topic). Krylov subspace solvers are often chosen also in the context of 3-D topology optimization (Borrvall and Petersson, 2001; Wang et al, 2007; Evgrafov et al, 2008; Aage and Lazarov, 2013). Among the various choices of iterative solvers, multigrid-preconditioned procedures appear to be very efficient due to the optimal computational performance of multigrid methods (e.g Trottenberg et al, 2001).

For iteratively solving equation systems arising from linear-elastic modeling, PCG is most suitable as it exploits both the symmetry and the positive-definiteness of the stiffness matrix. The combination of CG with multigrid preconditioning has been originally explored by Kettler (1982) and by Braess (1986). MGCG has shown to exhibit superior performance compared to traditional PCG which typically utilizes incomplete factorizations (Tatebe, 1993; Tatebe and Oyanagi, 1994; Ashby and Falgout, 1996). In the context of topology optimization, an MGCG procedure for minimum compliance topology optimization was recently suggested (Amir et al, 2013). It was shown that the number of MGCG iterations remains constant with mesh refinement, even for high-contrast layouts of solid and void. This leads to better computational performance than that of preconditioning with an incomplete Cholesky factorization. The same MGCG framework is utilized herein also for the 3-D extension of the proposed reanalysis-based minimum volume procedure.

Straightforward utilization of the MGCG framework in the current context is as follows. During "accurate" design cycles in 3-D optimization, a fully accurate MGCG solve is performed instead of a Cholesky decomposition in 2-D optimization. In subsequent "reanalyzed" design cycles, the coarse grid operators are reused for preconditioning the equation systems hence a recycled preconditioning scheme is performed. Therefore both matrix assembly and generation of the multigrid components are avoided. Further reduction in computational effort is possible by relaxing the conditions for MGCG convergence within "accurate" design cycles, meaning that also in such cycles we obtain an approximation $\widetilde{\mathbf{u}}$ rather than exact displacements $\mathbf{u}$. Two alternative stopping criteria for MGCG, related to the accuracy of the design sensitivities, are imposed as suggested recently (Amir et al, 2013). These are repeated here for the sake of completeness. Criterion (I) involves direct monitoring of all the gradients of the compliance functional given in Eq. (1) in the form

$$\frac{\left|\widetilde{\mathbf{u}}_i^{\mathsf{T}} \frac{\partial \mathbf{K}}{\partial \bar{\rho}_e} \widetilde{\mathbf{u}}_i - \widetilde{\mathbf{u}}_{i-1}^{\mathsf{T}} \frac{\partial \mathbf{K}}{\partial \bar{\rho}_e} \widetilde{\mathbf{u}}_{i-1}\right|}{\left|\widetilde{\mathbf{u}}_i^{\mathsf{T}} \frac{\partial \mathbf{K}}{\partial \bar{\rho}_e} \widetilde{\mathbf{u}}_i\right|} < \eta \qquad \forall e \tag{5}$$

where i and $i-1$ represent two consecutive MGCG iterations within a certain design cycle; and the parameter η controls the accuracy of the computed design sensitivities, implicitly controlling the residual forces as well. Criterion (II) relies on the argument that when the optimization problem is far from a candidate solution (a KKT point), a strictly accurate solution of the analysis equations may not be necessary. For intermediate densities the KKT conditions imply a constant strain energy density

$$\frac{\partial f_v}{\partial \rho_e} + \Lambda \frac{\partial g_c}{\partial \rho_e} = 0. \tag{6}$$

In case the residual corresponding to Eq. (6) is large, it makes sense to relax the requirement for highly accurate evaluation of the sensitivities $\frac{\partial g_c}{\partial \rho_e}$. Therefore criterion (II) has the form

$$\left|\widetilde{\mathbf{u}}_i^{\mathsf{T}} \frac{\partial \mathbf{K}}{\partial \bar{\rho}_e} \widetilde{\mathbf{u}}_i - \widetilde{\mathbf{u}}_{i-1}^{\mathsf{T}} \frac{\partial \mathbf{K}}{\partial \bar{\rho}_e} \widetilde{\mathbf{u}}_{i-1}\right| < \left|\frac{\partial f_v}{\partial \rho_e} + \Lambda \frac{\partial g_c}{\partial \rho_e}\right| \qquad \forall e \,|\, 0 < \rho_e < 1 \tag{7}$$

where the right-hand-side term is based on information from the previous design cycle.

Such a minimum volume recycled-preconditioning MGCG procedure will be demonstrated and benchmarked versus the minimum compliance MGCG procedure in the following examples.

