INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2010; **00**:1–18
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

An algorithm for the topology optimization of geometrically nonlinear structures

Francisco A. M. Gomes[1*] and Thadeu A. Senne[2]

Departamento de Matemática Aplicada, IMECC, Universidade Estadual de Campinas, Campinas, SP, Brazil
Departamento de Matemática Aplicada, ICTE, Universidade Federal do Triângulo Mineiro, Uberaba, MG, Brazil

SUMMARY

Most papers on topology optimization consider that there is a linear relation between the strains and the displacements of the structure, meaning that the displacements of the structure are small. However, when the external loads applied to the structure are large, the displacements also become large, so it is necessary to suppose that there is a nonlinear relation between strains and displacements. In this case, we say that the structure is geometrically nonlinear. In practice, this means that the linear system that needs to be solved each time the objective function of the problem is evaluated is replaced by an ill-conditioned nonlinear system of equations. Moreover, the stiffness matrix and the derivatives of the problem also become harder to compute.
The objective of this work is to solve topology optimization problems under large displacements through a new optimization algorithm, named Sequential Piecewise Linear Programming (SPLP). This method rely on the solution of convex piecewise linear programming subproblems that include second order information about the objective function. To speed up the algorithm, these subproblems are converted into linear programming ones. The SPLP algorithm is not only globally convergent to stationary points, but our numerical experiments also show that it is efficient and robust. Copyright © 2010 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Topology optimization; Geometric nonlinearity; Sequential piecewise linear programming

1. INTRODUCTION

Efficient methods for the solution of topology optimization problems include the well known Method of Moving Asymptotes (MMA) and its variants (see, for example, Svanberg [1, 2], Bruyneel *et al.* [3], Zillober [4], Zillober, Schittkowski and Moritzen [5]), as well as some standard mathematical programming methods such as the Sequential Linear Programming (SLP) algorithm (Gomes and Senne [6], Sigmund [7], Kikuchi *et al.* [8], Nishiwaki *et al.* [9]) and the Sequential Quadratic Programming (SQP) algorithm (Etman et al. [10]). Even a hybrid scheme that uses quadratic approximations of some separable convex approximation functions (Groenwold and Etman [11]) has been successfully employed to solve such problems.

Most works on topology optimization consider that the structure is under small displacements, so the relation between strain and displacement can be represented by a linear model. However, for large displacements this hypothesis is no longer valid, and we say that the structure is geometrically nonlinear, which means that it is necessary to adopt a nonlinear representation for

*Correspondence to: Francisco A. M. Gomes, Departamento de Matemática Aplicada, IMECC, UNICAMP. Rua Sergio Buarque de Holanda, 651, CEP 13083-859, Campinas - SP, Brazil. E-mail: chico@ime.unicamp.br

Contract/grant sponsor: CAPES and FAPESP; contract/grant number: 2006/53768-0

the strain-displacement relation. Due to the computational difficulties that arise when the geometric nonlinearities are considered, there is just a few papers that deal with this problem. Among them, we may cite those written by Jog [12], Buhl, Pedersen and Sigmund [13], Bruns and Tortorelli [14], Gea and Luo [15], Bruns, Sigmund and Tortorelli [16], Ohsaki and Nishiwaki [17], Luo and Tong [18], Lazarov, Schevenels and Sigmund [19], Lahuerta *et al.* [20], and Lee and Park [21].

A survey on this field shows that MMA-like algorithms represent the vast majority of methods developed for solving geometrically nonlinear topology optimization problems. In this paper we introduce an alternative algorithm, called Sequential Piecewise Linear Programming (SPLP), that is an extension of the Sequential Linear Programming (SLP) algorithm of Gomes and Senne [6], developed for the design of structures under small displacements.

The SPLP algorithm adds second order information to the objective function, requiring the solution of one piecewise linear subproblem per iteration. This subproblem is further converted to a linear programming one, in order to reduce the computational costs. The idea is to stay between the SLP algorithm, that has cheap iterations but takes too many steps to converge, and the SQP method, that performs a smaller number of computationally intensive iterations.

The paper is organized as follows. In Section 2 we present the topology optimization problem under large displacements. The SPLP method is described in Section 3 and its practical implementation is presented in Section 4. The performance of the algorithm is analyzed in Section 5, and some conclusions are presented in Section 6.

2. THE GEOMETRICALLY NONLINEAR TOPOLOGY OPTIMIZATION PROBLEM

Let Ω be a given two- or three-dimensional domain where a structure is to be built. A topology optimization problem is a mathematical problem that consists in determining how to distribute material over Ω in such a way that the structure generated is optimal in some sense, and satisfies a set of design constraints.

In the minimum compliance topology optimization problem, the objective is to find the stiffest structure that fits into the domain, and satisfies a volume constraint and some boundary conditions. When the structure is subject to small displacements, a linear model is used to describe the strain-displacement relation. In this case, after the discretization of Ω, the topology optimization problem can be written in the form

$$\begin{array}{ll} \min\limits_{\boldsymbol{\rho}} & \mathbf{f}^T\mathbf{u} \\ \text{s.t.} & \mathbf{K}(\tilde{\boldsymbol{\rho}})\mathbf{u} = \mathbf{f} \\ & \displaystyle\sum_{i=1}^{n_{el}} v_i\tilde{\rho}_i \leq V^* \\ & \tilde{\rho}_i \in \{0,1\}, \qquad i = 1,\ldots,n_{el}, \end{array} \tag{1}$$

where n_{el} is the number of elements of the domain, $\tilde{\rho}_i$ is a variable that indicates whether the i-th element is filled ($\tilde{\rho}_i = 1$) or void ($\tilde{\rho}_i = 0$), v_i is the volume of the i-th element, V is the upper limit for the volume of the structure, $\mathbf{f}$ is the vector of nodal forces associated to the external loads, $\mathbf{u}$ is the vector of nodal displacements, and $\mathbf{K}(\tilde{\boldsymbol{\rho}})$ is the (global) stiffness matrix.

Since topology optimization problems are usually huge, it would not be possible to solve an integer nonlinear programming problem such as (1). To circumvent this problem, Bendsøe [22] introduced the *Solid Isotropic Material with Penalization* (SIMP) method, where the integer variables $\tilde{\rho}_i$ are replaced by the continuous densities $0 \leq \rho_i \leq 1$, that are raised to a penalty parameter $p > 1$ in order to avoid the presence of intermediate values. When the SIMP method is used, the global stiffness matrix is given by $\mathbf{K}(\boldsymbol{\rho}) = \sum_{i=1}^{n_{el}} \rho_i^p \mathbf{k}_i$, where $\mathbf{k}_i$ is the stiffness matrix of the i-th element.

If we also define a lower limit ρ_{min} for the densities ρ_i, then $\mathbf{K}(\boldsymbol{\rho})$ becomes symmetric and positive definite, so $\mathbf{u} = \mathbf{K}(\boldsymbol{\rho})^{-1}\mathbf{f}$, and we can rewrite Problem (1) as

DOI: 10.1002/nme

$$\begin{aligned} \min_{\boldsymbol{\rho}} \quad & \mathbf{f}^T\mathbf{K}(\boldsymbol{\rho})^{-1}\mathbf{f} \\ \text{s.t.} \quad & \sum_{i=1}^{n_{el}} \frac{v_i\rho_i}{V^*} - 1 \leq 0 \\ & 0 < \rho_{\min} \leq \rho_i \leq 1, \qquad i = 1, \ldots, n_{el}. \end{aligned} \tag{2}$$

When the structure is subject to large displacements, the linear strain-displacement model is no longer valid, and we need to replace the equilibrium equations $\mathbf{K}(\boldsymbol{\rho})\mathbf{u} = \mathbf{f}$ by a nonlinear system in the form

$$\mathbf{R}(\mathbf{u}, \boldsymbol{\rho}) = \mathbf{0}. \tag{3}$$

Supposing that $\boldsymbol{\rho}$ is fixed, the displacement vector $\mathbf{u}$ that solves (3) can be obtained by means of the Newton's method. In this case, the nonlinear equations are approximated by the linear system

$$\mathbf{K}_T(\mathbf{u}_k, \boldsymbol{\rho})\Delta\mathbf{u}_k = -\mathbf{R}(\mathbf{u}_k, \boldsymbol{\rho}), \tag{4}$$

where $\mathbf{K}_T = \partial\mathbf{R}/\partial\mathbf{u}$ is the tangent stiffness matrix. Starting from a given vector $\mathbf{u}_0$, the method solves a sequence of systems in the form (4), and updates the displacements using $\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta\mathbf{u}_k$ until the condition $\|\mathbf{R}(\mathbf{u}_k, \boldsymbol{\rho})\| \leq \varepsilon$ is satisfied, for some $\varepsilon > 0$.