5 Examples

In this section we compare various computational procedures with the aim to demonstrate the benefits of the minimum volume recycled preconditioning approach. Significant computational savings are achieved by combining the approach proposed in Section 4 with stopping criteria of MGCG which involve direct control over the accuracy of the design sensitivities (Amir et al, 2013). The following numerical strategies are utilized and compared:

- [MinC-ACC] Standard minimum compliance using accurate MGCG analysis;
- [MinV-ACC] Standard minimum volume using accurate MGCG analysis;
- [MinC-SM] Minimum compliance using approximate MGCG analysis with sensitivity monitoring;
- [MinV-SM] Minimum volume using approximate MGCG analysis with sensitivity monitoring;

- [MinC-REs] Minimum compliance using recycled preconditioning with s MGCG iterations and accurate MGCG solves when force residual exceeds a certain tolerance;
- [MinV-REs] Minimum volume using recycled preconditioning with s MGCG iterations and accurate MGCG solves when force residual exceeds a certain tolerance;
- [MinV-MFREs] Minimum volume using matrix-free recycled preconditioning with s MGCG iterations and accurate MGCG solves when force residual exceeds a certain tolerance;
- [MinV-SM-MFREs] Minimum volume using matrix-free recycled preconditioning with s MGCG iterations and approximate MGCG analysis with sensitivity monitoring when force residual exceeds a certain tolerance.

In order to further demonstrate the attractiveness of the recycled preconditioning approach, and more generally the MGCG-based solver, a comparison to another existing 3-D topology optimization code is conducted. For this purpose we use Liu and Tovar's code (Liu and Tovar, 2013), with both a sparse Cholesky factorization (as in the publicly available code) and a PCG solver with an incomplete Cholesky factorization (zero fill-in). These will be labeled in the following [MinC-Top3D-SS] and [MinC-Top3D-ICPCG], respectively.

5.1 3-D cantilever

In this example the proposed minimum volume procedure with recycled preconditioning is benchmarked versus a recent result obtained by MGCG-based minimum compliance optimization. The design problem is of a three-dimensional cantilever beam with a sine-shaped load at the bottom of the free edge, see Figure 4(a) for the setup. The sine function has a value of zero at the two corners and 1 at the midpoint. The FE mesh consists of $48\times24\times24$ unity cubes, thus the number of elements is $N = 27,648$ and the number of DOF is 91,875; the material properties are $E_{min} = 10^{-9}, E_{max} = 1, \nu = 0.3$; the modified SIMP penalty is 3.0; and sensitivity filtering is applied with a radius of $r = \sqrt{3}$. Four multigrid levels are utilized, meaning a direct solve is performed on a $6\times3\times3$ grid. Within the multigrid V-cycle, a single damped Jacobi smoothening cycle is performed with $\omega = 0.6$. All experiments involve 50 design cycles with a move limit of 0.2 imposed on the optimality criteria updates. Within procedures involving recycled preconditioning, accurate or sensitivity-monitored analysis was performed in case the recent approximation resulted in a relative residual norm exceeding 10^{-2}. With an available volume fraction of 0.12, Amir et al (2013) reported an optimized compliance of $3,330$. The layout they obtained, essentially by the procedure labeled [MinC-ACC], is displayed in Figure 4(b).

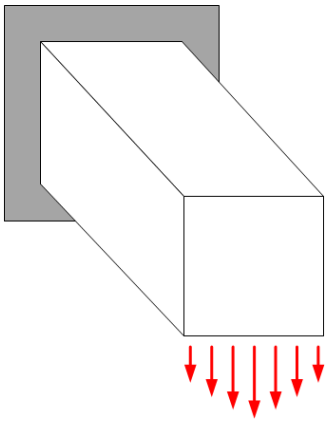
The results obtained with various computational procedures are summarized in Table 2. Most noteworthy is that minimum volume matrix-free recycled preconditioning procedures [MinV-MFRE5] and [MinV-SM-MFRE5] were the fastest for this example problem, requiring less than half the time of the standard minimum compliance implementation. This comes with no compromise on the accuracy of the optimization process: The same objective was achieved as with a standard minimum volume implementation, while satisfying the compliance constraint and in fact providing a slightly better compliance-to-volume trade-off than that achieved with the minimum compliance procedure. The layout obtained by [MinV-SM-MFRE5] is displayed in Figure 4(c). It differs from the layout in Figure 4(b), as also occurred with previous examples in Section 4: Minimum compliance and minimum volume procedures may lead to different layouts corresponding to distinct local minima solutions that share the same performance.

As expected, minimum compliance with recycled preconditioning fails due to divergence of the MGCG procedure within reanalyzed design cycles: The multigrid coarse operators correspond to a less stiff design thus the spectral radius of the preconditioned system implies inferior convergence properties and possible divergence of the multigrid V-cycle (see e.g. Notay, 2007). This can be overcome by either imposing a very restrictive move limit so that design changes are small, or by performing a very large number of MGCG iterations. Therefore it can be concluded that with MGCG as the equation solver, minimum compliance optimization with recycled preconditioning appears to be impractical. This is in complete contrast to minimum volume optimization with recycled preconditioning which is shown to be most effective. To complement the results in Table 2, we note that the number of recycled-preconditioning design cycles for procedures [MinV-RE5], [MinV-MFRE5] and [MinV-SM-MFRE5] was 30, 31 and 27 respectively. This means that with [MinV-RE5] and [MinV-MFRE5] the number of accurate MGCG solves performed was 20 and 19 respectively. With [MinV-SM-MFRE5] there were 23 assemblies followed by approximate MGCG solves, with a stopping criterion related to the accuracy of the design sensitivities. It can be seen that the number of accurate solves, and consequently MGCG iterations, performed when using [MinV-RE5] is slightly different than with [MinV-MFRE5]. This is because boundary conditions are enforced differently when matrix-vector products on the fine grid level are performed without assembly.

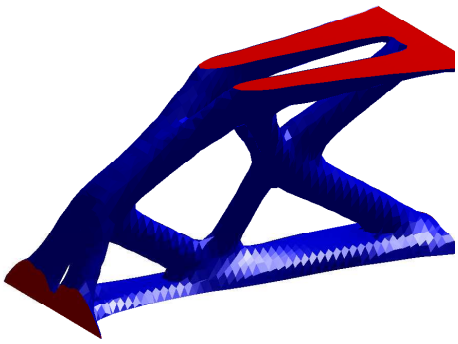
A close look at the top two rows of Table 2 reveals a somewhat surprising result: Even the standard minimum volume procedure is faster than minimum compliance - due to the fewer number of MGCG iterations. The number of MGCG iterations performed at each design cycle within four selected variants is plotted in Figure 5. Examining the number of MGCG iterations performed for achieving full accuracy, it can be seen that in the minimum compliance case this number rises sharply over the first 20 design cycles up to 147, then in the following 30 cycles it stabilizes at approximately 70. In

Table 2: Summary of results with various minimum compliance and minimum volume procedures, 50 optimization cycles of a $48 \times 24 \times 24$ cantilever. Minimum volume with recycled preconditioning runs more than twice faster than standard minimum compliance.
* In Liu and Tovar's code, the cubes appear to be of size $2 \times 2 \times 2$, resulting in a doubling of the stiffness.
** Node numbering was modified with respect to the original code for improving the performance.

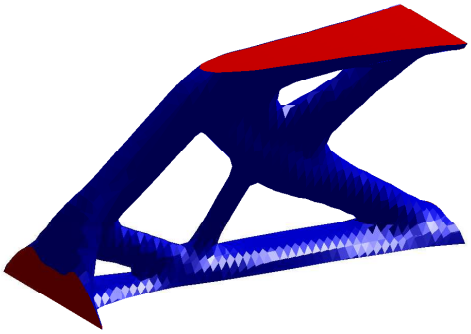
Procedure	Objective	Constraint	MGCG/PCG it.	MATLAB time	Sensitivity monitoring
MinC-ACC	$f_c = 3,330$	$V^\star = 0.120 \times N$	3,473	250.78	
MinV-ACC	$f_v = 0.114$	$c^\star = 3,330$	2,535	211.83	
MinC-SM	$f_c = 3,330$	$V^\star = 0.120 \times N$	867	166.48	Cr. (I) with $\eta = 10^{-1}$ and Cr. (II)
MinC-SM	$f_c = 3,330$	$V^\star = 0.120 \times N$	760	160.34	Cr. (I) with $\eta = 10^{0}$ and Cr. (II)
MinC-SM	$f_c = 3,330$	$V^\star = 0.120 \times N$	631	151.79	Cr. (I) with $\eta = 10^{2}$ and Cr. (II)
MinV-SM	$f_v = 0.114$	$c^\star = 3,330$	689	157.11	Cr. (I) with $\eta = 10^{-1}$ and Cr. (II)
MinV-SM	$f_v = 0.114$	$c^\star = 3,330$	657	152.64	Cr. (I) with $\eta = 10^{0}$ and Cr. (II)
MinV-SM	$f_v = 0.114$	$c^\star = 3,330$	626	151.59	Cr. (I) with $\eta = 10^{2}$ and Cr. (II)
MinC-RE5			optimization failed		
MinV-RE5	$f_v = 0.114$	$c^\star = 3,330$	1,254	159.79	
MinV-MFRE5	$f_v = 0.114$	$c^\star = 3,330$	1,170	123.43	
MinV-SM-MFRE5	$f_v = 0.114$	$c^\star = 3,330$	546	113.12	Cr. (I) with $\eta = 10^{-1}$ and Cr. (II)
MinV-SM-MFRE5	$f_v = 0.114$	$c^\star = 3,330$	502	110.37	Cr. (I) with $\eta = 10^{0}$ and Cr. (II)
MinV-SM-MFRE5	$f_v = 0.114$	$c^\star = 3,330$	502	113.77	Cr. (I) with $\eta = 10^{2}$ and Cr. (II)
MinC-Top3D-SS	$f_c = 1,665$*	$V^\star = 0.120 \times N$	—	1,347.50**	
MinC-Top3D-ICPCG	$f_c = 1,665$*	$V^\star = 0.120 \times N$	19,365**	601.14**	



(a) Problem setting



(b) Minimum compliance, accurate MGCG analysis [Mi ACC]



(c) Minimum volume, approximate MGCG with sensitivity monitoring and recycled preconditioning [MinV-SM-MFRE5]