The definition of $\mathbf{R}(\mathbf{u}, \boldsymbol{\rho})$ and of $\mathbf{K}_T(\mathbf{u}_k, \boldsymbol{\rho})$ varies according to the hyperelastic model adopted for the material. In this work, we use the neo-Hookean material model of Simo-Ciarlet [23, 24], instead of the usual Saint Venant-Kirchhoff model.

Although the Saint Venant-Kirchhoff material law is far more used than other material models, it does not guarantee the existence of a solution for (3). In fact, this model can pose several difficulties for Newton's method, preventing its convergence (Lahuerta *et al.* [20]). On the other hand, Ball [25] has shown that the system (3) always has a solution when a polyconvex constitutive model, such as the Simo-Ciarlet neo-Hoolean material law, is used.

Moreover, the tangent stiffness matrix of the Simo-Ciarlet model is cheaper to compute than the matrix related to the Saint Venant-Kirchhoff model, that is build up summing several terms that depend on $\boldsymbol{\rho}$. (see, for example, [26]).

To see how systems (3) and (4) are defined, let $\mathbf{F} = [f_{ij}]$ represent the deformation gradient tensor of the structure. In the Simo-Ciarlet model, the strain-energy density function is given by (see [20])

$$\widehat{W} \equiv \widehat{W}(J, I_C) = \frac{1}{2}\lambda\left[\frac{1}{2}(J^2 - 1) - \ln J\right] + \frac{1}{2}\mu(I_C - 3 - 2\ln J),$$

where λ and μ are Lamé constants, $J = \det(\mathbf{F})$ and $I_C = \text{tr}(\mathbf{F}^T\mathbf{F})$.

Gathering the components of the first Piola-Kirchoff stress tensor into the vector

$$\boldsymbol{\sigma} = \left[\frac{\partial\widehat{W}}{\partial f_{11}} \quad \frac{\partial\widehat{W}}{\partial f_{21}} \quad \frac{\partial\widehat{W}}{\partial f_{12}} \quad \frac{\partial\widehat{W}}{\partial f_{22}}\right]^T$$

and defining $\mathbf{G}$ as the matrix of the derivatives of the shape functions with respect to the displacements, we define

$$\mathbf{q}(\widehat{\mathbf{u}}^{(i)}) = \int_{\Omega_i} \mathbf{G}^T\boldsymbol{\sigma}\, d\Omega_i,$$

where Ω_i and $\widehat{\mathbf{u}}^{(i)}$ are, respectively, the domain and the vector of nodal displacements of the i-th element. Defining the vector of internal nodal forces of the structure as

$$\mathbf{f}_{int}(\mathbf{u}, \boldsymbol{\rho}) = \sum_{i=1}^{n_{el}} \rho_i^p \mathbf{q}(\widehat{\mathbf{u}}^{(i)})$$

we may write the nonlinear system (3) in the form

$$\mathbf{R}(\mathbf{u}, \boldsymbol{\rho}) = \mathbf{f}_{int}(\mathbf{u}, \boldsymbol{\rho}) - \mathbf{f} = \mathbf{0}.$$

DOI: 10.1002/nme

Using this system we can finally write the nonlinear version of the topology optimization (2) as

$$\begin{array}{ll} \min\limits_{\boldsymbol{\rho}} & [\mathbf{f}_{int}(\mathbf{u}(\boldsymbol{\rho}),\boldsymbol{\rho})]^T\mathbf{u}(\boldsymbol{\rho}) \\ \text{s.t.} & \sum\limits_{i=1}^{n_{el}} \dfrac{v_i\rho_i}{V^*} - 1 \leq 0 \\ & 0 < \rho_{\min} \leq \rho_i \leq 1, \qquad i = 1,\ldots,n_{el}, \end{array} \tag{5}$$

where we write $\mathbf{u} \equiv \mathbf{u}(\boldsymbol{\rho})$ to stress that the displacements depend on $\boldsymbol{\rho}$. The tangent stiffness matrix used in (4) is assembled combining the contributions of the n_{el} elements, i.e.

$$\mathbf{K}_T(\mathbf{u},\boldsymbol{\rho}) = \sum_{i=1}^{n_{el}} \rho_i^p \mathbf{k}_T(\widehat{\mathbf{u}}^{(e)}),$$

where $\mathbf{k}_T(\widehat{\mathbf{u}}^{(e)})$, the tangent matrix of the i-th element, is given by

$$\mathbf{k}_T(\widehat{\mathbf{u}}^{(e)}) = \int_{\Omega_e} \mathbf{G}^T\mathbf{D}\mathbf{G}d\Omega_e.$$

The tangent stiffness modulus matrix $\mathbf{D}$ used in this last formula is obtained deriving $\boldsymbol{\sigma}$ with respect to the components of the deformation gradient.

3. SEQUENTIAL PIECEWISE LINEAR PROGRAMMING

The sequential linear programming method proposed by Gomes and Senne [6] has shown a good performance for the solution of topology optimization problems under small displacements, mainly due to its quite cheap iterations.

In the presence of large displacements, however, it is not just the time spent per each iteration that counts. Since each computation of the objective function of problem (5) requires the solution of one nonlinear system, it is worth devising a method that reduces the number of iterations, as it may save time. But some care still must be taken to avoid spoiling the benefits of this time reduction by taking a computationally expensive iteration.

In this section we present an algorithm that is based on the solution of piecewise linear programming problems. The method is particularly well suited for solving difficult topology optimization problems, since it converges as quickly as a SQP algorithm, but its iterations require just the solution of linear programming problems, as in a SLP method.

3.1. Description of the method

Consider the nonlinear programming problem

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{c}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u, \end{array} \tag{6}$$

where the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{c} : \mathbb{R}^n \to \mathbb{R}^m$ have Lipschitz continuous first derivatives, and $\mathbf{x}_l, \mathbf{x}_u \in \mathbb{R}^n$ are vectors that define, respectively, the lower and upper bounds for $\mathbf{x}$. The conversion of a topology optimization problem such as (5) to the form (6) is an easy task, requiring just the introduction of a slack variable to transform the volume constraint into an equation.

A solution vector $\overline{\mathbf{x}} \in \mathbb{R}^n$ is called *feasible* if all of the constraints and bounds of (6) are satisfied at $\overline{\mathbf{x}}$. Since the bound constraints are never violated by the SPLP algorithm, the infeasibility of a point $\mathbf{x}$ is measured by the function

$$\varphi(\mathbf{x}) = ||\mathbf{c}(\mathbf{x})||_1.$$

We say that $\mathbf{x}$ is a *φ-stationary* point if it satisfies the Karush-Kuhn-Tucker (KKT) conditions of problem

$$\begin{array}{ll} \min & \varphi(\mathbf{x}) \\ \text{s.t.} & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u, \end{array}$$

DOI: 10.1002/nme

We also say that a feasible point $\mathbf{x}$ is *regular* if the gradients of the active constraints at $\mathbf{x}$ are linearly independent.