Figure 4: Topology optimization of a 3-D cantilever beam. A standard minimum compliance procedure is compared to a minimum volume procedure with recycled preconditioning. The optimized layouts obtained after 50 design iterations differ but their structural performance is practically equivalent. The minimum volume approximate procedure runs more than twice faster: 114.7 compared to 250.8 seconds in MATLAB on a single processor.

the minimum volume case however, the number of MGCG iterations rises moderately over the first 20 design cycles up to 63, then in the following 30 cycles it declines slowly to 50. This inherent advantage of the minimum volume approach may be related to specific characteristics of MGCG or perhaps to more general consequences of employing preconditioned Krylov methods. In any case, further research is necessary in order to fully comprehend this computational behavior. The efficiency of approximate minimum volume procedures is also demonstrated in Figure 5: The number of MGCG iterations performed within [MinV-MFRE5] and [MinV-SM-MFRE5] (the data refers to $\eta = 10^0$) is significantly smaller than within [MinV-ACC].

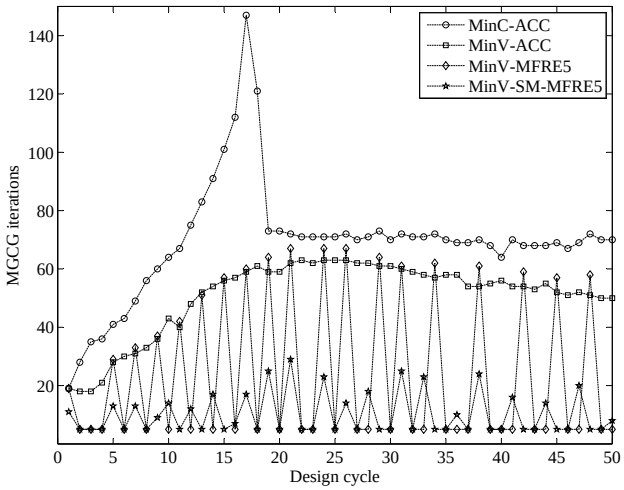


Figure 5: The number of MGCG iterations performed at each design cycle of the 3-D cantilever example. The standard minimum volume procedure requires fewer MGCG iterations than the minimum compliance procedure - a total of 2,535 compared to 3,473. Minimum volume procedures with recycled preconditioning offer further reductions: 1,170 and 502 MGCG iterations, the latter corresponding to a sensitivity-monitored procedure.

Finally, in order to examine the efficiency of the proposed approach on a finer grid, the cantilever example is re-run on a $80\times40\times40$ mesh. The physical filter size is kept constant, i.e. $r = \sqrt{3} * 80/48$, so that the same physical layout should be generated. The results obtained with several computational variants are presented in Table 3. With all variants, the number of MGCG iterations is similar to the number of iterations performed on the coarser example. This is expected with multigrid procedures that typically exhibit mesh-independent convergence. It can be seen that on the $80\times40\times40$ mesh the number of MGCG iterations is in fact slightly smaller than on the $48\times24\times24$ mesh - this is because the filter size was enlarged while the number of grid levels remained constant. Therefore fewer steps are required for the coarse-grid solution to propagate into the filtered domain, leading to faster MGCG convergence (Amir et al, 2013). In overall, the minimum volume procedures with recycled preconditioning are again the most efficient, thus indicating the potential also for large-scale applications.

Table 3: Summary of results with accurate minimum compliance and various minimum volume procedures, 50 optimization cycles of a $80 \times 40 \times 40$ cantilever. Minimum volume with recycled preconditioning runs more than twice faster than standard minimum compliance.
* In Liu and Tovar's code, the cubes appear to be of size $2\times2\times2$, resulting in a doubling of the stiffness.
** Node numbering was modified with respect to the original code for improving the performance.

Procedure	Objective	Constraint	MGCG/PCG it.	MATLAB time	Sensitivity monitoring
MinC-ACC	$f_c = 5,562$	$V^\star = 0.120 \times N$	2,535	1,053.31	
MinV-ACC	$f_v = 0.114$	$c^\star = 5,562$	2,005	896.05	
MinV-RE5	$f_v = 0.114$	$c^\star = 5,562$	956	685.08	
MinV-MFRE5	$f_v = 0.114$	$c^\star = 5,562$	900	544.91	
MinV-SM-MFRE5	$f_v = 0.114$	$c^\star = 5,562$	440	484.71	Cr. (I) with $\eta = 10^2$ and Cr. (II)
MinC-Top3D-ICPCG	$f_c = 2,780$*	$V^\star = 0.120 \times N$	34,174**	4,974.98**	