For most topology optimization problems, the second derivatives of the objective function are expensive to evaluate, so algorithms that require this kind of information are seldom used. With the aim of obtaining a good approximation for $f(\mathbf{x})$ (i.e. using some second order information) without sacrificing the efficiency of the optimization algorithm, in the SPLP method the objective function of (6) around a point $\mathbf{x}$ is approximated by

$$f(\mathbf{x}+\mathbf{s}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T\mathbf{s} + \Gamma_k(\mathbf{s}), \tag{7}$$

where $\Gamma_k : \mathbb{R}^n \to \mathbb{R}$ is a convex non-negative piecewise linear function that gives some information about the curvature of f. The definition of Γ_k will be discussed in detail in Subsection 4.1.

On their turn, the equality constraints of problem (6) are approximated by the linear model

$$\mathbf{c}(\mathbf{x}+\mathbf{s}) \approx \mathbf{c}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{s}, \tag{8}$$

where $\mathbf{A}(\mathbf{x}) = [\nabla c_1(\mathbf{x}) \dots \nabla c_m(\mathbf{x})]^T$ is the Jacobian matrix of the constraints.

Using (7) and (8), problem (6) can be approximated by

$$\begin{array}{ll} \min & m_k(\mathbf{s}) = \nabla f(\mathbf{x}^{(k)})^T\mathbf{s} + \Gamma_k(\mathbf{s}) \\ \text{s.t.} & \mathbf{A}(\mathbf{x}^{(k)})\mathbf{s} + \mathbf{c}(\mathbf{x}^{(k)}) = \mathbf{0} \\ & \mathbf{x}_l \le \mathbf{x}^{(k)} + \mathbf{s} \le \mathbf{x}_u \\ & \|\mathbf{s}\|_\infty \le \delta_k \end{array} \tag{9}$$

where $\delta_k > 0$. The last inequality in (9) defines a trust region, adopted to prevent the subproblem from becoming unbounded and to ensure the global convergence of the SPLP algorithm. In practice, the box constraints and the trust region inequality are coupled together, so we simply write $\mathbf{s}_l \le \mathbf{s} \le \mathbf{s}_u$ where

$$\mathbf{s}_l = \max\{-\delta, \mathbf{x}_l - \mathbf{x}^{(k)}\} \qquad \text{and} \qquad \mathbf{s}_u = \min\{\delta, \mathbf{x}_u - \mathbf{x}^{(k)}\}. \tag{10}$$

The central idea of the algorithm is simple: at the k-th iteration of the algorithm, we solve the piecewise linear problem (9), and use its solution, $\mathbf{s}_c$, to obtain a trial point $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}_c$ that is a (supposedly) better approximate solution for the original problem (6).

However, this scheme does no prevent the feasible set of problem (9) from being empty. When this happens, a feasibility restoration step must precede the computation of $\mathbf{s}_c$. Following the suggestion of Gomes and Senne [6], this restoration step is obtained solving the auxiliar subproblem

$$\begin{array}{ll} \min & M(\mathbf{x}^{(k)}, \mathbf{s}) = ||\mathbf{A}(\mathbf{x}^{(k)})\mathbf{s} + \mathbf{c}(\mathbf{x}^{(k)})||_1 \\ \text{s.t.} & \mathbf{x}_l \le \mathbf{x}^{(k)} + \mathbf{s} \le \mathbf{x}_u \\ & \|\mathbf{s}\|_\infty \le 0,8\delta_k. \end{array} \tag{11}$$

where, $M(\mathbf{x}^{(k)}, \mathbf{s})$ is the first order approximation for $\varphi(\mathbf{x})$ at $\mathbf{x}^{(k)}$.

The one-norm is used in (11) to allow the conversion of this problem into the linear programming problem

$$\begin{array}{ll} \min & \overline{M}(\mathbf{x}^{(k)}, \mathbf{s}, \mathbf{z}) = \mathbf{e}^T\mathbf{z} \\ \text{s.t.} & \mathbf{A}(\mathbf{x}^{(k)})\mathbf{s} + \mathbf{E}(\mathbf{x}^{(k)})\mathbf{z} = -\mathbf{c}(\mathbf{x}^{(k)}) \\ & \max\{-0.8\delta_k,\ \mathbf{x}_l - \mathbf{x}^{(k)}\} \le \mathbf{s} \le \min\{0.8\delta_k,\ \mathbf{x}_u - \mathbf{x}^{(k)}\} \\ & \mathbf{z} \ge \mathbf{0}, \end{array} \tag{12}$$

where $\mathbf{z} \in \mathbb{R}^{m_I}$ is the vector of slack variables of the m_I infeasible linearized constraints, $\mathbf{e} = [1\,1\,\dots\,1]^T$, and matrix $\mathbf{E}$ is defined by

$$\mathbf{E}_j(\mathbf{x}^{(k)}) = \begin{cases} \mathbf{I}_{i_j}, & \text{if } c_{i_j}(\mathbf{x}^{(k)}) < 0, \\ -\mathbf{I}_{i_j}, & \text{if } c_{i_j}(\mathbf{x}^{(k)}) > 0, \end{cases}$$

DOI: 10.1002/nme

where $\mathbf{I}_i$ is the i-th column of the identity matrix of order m_I, and $i_1, i_2, \ldots, i_{m_I}$ are the indices of the nonzero components of $\mathbf{c}(\mathbf{x}^{(k)})$. A feasible point for (12) can be trivially obtained by taking $\mathbf{s} = \mathbf{0}$ and $z_j = |c_{i_j}(\mathbf{x}^{(k)})|$, $j = 1, \ldots, m_I$.

We start an iteration of the SPLP algorithm finding $\mathbf{s}_n$, the solution of problem (12). If $M(\mathbf{x}^{(k)}, \mathbf{s}_n) = 0$, we also solve problem (9), obtaining the step $\mathbf{s}_c$. On the other hand, if $M(\mathbf{x}^{(k)}, \mathbf{s}_n) > 0$, we simply set $\mathbf{s}_c = \mathbf{s}_n$. One should notice that the trust region radius used in (11) is smaller than δ_k, in order to increase the feasible region of problem (9) and allow the reduction of the objective function.

A trial step $\mathbf{s}_c$ is only accepted if $\mathbf{x}^{(k)} + \mathbf{s}_c$ is a better solution for (6) than $\mathbf{x}^{(k)}$. To take this decision we rely on the merit function

$$\psi(\mathbf{x}, \theta) = \theta f(\mathbf{x}) + (1 - \theta)\varphi(\mathbf{x}),$$

where $\theta \in [0, 1]$ is a parameter that defines the balance between the two objectives of the method, that are the reduction of the objective function $f(\mathbf{x})$ and of the infeasibility measure $\varphi(\mathbf{x})$. Following the scheme presented in [27, 6], we define $\theta_k = \min\{\theta_k^{sup}, \theta_k^{large}\}$, where

$$\theta_k^{min} = \min\{1, \theta_0, \ldots, \theta_{k-1}\}, \qquad \theta_k^{large} = \left[1 + \frac{N}{(k+1)^{1.1}}\right] \theta_k^{min},$$

$$\theta_k^{sup} = \sup\{\theta \in [0, 1] \,|\, P_{red} \geq 0.5 P_{red}^{fsb}\},$$

and $N \geq 0$ is a large parameter (e.g. $N = 10^6$) used to allow a nonmonotone decrease of θ.