5.2 3-D bridge

In this example the conceptual design of a three-dimensional bridge structure is presented. The bridge is subject to a uniform vertical unit load on the carriageway and fixed supports are available at the bottom corners of the design domain, see Figure 6(a) for the setup. Exploiting double symmetry, one quarter of the design domain is discretized using a mesh of 64×16×48 unity cubes, thus the number of elements is $N = 49,152$ and the number of DOF is 162,435. The carriageway is represented by two layers of prescribed solid elements which are excluded from the optimization. The width of the carriageway is half of the total size of the design domain in the Y-direction, i.e. occupying 8 elements. Hence the number of elements active in the optimization is $N_{active} = 48,128$. The length of each of the support regions in the X-direction is a rounding of 1/12 of the total length, i.e. in the double-symmetric quarter the supports span over a region of 11×16 elements. As in the previous example, the modified SIMP penalty is 3.0; the material properties are $E_{min} = 10^{-9}, E_{max} = 1, \nu = 0.3$; and sensitivity filtering is applied with a radius of $r = \sqrt{3}$. Four multigrid levels are utilized, meaning a direct solve is performed on a 8×2×6 grid. Within the multigrid V-cycle, a single damped Jacobi smoothening cycle is performed with $\omega = 0.6$. All experiments involve 50 design cycles with a move limit of 0.2 imposed on the optimality criteria updates.

First a standard minimum compliance optimization is performed based on accurate MGCG analysis. With an available volume fraction of 0.1 a compliance of $4.326 \cdot 10^5$ is obtained. With the above compliance imposed as a constraint, standard volume minimization leads to a volume fraction of 0.0969. For demonstrating the efficiency of the proposed approach, the same problem setting was solved also by two recycled-preconditioning minimum volume procedures, namely [MV-MFRE5] and [MV-SM-MFRE5]. Accurate or sensitivity-monitored MGCG solves were performed in case the recent approximation resulted in a relative residual norm exceeding 10^{-2}. Results of the four experiments are presented in Table 4. Again, the procedure labeled [MV-SM-MFRE5] required roughly half the run time needed for the standard minimum compliance approach. Monitoring of the sensitivities was based exclusively on criterion (II) of Eq. (7) by setting $\eta = 10^2$ in criterion (I) of Eq. (5). Throughout 50 design cycles, 22 cycles involved matrix assembly and an approximate MGCG solve with sensitivity monitoring; the other 28 cycles involved matrix-free recycled preconditioning. Significant savings are offered also by employing [MV-MFRE5] where again 22 accurate MGCG solves were performed and 28 design cycles were based on matrix-free recycled preconditioning. The number of MGCG iterations performed at each design cycle within the four procedures is plotted in Figure 7, highlighting the potential savings offered by the minimum volume approach with recycled preconditioning. The volume-compliance trade-offs achieved by all procedures are practically identical though the layouts are slightly different, see Figures 6(b) and 6(c) (only densities above 0.8 are displayed). This again demonstrates the variety of local minima solutions that provide similar performance.

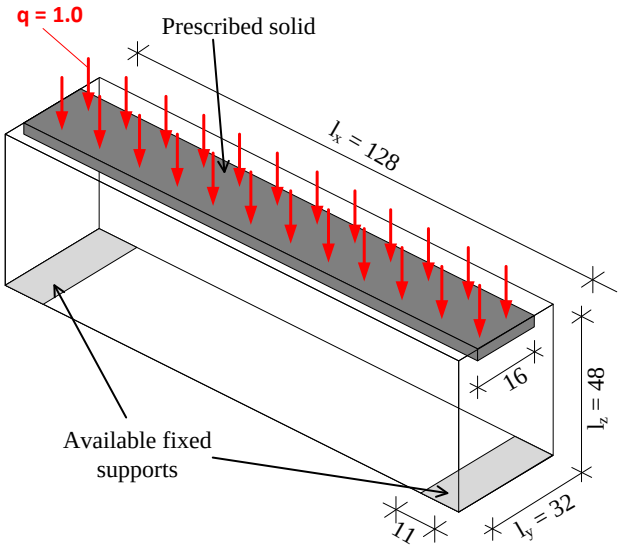
Table 4: Summary of results with various minimum compliance and minimum volume procedures, optimization of a $64 \times 16 \times 48$ bridge (double-symmetric quarter).
* In Liu and Tovar's code, the cubes appear to be of size 2×2×2, resulting in a doubling of the stiffness.
** Node numbering was modified with respect to the original code for improving the performance.

Procedure	Objective	Constraint	MGCG/PCG it.	MATLAB time
MinC-ACC	$f_c = 4.326 \cdot 10^5$	$V^\star = 0.1 \times N_{active}$	2,168	354.78
MinV-ACC	$f_v = 0.0969$	$c^\star = 4.326 \cdot 10^5$	1,791	320.13
MinV-MFRE5	$f_v = 0.0967$	$c^\star = 4.326 \cdot 10^5$	942	210.49
MinV-SM-MFRE5	$f_v = 0.0968$	$c^\star = 4.326 \cdot 10^5$	346	176.12
MinC-Top3D-ICPCG	$f_c = 2.140 \cdot 10^5$*	$V^\star = 0.1 \times N_{active}$	11,960**	784.21**