To decide if the step $\mathbf{s}_c$ is to be accepted or not, we compare the actual reduction of the merit function with the reduction predicted by the model (9). The actual reduction of ψ between $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k)} + \mathbf{s}_c$ is given by

$$A_{red} = \theta[f(\mathbf{x}) - f(\mathbf{x} + \mathbf{s}_c)] + (1 - \theta)[\varphi(\mathbf{x}) - \varphi(\mathbf{x} + \mathbf{s}_c)],$$

while the predicted reduction is defined as

$$P_{red} = \theta[-\nabla f(\mathbf{x})^T \mathbf{s}_c - \Gamma_k(\mathbf{s}_c)] + (1 - \theta)[M(\mathbf{x}, \mathbf{0}) - M(\mathbf{x}, \mathbf{s}_c)],$$

The step $\mathbf{s}_c$ is accepted if the merit function is reduced at least by one tenth of the reduction predicted by the linear model, i.e. if $A_{red} \geq 0.1 P_{red}$. If this condition is not verified, δ is reduced and the step is recomputed. Besides, the trust region radius may also be increased or decreased depending on the ratio A_{red}/P_{red}. These and other details of the method are fully described in Algorithm 1.

3.2. *Global convergence*

In [6], Gomes and Senne present a SLP algorithm that is globally convergent under the hypothesis that the sequence $\{\mathbf{x}^{(k)}\}$ is bounded, which is trivially satisfied by bound constrained problems such as (5). The global convergence of Algorithm 1 can be easily derived from the lemmas and theorems presented in [6], rewriting Lemmas 3.3 and 3.7 to cope with the piecewise linear function $\Gamma_k(\mathbf{s})$.

The convergence proofs are divided into three steps. First, it is shown that the algorithm is well defined, i.e., that a new iterate $\mathbf{x}^{(k+1)}$ is eventually obtained after repeating the steps of the algorithm a finite number of times. Then it is proved that every limit point of an infinite sequence $\{\mathbf{x}^{(k)}\}$ is φ-stationary. Finally, it is shown that there exists a limit point $\mathbf{x}^*$ that is stationary for (6) whenever the limit points of the sequence $\{\mathbf{x}^{(k)}\}$ are feasible and regular. The reader is referred to [28] for the complete convergence proofs.

4. IMPLEMENTATION DETAILS

4.1. *The piecewise linear model*

The piecewise linear function $\Gamma_k(\mathbf{s})$ that is used to construct the model $m_k(\mathbf{s})$ must approximate the quadratic function $\gamma_k(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T \mathbf{B}_k \mathbf{s}$, where $\mathbf{B}_k$ is a symmetric positive definite matrix. Since such model is rebuilt at each iteration, we will drop off the subscript k to simplify the notation.

DOI: 10.1002/nme

Algorithm 1 The SPLP algorithm.

Require: $\delta_{\min} > 0$, $\delta_0 \geq \delta_{\min}$ and $\mathbf{x}^{(0)}$ such that $\mathbf{x}_l \leq \mathbf{x}^{(0)} \leq \mathbf{x}_u$
$\theta_0 \leftarrow 1$; $\theta^{\max} \leftarrow 1$; $k \leftarrow 0$
while a stopping criterion is not satisfied, **do**
  Determine $\mathbf{s}_n$, the solution of (12)
  if $\bar{M}(\mathbf{x}^{(k)}, \mathbf{s}_n, \mathbf{z}) = 0$, **then**
    Starting from $\mathbf{s}_n$, determine $\mathbf{s}_c$, the solution of (9)
  else
    $\mathbf{s}_c \leftarrow \mathbf{s}_n$.
  end if
  Compute θ_k^{large} and θ_k^{sup}
  $\theta_k \leftarrow \min\{\theta_k^{large}, \theta_k^{sup}, \theta^{max}\}$
  if $A_{red} \geq 0.1 P_{red}$ **then**
    $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \mathbf{s}_c$
    Compute $\mathbf{A}(\mathbf{x}^{(k+1)})$, $\mathbf{B}_{k+1}$, $\mathbf{E}(\mathbf{x}^{(k+1)})$ and $\nabla f(\mathbf{x}^{(k+1)})$.
    if $A_{red} \geq 0.5 P_{red}$, **then**
      $\delta_{k+1} \leftarrow \min\{1.5\delta_k, \|\mathbf{x}_u - \mathbf{x}_l\|_\infty\}$
    else if $A_{red} \geq 0.2 P_{red}$, **then**
      $\delta_{k+1} \leftarrow \delta_k$
    else
      $\delta_{k+1} \leftarrow 0.25\delta_k$
    end if
    $\delta_{k+1} \leftarrow \max\{\delta_{k+1}, \delta_{min}\}$
    $\theta_{max} \leftarrow 1$
    $k \leftarrow k + 1$
  else
    $\delta_k \leftarrow \max\{0.25\|\mathbf{s}_c\|_\infty, 0.1\delta_k\}$
    $\theta_{max} \leftarrow \theta_k$
  end if
end while

Supposing that $\mathbf{B}$ is diagonal, both $\gamma(\mathbf{s})$ and $\Gamma(\mathbf{s})$ are separable, so we can write

$$\Gamma(\mathbf{s}) = \sum_{i=1}^{n} \Gamma_i(s_i) \;\approx\; \gamma(s) = \sum_{i=1}^{n} \gamma_i(s_i) = \sum_{i=1}^{n} \frac{1}{2} b_i s_i^2,$$

where b_i is the i-th diagonal element of $\mathbf{B}$. We define each function $\Gamma_i(s_i)$ following the guidelines proposed by Byrd *et al.* [29, 30].

Let $\Gamma_i(s_i)$ be formed by $2r+1$ line segments, each one defined by a linear function $\ell_i^{(j)}(s_i)$, $j = 0, \ldots, 2r$. In this case, since $\mathbf{B}$ is positive definite, we may write

$$\Gamma_i(s_i) = \max_{j \in \{0,\ldots,2r\}} \left\{ \ell_i^{(j)}(s_i) \right\}.$$

Each $\ell_i^{(j)}(s_i)$ should interpolate $\gamma_i(s_i)$ and its first derivative at a point $t_i^{(j)}$. The definition of the interpolation points depends on two scalars, D_i and U_i, and on the lower and upper limits for $\mathbf{s}$, given in (10). Three cases are considered.

Case 1. If $|x_i - x_{l_i}| \leq 10^{-6}$, then we define $t_i^{(0)} = D_i = 0$, choose $U_i > 0$ and set

$$t_i^{(j)} = 0.3^{2r-j} U_i, \qquad j = 1, \ldots, 2r.$$

Case 2. If $|x_i - x_{u_i}| \leq 10^{-6}$, then we define $t_i^{(2r)} = U_i = 0$, choose $D_i < 0$ and set

$$t_i^{(j)} = 0.3^{j} D_i, \qquad j = 0, \ldots, 2r-1.$$

DOI: 10.1002/nme

Case 3. Otherwise, we choose $D_i < 0$ and $U_i > 0$, and set

$$t_i^{(j)} = \begin{cases} 0.3^j D_i, & j = 0, \ldots, r-1, \\ 0, & j = r, \\ 0.3^{2r-j} U_i, & j = r+1, \ldots, 2r. \end{cases}$$

Instead of adopting the elegant but complicated choice of D_i and U_i given in [30], we simply define

$$D_i = \max\{s_{l_i}, \min\{-0.5\delta, 2v_i\}\} \qquad \text{and} \qquad U_i = \min\{s_{u_i}, \max\{0.5\delta, 2v_i\}\},$$

where $v_i = -(\partial f(\mathbf{x})/\partial x_i)/b_i$ is the unconstrained minimum of the quadratic function

$$q(s_i) = \frac{1}{2} b_i s_i^2 + \frac{\partial f(\mathbf{x})}{\partial x_i} s_i.$$

Since $\Gamma(s)$ is a convex function, we can convert (9) into a linear programming with $n(2r+1)$ variables. This is done by means of a change of variables defined by

$$s_i = \sum_{j=0}^{2r} \omega_{ij}, \qquad i = 1, \ldots, n, \tag{13}$$

where the new variable ω_{ij} is associate to the j-th line segment of $\Gamma_i(s_i)$. The upper and lower bounds for these new variables are given by

$$\begin{array}{ll} s_{l_i} \le \omega_{i0} \le \beta_i^{(0)}, & \\ 0 \le \omega_j \le \beta_i^{(j)} - \beta_i^{(j-1)}, & j = 1, \ldots, 2r-1, \\ 0 \le \omega_{2r} \le s_{u_i} - \beta_i^{(2r)}, & \end{array}$$

where $\beta_i^{(j)} = \frac{1}{2}(t_i^{(j)} + t_i^{(j+1)})$ is the breakpoint between the j-th and the $(j+1)$-th adjacent segments of $\Gamma_i(s_i)$.

Preliminary numerical experiments suggest that it is sufficient to use three interpolation points (i.e. $r = 1$), since no significant improvement on the number of iterations is obtained for five points ($r = 2$), and the linear programming problems becomes prohibitive large for $r > 2$.

4.2. Choice of the diagonal matrix

In principle, a good choice for matrix $\mathbf{B}_k$ would be the diagonal of the Hessian of the objective function (or of the Lagrangian) of problem (6), evaluated at $\mathbf{x}^{(k)}$. However, for topology optimization problems, even this matrix is too expensive to obtain. Moreover, in the presence of large displacements, there is no guarantee that the Hessian is positive definite.

If we put aside the idea of using the true Hessian, the first alternative that comes to mind is the adoption of a limited-memory quasi-Newton approximation for $\mathbf{B}_k$, as the one proposed by Goulart and Herskovits [31].

Another interesting option is to approximate the Hessian by a diagonal matrix defined upon intermediate variables. In this case, given a set of n intermediate variables $y_i \equiv y_i(\mathbf{x})$, the function $f(\mathbf{x})$ is approximated by

$$\widehat{f}(\mathbf{y}) = f(\mathbf{y}^{(k)}) + \sum_{i=1}^{n} \frac{\partial f(\mathbf{y}^{(k)})}{\partial y_i}(y_i - y_i^{(k)}), \tag{14}$$

where $y_i = 1/x_i$ or $y_i = x_i^{a_i}$, depending on if we use the reciprocal intermediate variables introduced by Etman, Groenwold and Rooda [10], or the exponential variables of Groenwold and Etman [11]. In this kind of approximation, $\widehat{f}(\mathbf{y})$ is a linear function on $\mathbf{y}$, but a nonlinear function of the original variables $\mathbf{x}$, so its Hessian may be used to define $\mathbf{B}_k$.

DOI: 10.1002/nme

In our preliminary tests, the use of exponential intermediate variables outperformed both the reciprocal variables scheme and the quasi-Newton approximation. Therefore, we decided to use this approach, although we recognize that further experiments should be conducted in order to establish the best alternative for computing $\mathbf{B}_k$.

For the exponential variables, the approximate function (14) is given by

$$\widehat{f}_E(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \sum_{i=1}^{n} \left[\left(\frac{x_i}{x_i^{(k)}} \right)^{a_i^{(k)}} - 1 \right] \left(\frac{x_i^{(k)}}{a_i^{(k)}} \right) \frac{\partial f(\mathbf{x}^{(k)})}{\partial x_i},$$

where, acoording to Fadel, Riley and Barthelemy [32], the exponents $a_i^{(k)}$ should be obtained imposing the condition

$$\frac{\partial \widehat{f}_E(\mathbf{x}^{(k-1)})}{\partial x_i} = \frac{\partial f(\mathbf{x}^{(k-1)})}{\partial x_i},$$

which gives

$$a_i^{(k)} = 1 + \ln \left(\frac{\partial f(\mathbf{x}^{(k-1)})}{\partial x_i} \Big/ \frac{\partial f(\mathbf{x}^{(k)})}{\partial x_i} \right) \Big/ \ln \left(x_i^{(k-1)} / x_i^{(k)} \right).$$

Naturally, some precautions should be taken when the terms inside a logarithm is negative or zero. In such cases, we adopt $a_i^{(k)} = -1$, which reverts to the reciprocal approximation scheme.

4.3. Filtering

A naïve implementation of a topology optimization algorithm that uses the SIMP method in combination with 4-node rectangular finite elements and bilinear interpolating functions may result in a structure containing a checkerboard-like pattern (Díaz and Sigmund [33]).

To circumvent this problem, it is a common practice to adopt a filter that replaces the density of each element e_i by a value that depends on the densities of the elements that belong to a ball $B(i, r)$ with radius r_{max}, centered in e_i.

Two filters are available in the SPLP algorithm: the density filter of Bruns and Tortorelli [34], and the Heaviside filter proposed by Guest, Prevost and Belytschko [35], and adapted by Sigmund [36].

When the density filter is used, ρ_i is replaced by a weighted mean of the densities of the elements belonging to a neighborhood B_i. This weighted density is given by

$$\phi_i \equiv \phi_i(\boldsymbol{\rho}) = \sum_{j \in B_i} \frac{\xi_j(s_{ij})}{\sum_{j \in B_i} \xi_j(s_{ij})} \rho_j,$$

where

$$\xi_j(s_{ij}) = \begin{cases} \frac{\exp(-s_{ij}^2/2(r/3)^2)}{2\pi(r/3)}, & \text{if } s_{ij} \le r, \\ 0, & \text{if } s_{ij} > r, \end{cases}$$

and s_{ij} is the Euclidean distance between the centroids of elements i and j.

Although the filtered densities must be used not only in the objective function, but also in the constraints, this filter preserves the linearity of the volume constraint.

The idea behind the Heaviside filter is to round up each weighted density ϕ_i to 1 if $\phi_i > \rho_{\min}$, so ϕ_i can assume just two discrete values, $\rho_{\min}$ or 1. However, with this modification, the optimization problem becomes non-differentiable. This difficulty is circumvented approximating ϕ_i by

$$\eta_i(\rho) = 1 - \exp(-\beta\phi_i) + \phi_i \exp(-\beta),$$

where $\beta \ge 0$ is a penalty parameter that controls the curvature of this function.

It should be noted that not only $\eta_i(\rho)$ is differentiable, but it also reduces to the density filter when $\beta = 0$. To avoid numerical instabilities, Sigmund [36] suggests the gradual increase of β from 1 to 500. However, in our experiments, best results were obtained adopting the fixed value $\beta = 1.5$. The only side effect of this approach is that the volume constraint becomes nonlinear.

DOI: 10.1002/nme

4.4. Stopping criteria

One important feature of our algorithm is the fact that the stopping criteria is not based only on the step size, or on the variation of the objective function value. Instead, we adopt a mathematically sound criterion, aiming to ensure that the solution obtained is close to a stationary point for problem (6).

Let $P_X(y)$ be the orthogonal projection of y onto the set $X = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u\}$, $\nabla \ell(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)})$ be the gradient of the Lagrangian and $\boldsymbol{\lambda}$ denote the vector of the Lagrange multipliers associated to the equality constraints of (6) (given as a by product of the tangent step). We say that Algorithm 1 has found an good approximation for a stationary point when

$$\|g_P(\mathbf{x}^{(k)})\|_\infty < 10^{-3},$$

where $g_P(\mathbf{x}^{(k)}) = P_X(\mathbf{x}^{(k)} - \nabla \ell(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)})) - \mathbf{x}^{(k)}$ is the projected gradient at $\mathbf{x}^{(k)}$. In addition to this criterion, we also limit to 50 the number of iterations of the algorithm for the intermediate values of the penalty parameter of the SIMP method. For the last value of ρ, the limit on the number of iterations is increased to 10000, so only the projected gradient is used in practice.