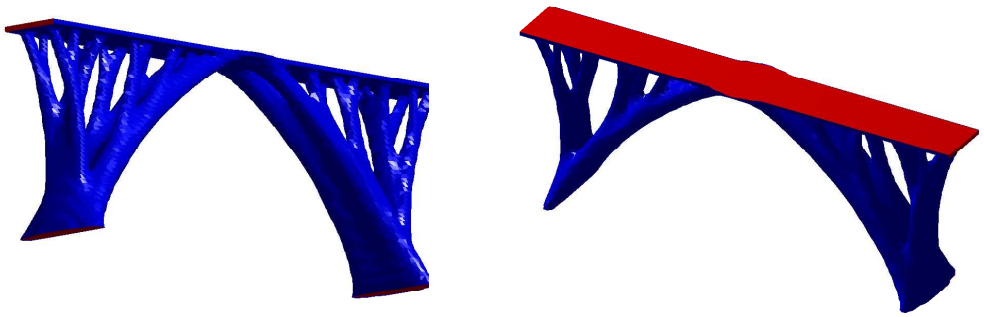
6 Conclusions

An efficient procedure for continuum structural topology optimization was presented. The main purpose is to provide an effective computational approach that can facilitate the integration of 3-D topology optimization into CAD software and mobile applications. The key point in achieving significant reduction in computational time is the exploitation of "stiff" preconditioning arising from reanalysis-based optimization. Reanalysis concepts applicable to 2-D problems are extended to 3-D in the form of recycled preconditioning within a general MGCG framework. It is shown that combining recycled preconditioning with a minimum volume problem formulation leads to a more efficient procedure than the popular minimum compliance approach.

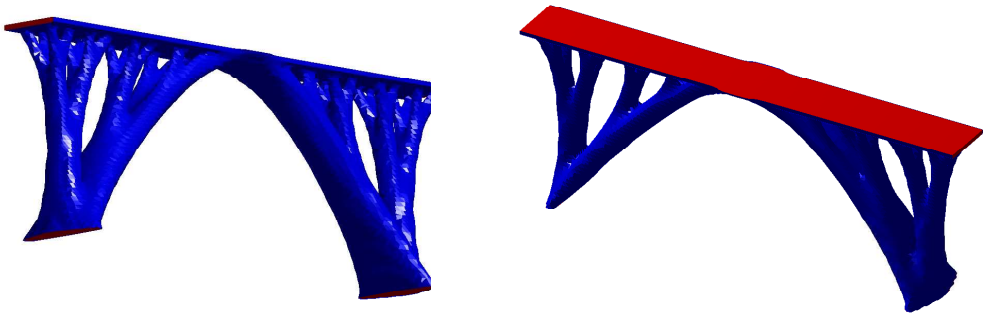
The proposed minimum volume procedure with recycled preconditioning was benchmarked in MATLAB versus standard procedures for both minimizing compliance and minimizing volume. Furthermore, it was favorably compared to the original integration of approximate reanalysis into minimum compliance topology optimization. Results of three-dimensional test cases clearly highlight the advantages of the proposed procedure: Run time in MATLAB was roughly twice faster than that of standard procedures, without any compromise on the quality of the result in terms of



(a) Problem setting



(b) Minimum compliance, standard MGCG analysis [MinC-ACC]



(c) Minimum volume, approximate MGCG with sensitivity monitoring and recycled preconditioning [MinV-SM-MFRE5]

Figure 6: Problem setting and optimized layouts obtained after 50 design iterations of the 3-D bridge example. The optimized layouts differ slightly but their structural performance is practically equivalent. The minimum volume approximate procedure runs twice faster: 176.12 compared to 354.78 seconds in MATLAB on a single processor.

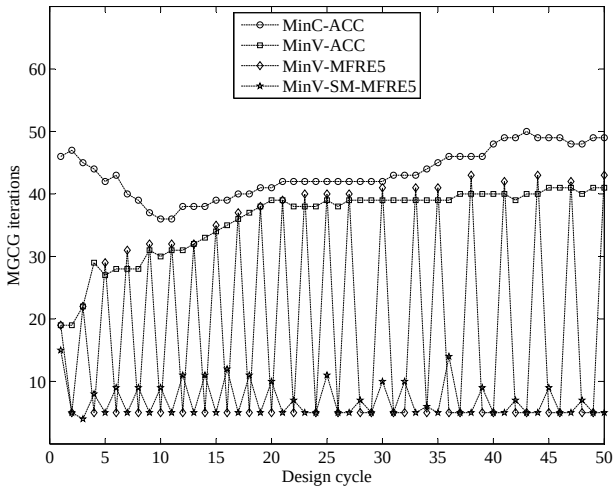


Figure 7: The number of MGCG iterations performed at each design cycle of the 3-D bridge example. The standard minimum volume procedure requires fewer MGCG iterations than the minimum compliance procedure - a total of 1,791 compared to 2,168. Minimum volume procedures with recycled preconditioning offer further reductions: 942 and 346 MGCG iterations, the latter corresponding to a sensitivity-monitored procedure.

the compliance-to-weight trade-off achieved. A comparison with an existing implementation shows that computational savings are also possible in 2-D, even though this is not of primary concern in the current study.