4.5. Solution of the nonlinear systems

Each time the objective function of problem (5) is evaluated, it is necessary to solve the nonlinear system $\mathbf{R}(\mathbf{u}(\boldsymbol{\rho}), \boldsymbol{\rho}) = \mathbf{0}$. In our algorithm, this is done applying Newton's method until $\|\mathbf{R}(\mathbf{u}_k, \boldsymbol{\rho})\| \leq 10^{-6}$.

The only drawback of this approach is that there is no guarantee that the tangent stiffness matrix $\mathbf{K}_T(\boldsymbol{\rho})$ that appears in the linear system $\mathbf{K}_T(\mathbf{u}_k, \boldsymbol{\rho})\Delta\mathbf{u}_k = -\mathbf{R}(\mathbf{u}_k, \boldsymbol{\rho})$ is positive definite, so Newton's method may fail to converge to a solution. A strategy for circumventing this problem was proposed by Buhl, Pedersen and Sigmund [13], and consists in the removal of nodes surrounded by void (or minimum density) elements from the convergence criterion.

Another strategy is the arc-length method, proposed by Wempner [37] and improved by Riks [38], Batoz and Dhatt [39] and Crisfield [40, 26]. To understand how this method works, it is important to remember that, if the nodal displacements of a structure are large, the load-deflection curve is not only nonlinear, but also may contain limit points, i.e. points where the curve attain a local maximum, a local minimum or where the tangent is vertical. When Newton's method reaches a limit point, $\mathbf{K}_T(\boldsymbol{\rho})$ becomes singular, and the method diverges. In the arc-length method, the load is gradually increased by means of the introduction of a "load level" variable, that belongs to the $(0, 1]$ interval. An additional constraint is used to fix the length of the load step.

In our numerical experiments, both the elimination of elements surrounded by void and the arc-length method fail to ensure the convergence of Newton's method for large values of the external load.

Fortunately, we also noted that the numerical instabilities of Newton's method were related to the density threshold ρ_{min}. When this value was small, $\mathbf{K}_T(\boldsymbol{\rho})$ became near singular, and the method failed to converge. Therefore we decided to perform a simple change of variables. We rescaled the densities so their original lower limit, $\rho_{min} = 0.001$, was converted into the more palatable value $\overline{\rho}_{min} = 0.1$. The scaled density vector $\boldsymbol{\tau}$ was defined by

$$\tau_i = \left(\frac{1 - \overline{\rho}_{\min}}{1 - \rho_{min}}\right)\rho_i + \left(\frac{\overline{\rho}_{\min} - \rho_{\min}}{1 - \rho_{\min}}\right), \qquad i = 1, \ldots, n_{el}.$$

Naturally, the volume constraint was altered accordingly, so V^* was changed to

$$\widetilde{V} = \left(\frac{1 - \overline{\rho}_{\min}}{1 - \rho_{min}}\right)V^* + \left(\frac{\overline{\rho}_{\min} - \rho_{\min}}{1 - \rho_{\min}}\right)\sum_{i=1}^{n_{el}} v_i.$$

DOI: 10.1002/nme

5. NUMERICAL TESTS

Our analysis of the practical performance of the SPLP algorithm was based on the solution of five well known topology problems, to make sure that the structures obtained are compatible with the results presented in the literature. All of the problems were discretized into 4-node rectangular finite elements, using bilinear interpolating functions to approximate the displacements.

Since the SLP method of Gomes and Senne outperformed the CCSA algorithm of Svanberg [2] for the linear compliance minimization problems (see [6]), we decided to compare the new algorithm with the SLP method, as well as to a SQP algorithm with a diagonal approximation to the Hessian matrix. The main difference between the three methods tested is that the objective function of problem (9) is replaced by $\nabla f(\mathbf{x}^{(k)})^T\mathbf{s}$ for the SLP algorithm, and by $\nabla f(\mathbf{x}^{(k)})^T\mathbf{s} + \frac{1}{2}\mathbf{s}^T\mathbf{B}_k\mathbf{s}$ for the SQP algorithm, where $\mathbf{B}_k$ is described in Subsection 4.2. In practical terms, this means that the SQP algorithm requires the solution of quadratic programming problems, while the other two methods involve the solution of linear programming problems. On the other hand, the SPLP method deals with larger problems, since it replaces vector $\mathbf{s} \in \mathbb{R}^n$ by $\boldsymbol{\omega} \in \mathbb{R}^{3n}$, as described in (13).

All of the algorithms were coded in C++. The subproblems (9) and (12) were solved using the CPLEX (ver. 12.1) software library, no matter the objective function adopted. The solution of the linear systems that appear at each iteration of Newton's method was obtained using the Cholesky factorization routine of the CHOLMOD library (ver. 1.7), developed by Davis and his colleagues[41]. The tests were performed on a personal computer with a Intel Core i7-3612QM processor, under the Ubuntu Linux operating system.

For all of the problems, the initial density vector was chosen so that the elements had the same density, and the volume of the structure was equal to V^*. The penalty parameter of the SIMP method, p, was gradually increased from 1 to 3, in steps of 0.1. Each time this parameter was changed, the objective function and the constraints were rescaled by

$$\widetilde{f}(\boldsymbol{\tau}) = \frac{f(\boldsymbol{\tau})}{\|\nabla f(\boldsymbol{\tau}_0)\|_\infty} \qquad \text{and} \qquad \widetilde{c}(\boldsymbol{\tau}) = \frac{c(\boldsymbol{\tau})}{\|\nabla c(\boldsymbol{\tau}_0)\|_\infty},$$

where $\boldsymbol{\tau}_0$ is the starting scaled density vector for that value of p. Other initial parameters used in the algorithm were $\delta_0 = 0.1$, and $\delta_{min} = 10^{-6}$.

5.1. Problem 1

The design domain of this problem, presented by Buhl, Pedersen and Sigmund [13], is shown in Figure 1. The structure has a thickness of $0.1\,m$. A load of $300\,kN$ is applied downwards at the center of the right side of the domain. The Young's modulus and the Poisson's ratio of the material are set to $3 \times 10^9\ N/m^2$ and 0.4, respectively. The domain is discretized into 2500 square elements. The optimal structure must contain no more than 50% of the domain's volume. The radius r_{max} adopted for both the density and the Heaviside filters are set to the length of 3.5 elements.

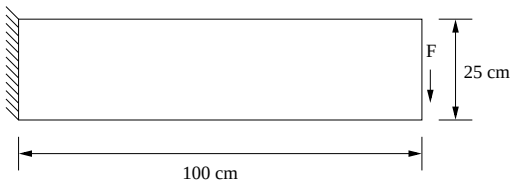

Figure 1. Design domain for the first problem.

The results obtained for Problem 1 are shown on Table I. The table presents the final objective function value, the number of iterations and the time spent by each algorithm for both the density and the Heaviside filters.

The structures found by the SPLP algorithm are presented in Figure 2. To show how the nonlinear analysis affects the topology of the structure, this figure also includes the results obtained using the small displacement model. The structures found are compatible with those presented in [13].

DOI: 10.1002/nme

Table I. Results for the first problem.

Method	Density filter			Heaviside filter		
	Objective	Iter.	Time (s)	Objective	Iter.	Time (s)
SPLP	1.0863×10^5	1102	135.54	1.1312×10^5	1063	127.94
SLP	1.0801×10^5	1640	175.10	1.1310×10^5	1835	193.99
SQP	1.0863×10^5	1001	160.34	1.1304×10^5	933	148.74

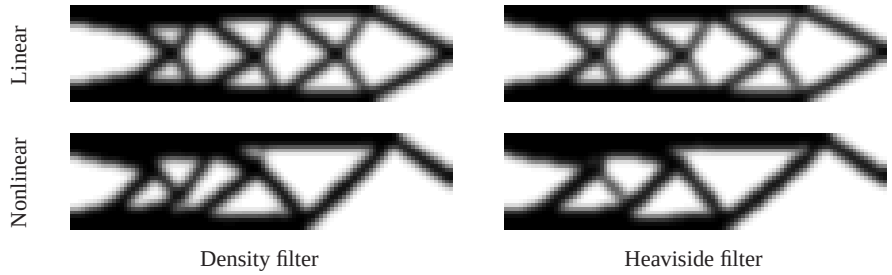

Figure 2. Structures obtained for the first problem.

5.2. *Problem 2*

This problem was proposed by Gea and Luo in [15]. The design domain is shown in Figure 3. The structure's thickness is set to $0.1\,cm$. A load of $200\,N$ is applied upwards at the center of the top side of the domain. The Young's modulus and the Poisson's ratio of the material are, respectively, $10^5\,N/cm^2$ and 0.3. The domain is discretized into 3600 square elements. The volume of the optimal structure is limited to 25% of the domain's volume. A radius of 2.5 elements is used for the filters.

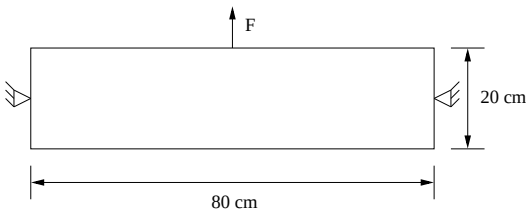

Figure 3. Design domain for the second problem.

Table II contains the results obtained for this problem. The structures found by the SPLP algorithm are shown on Figure 4. It is worth noting that, for this problem, there is a huge difference between the structures obtained considering the linear and the nonlinear model. Moreover, the topology is also significantly affected by the filter used.

Table II. Results for the second problem.

Method	Density filter			Heaviside filter		
	Objective	Iter.	Time (s)	Objective	Iter.	Time (s)
SPLP	$325,14$	741	111.06	428.08	2318	336.66
SLP	$325,50$	961	122.63	428.08	3980	478.73
SQP	$325,33$	810	172.71	428.08	2451	522.34

DOI: 10.1002/nme

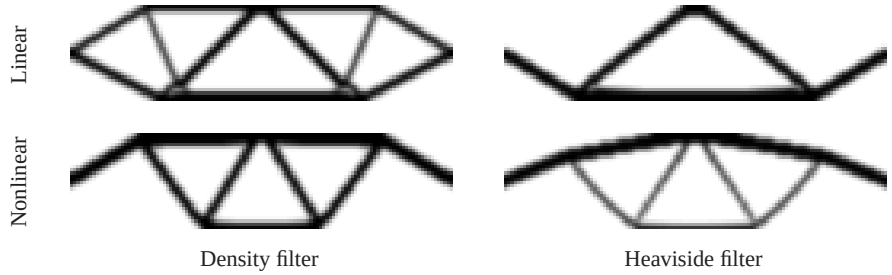

Figure 4. Structures obtained for the second problem.

5.3. *Problem 3*

Figure 5 shows the design domain for this problem, that was suggested by Gea and Luo [15]. The thickness of the domain is set to $0.1\,cm$. The magnitudes of the external loads are $F_1 = F_3 = 15\,N$ and $F_2 = 30\,N$. The material has a Young's modulus of $10000\,N/cm^2$ and a Poisson's ratio of 0.3. The domain is discretized into 2500 square elements. The maximum admissible volume is set to 25% of the domain's volume. A radius of 2 elements is used for the filters.

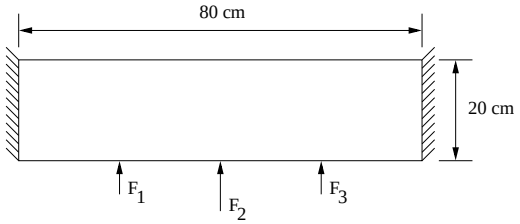

Figure 5. Design domain for the third problem.

Table III shows the performance of the three algorithms for this problem. The optimal topologies are given in Figure 6.

Table III. Results for the third problem.

Method	Density filter			Heaviside filter		
	Objective	Iter.	Time (s)	Objective	Iter.	Time (s)
SPLP	88.644	900	90.40	121.95	1262	119.14
SLP	88.695	3625	295.65	121.83	2693	178.68
SQP	88.644	887	121.40	121.95	1391	180.59

5.4. *Problem 4*

The fourth problem, proposed by Jung and Gea [42], is shown in Figure 7. The thickness of the structure is set to $0.1\,cm$ and the magnitude of the external load is set to $30\,N$. The material has a Young's modulus of $3000\,N/cm^2$ and a Poisson's ratio of 0.3. The domain is discretized into 3200 square elements. The final structure must contain only 20% of the domain's volume. The radius adopted for the filters corresponds to the length of 2 elements.

The results obtained for this example are presented in Table IV. The optimal topologies are shown in Figure 8. One should notice the difference between the structures obtained for the small and the large displacement models. The structures obtained using the density filter are compatible to those presented in [42].

DOI: 10.1002/nme

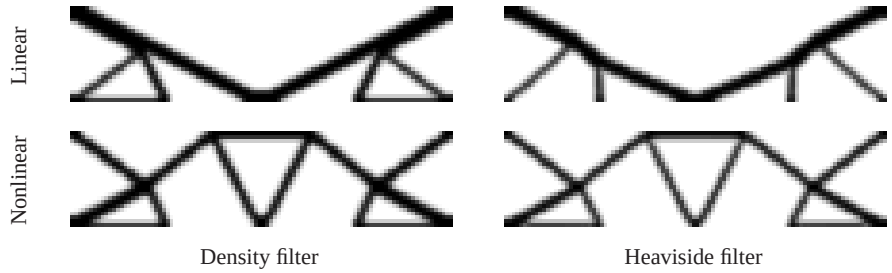

Figure 6. Structures obtained for the third problem.

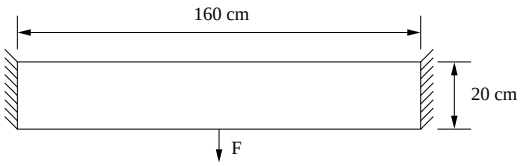

Figure 7. Design domain for the fourth example.

Table IV. Results for the fourth example.

Method	Density filter			Heaviside filter		
	Objective	Iter.	Time (s)	Objective	Iter.	Time (s)
SPLP	67.865	1219	139.11	91.447	1092	127.67
SLP	68.010	1929	178.54	91.447	1428	139.04
SQP	67.816	1159	191.64	91.487	1359	223.39

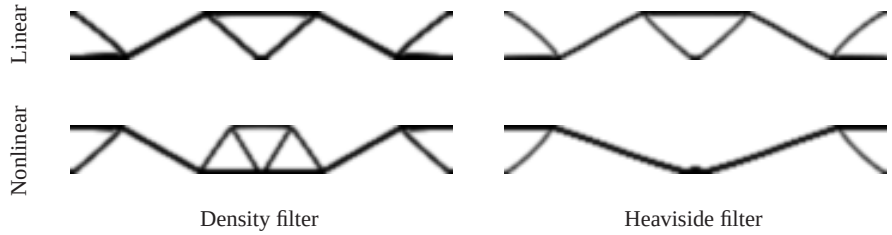

Figure 8. Structures obtained for the fourth problem.

5.5. *Problem 5*

Our fifth problem is the well-known MBB beam, whose design domain is shown in Figure 9. The beam has a thickness of $0.1\,m$ and the external load has a magnitude of $400\,kN$. The Young's modulus of the material is set to $3 \times 10^9\,N/m^2$ and the Poisson's ratio to 0.4. The domain is discretized into 2400 square elements. The optimal structure must contain no more than 50% of the domain's volume. The radius of the filters has the same length as of 3.5 elements.