Future work will focus on gaining further insight regarding the inherent advantage of the minimum volume formulation when MGCG is utilized for solving the analysis equations. Another aspect to be examined is the applicability of the proposed approach, and more generally of the MGCG-based solvers, for topology optimization problems on unstructured grids. Investigating the performance in parallel computational environments will also be pursued. MATLAB codes complementing the article can be downloaded from the author's personal webpage `http://tx.technion.ac.il/~odedamir/`.

7 Acknowledgments

The author is grateful to the anonymous reviewers for many insightful comments and for numerous suggestions that helped improved the article. The author wishes to thank Niels Aage and Boyan S. Lazarov for fruitful discussions on related topics. Financial support received from the European Commission Research Executive Agency, grant agreement PCIG12-GA-2012-333647, is gratefully acknowledged.

References

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

Aage N, Nobel-Jørgensen M, Andreasen CS, Sigmund O (2013) Interactive topology optimization on hand-held devices. Structural and Multidisciplinary Optimization 47(1):1–6, URL `http://dx.doi.org/10.1007/s00158-012-0827-z`

Amir O, Sigmund O (2011) On reducing computational effort in topology optimization: how far can we go? Structural and Multidisciplinary Optimization 44:25–29

Amir O, Bendsøe MP, Sigmund O (2009) Approximate reanalysis in topology optimization. International Journal for Numerical Methods in Engineering 78:1474–1491

Amir O, Stolpe M, Sigmund O (2010) Efficient use of iterative solvers in nested topology optimization. Structural and Multidisciplinary Optimization 42:55–72

Amir O, Sigmund O, Schevenels M, Lazarov B (2012) Efficient reanalysis techniques for robust topology optimization. Computer Methods in Applied Mechanics and Engineering 245-246:217–231

Amir O, Aage N, Lazarov BS (2013) On multigrid-CG for efficient topology optimization. Structural and Multidisciplinary Optimization DOI 10.1007/s00158-013-1015-5, URL http://dx.doi.org/10.1007/s00158-013-1015-5, online

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in matlab using 88 lines of code. Structural and Multidisciplinary Optimization 43:1–16, URL http://link.springer.com/article/10.1007%2Fs00158-010-0594-7

Arioli M (2004) A stopping criterion for the conjugate gradient algorithm in a finite element method framework. Numerische Mathematik 97(1):1–24

Ashby SF, Falgout RD (1996) A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. Nuclear Science and Engineering 124:145–159

Bendsøe MP (1989) Optimal shape design as a material distribution problem. Structural Optimization 1:193–202

Bendsøe MP, Sigmund O (2003) Topology Optimization - Theory, Methods and Applications. Springer, Berlin

Bogomolny M (2010) Topology optimization for free vibrations using combined approximations. International Journal for Numerical Methods in Engineering 82(5):617–636, DOI 10.1002/nme.2778, URL http://dx.doi.org/10.1002/nme.2778

Borrvall T, Petersson J (2001) Large-scale topology optimization in 3d using parallel computing. Computer methods in applied mechanics and engineering 190(46):6201–6229

Bourdin B (2001) Filters in topology optimization. International Journal for Numerical Methods in Engineering 50:2143–2158

Braess D (1986) On the combination of the multigrid method and conjugate gradients. In: Hackbusch W, Trottenberg U (eds) Multigrid Methods II, Springer, pp 52–64

Bruns TE, Tortorelli DA (2001) Topology optimization of non-linear elastic structures and compliant mechanisms. Computer Methods in Applied Mechanics and Engineering 190:3443–3459

Chen Y, Davis TA, Hager WW, Rajamanickam S (2008) Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. ACM Transactions on Mathematical Software (TOMS) 35(3):22, URL http://dl.acm.org/citation.cfm?id=1391995

Davis TA (2006) Direct Methods for Sparse Linear System. SIAM

Evgrafov A, Rupp CJ, Maute K, Dunn ML (2008) Large-scale parallel topology optimization using a dual-primal substructuring solver. Structural and Multidisciplinary Optimization 36:329–345

Farhat C, Lesoinne M, LeTallec P, Pierson K, Rixen D (2001) FETI-DP: a dual–primal unified FETI method–part I: A faster alternative to the two-level FETI method. International journal for numerical methods in engineering 50(7):1523–1544

Fleury C, Braibant V (1986) Structural optimization - a new dual method using mixed variables. International Journal for Numerical Methods in Engineering 23(3):409–428

Kettler R (1982) Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. In: Multigrid methods, Springer, pp 502–534

Kim JE, Jang GW, Kim YY (2003) Adaptive multiscale wavelet-galerkin analysis for plane elasticity problems and its applications to multiscale topology design optimization. International journal of solids and structures 40(23):6473–6496

Kim TS, Kim JE, Kim YY (2004) Parallelized structural topology optimization for eigenvalue problems. International Journal of Solids and Structures 41:2623–2641

Kim YY, Yoon GH (2000) Multi-resolution multi-scale topology optimization-a new paradigm. International Journal of Solids and Structures 37(39):5529–5559

Kirsch U (1991) Reduced basis approximations of structural displacements for optimal design. AIAA journal 29:1751–1758

Kirsch U (2002) Design-Oriented Analysis of Structures. Kluwer Academic Publishers, Dordrecht

Kirsch U (2008) Reanalysis of Structures. Springer, Dordrecht

Kirsch U, Kočvara M, Zowe J (2002) Accurate reanalysis of structures by a preconditioned conjugate gradient method. International Journal for Numerical Methods in Engineering 55:233–251

Liu K, Tovar A (2013) URL `http://et.engr.iupui.edu/~tovara/top3d/`

Mahdavi A, Balaji R, Freckerand M, Mockensturm EM (2006) Topology optimization of 2D continua for minimum compliance using parallel computing. Structural and Multidisciplinary Optimization 32:121–132

MATLAB (2013) MATLAB version 8.1.0.604 (R2013a). The MathWorks, Natick, Massachusetts

Nguyen TH, Paulino GH, Song J, Le CH (2010) A computational paradigm for multiresolution topology optimization (mtop). Structural and Multidisciplinary Optimization 41:525–539, DOI 10.1007/s00158-009-0443-8, URL `http://dx.doi.org/10.1007/s00158-009-0443-8`

Nguyen TH, Paulino GH, Song J, Le CH (2012) Improving multiresolution topology optimization via multiple discretizations. International Journal for Numerical Methods in Engineering 92(6):507–530, DOI 10.1002/nme.4344, URL `http://dx.doi.org/10.1002/nme.4344`

Notay Y (2007) Convergence analysis of perturbed two-grid and multigrid methods. SIAM journal on numerical analysis 45(3):1035–1044, URL `http://epubs.siam.org/doi/abs/10.1137/060652312`

Rozvany GIN, Zhou M (1992) COC methods for a single global constraint. In: Rozvany GIN (ed) Shape and Layout Optimization of Structural Systems and Optimality Criteria Methods, Springer-Verlag

Saad Y (2003) Iterative Methods for Sparse Linear Systems, Second Edition. SIAM

Schmidt S, Schulz V (2011) A 2589 line topology optimization code written for the graphics card. Computing and Visualization in Science 14(6):249–256

Sigmund O (1997) On the design of compliant mechanisms using topology optimization. Mechanics Based Design of Structures and Machines 25:493–524

Sigmund O (2001) A 99 line topology optimization code written in matlab. Structural and Multidisciplinary Optimization 21:120–127

Sigmund O, Maute K (2012) Sensitivity filtering from a continuum mechanics perspective. Structural and Multidisciplinary Optimization 46:471–475, DOI 10.1007/s00158-012-0814-4, URL `http://link.springer.com/article/10.1007/s00158-012-0814-4`

Sigmund O, Torquato S (1997) Design of materials with extreme thermal expansion using a three-phase topology optimization method. Journal of the Mechanics and Physics of Solids 45(6):1037–1067

Stainko R (2006) An adaptive multilevel approach to the minimal compliance problem in topology optimization. Communications in Numerical Methods in Engineering 22(2):109–118, DOI 10.1002/cnm.800, URL `http://dx.doi.org/10.1002/cnm.800`

Suresh K (2013) Efficient generation of large-scale pareto-optimal topologies. Structural and Multidisciplinary Optimization 47(1):49–61

Svanberg K (1987) The method of moving asymptotes - a new method for structural optimization. International Journal for Numerical Methods in Engineering 24:359–373

Tatebe O (1993) The multigrid preconditioned conjugate gradient method. In: NASA CONFERENCE PUBLICATION

Tatebe O, Oyanagi Y (1994) Efficient implementation of the multigrid preconditioned conjugate gradient method on distributed memory machines. In: Supercomputing'94. Proceedings, IEEE, pp 194–203

Trottenberg U, Oosterlee C, Schuller A (2001) Multigrid. Academic Press

Vemaganti K, Lawrence EW (2005) Parallel methods for optimality criteria-based topology optimization. Computer Methods in Applied Mechanics and Engineering 194:3637–3667

Wadbro E, Berggren M (2009) Megapixel topology optimization on a graphics processing unit. SIAM review 51(4):707–721

Wang S, de Sturler E, Paulino GH (2007) Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. International Journal for Numerical Methods in Engineering 69:2441–2468

Zegard T, Paulino GH (2013) Toward GPU accelerated topology optimization on unstructured meshes. Structural and Multidisciplinary Optimization 48(3):473–485, DOI 10.1007/s00158-013-0920-y

Zhou M, Rozvany G (1991) The coc algorithm, part ii: topological, geometrical and generalized shape optimization. Computer Methods in Applied Mechanics and Engineering 89(1):309–336

Zuo W, Xu T, Zhang H, Xu T (2011) Fast structural optimization with frequency constraints by genetic algorithm using adaptive eigenvalue reanalysis methods. Structural and Multidisciplinary Optimization 43:799–810, DOI 10.1007/s00158-010-0610-y, URL http://dx.doi.org/10.1007/s00158-010-0610-y