The results obtained by the algorithms are given in Table V, and the optimal topologies are shown in Figure 10. As this figure reveals, the structures obtained when the nonlinear model is used have a large number of (curved) bars.

DOI: 10.1002/nme

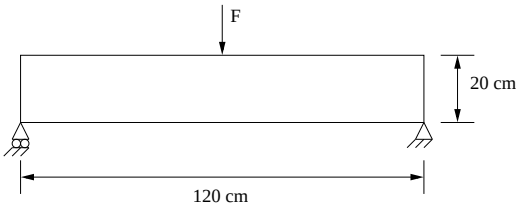

Figure 9. Design domain for the fifth problem.

Table V. Results for the fifth problem.

Method	Density filter			Heaviside filter		
	Objective	Iter.	Time (s)	Objective	Iter.	Time (s)
SPLP	6.9992×10^4	805	84.09	7.3741×10^4	1154	120.78
SLP	6.9903×10^4	1047	92.82	7.3818×10^4	2754	223.18
SQP	7.0018×10^4	809	112.80	7.3731×10^4	1090	157.48

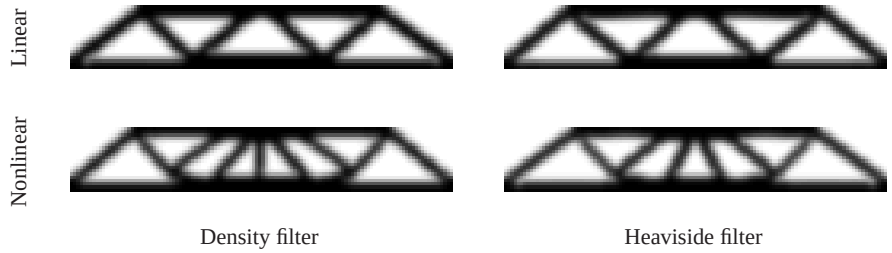

Figure 10. Structures obtained for the fifth problem.

5.6. Analysis

To allow an overall analysis of the performance of the algorithms, their runtimes (presented in Tables I to V) were also gathered using the performance profiles introduced by Dolan and Moré [43].

For each problem p and each algorithm s, let $t_{p,\,s}$ be the time spent by s to solve p. In this case, the performance ratio of one method with respect to the best algorithm is given by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s},\ \forall s \in S\}},$$

so the overall performance of algorithm s is defined by the function

$$\rho_s(\tau) = \frac{\#\{p \in \mathcal{P} \,|\, r_{p,s} \leq \tau\}}{n_p},$$

where $\#\{C\}$ is the cardinality of the set C. Function $\rho_s(\tau)$ gives the fraction of the problems that are solved by algorithm s with a performance ratio not greater than τ. Therefore, $\rho_s(1)$ gives the fraction of the problems for which algorithm s is the best algorithm.

Figure 11 shows the performance profiles of the three algorithms for the 10 problems presented above. The profiles for the small displacement model were also included in the figure, to highlight the effect of the introduction of the nonlinear model.

As we see in Figure 11, when the linear model is considered, the SLP algorithm is the fastest method for all of the problems. For these problems, the mean difference between SPLP and SLP reaches 30%, while SQP is about 160% slower than the SLP algorithm. On the other

DOI: 10.1002/nme

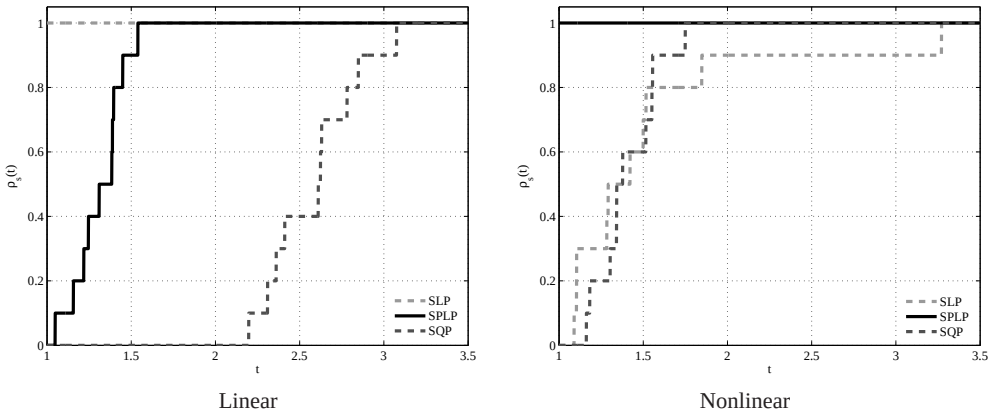

Figure 11. Performance profiles for the 10 problems.

hand, for geometrically nonlinear structures, the SPLP algorithm is by large the fastest algorithm, outperforming SLP by more than 50% and the SQP method by about 40%, on the average.

A good explanation for these results can be found in Figure 12, that shows the profiles obtained considering the number of iterations, instead of the time spent by the algorithms. As we see in this figure, the SPLP and the SQP methods take almost the same number of iterations to reach the solution, while the SLP algorithm requires, on average, 20% more iterations for the linear model, and nearly twice the number of iterations when considering the geometrically nonlinear problems. In other words, when the iterations are cheap, as it occurs under the small displacements hypothesis, we can afford to spent more iterations to solve a problem. However, as the time spent on each iteration increases, saving iterations become crucial.

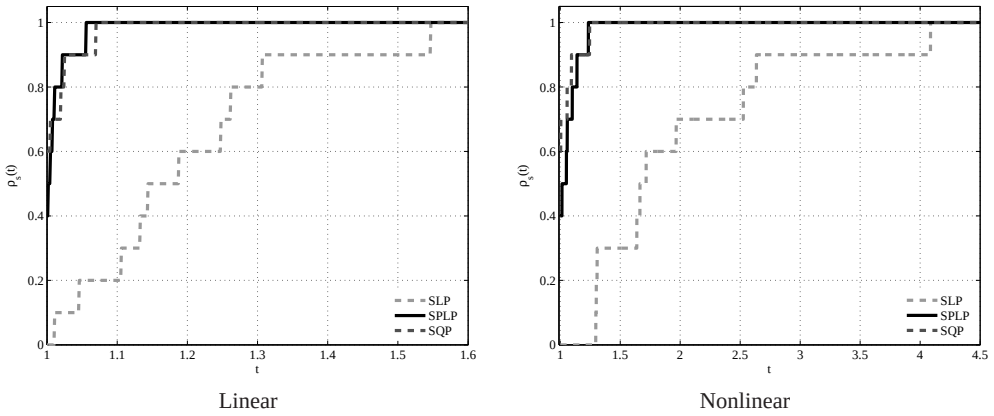

Figure 12. Profiles based on the number of iterations.

Another interesting conclusion that can be drawn from Figures 11 and 12, is that the piecewise linear function $\Gamma(\mathbf{s})$ is a very efficient approximation for the quadratic separable function $\gamma(\mathbf{s})$. The SPLP method retained the good convergence properties of the SQP algorithm, but with a faster iteration, due to its linear, instead of quadratic, subproblems.

Naturally, the number of external iterations could be further reduced using the SQP method with a better approximation for the Hessian of the Lagrangian. Yet this would create two overheads: the computation of matrix $\mathbf{B}_k$ would become more costly, and it would be necessary to solve one tougher quadratic problem per iteration. In this case, it is doubtful that the overall time spent by the SQP algorithm would decrease.

DOI: 10.1002/nme

6. CONCLUSIONS

In this paper, we have presented a new algorithm for the topology optimization of geometrically nonlinear structures. This challenging class of problems requires the solution of one nonlinear system per objective function evaluation, with the aggravation that the Jacobian matrix of this system may become singular when the structure is subject to large displacements, preventing Newton's method from converging.

For dealing with iterations that are more time consuming than those encountered when solving the linear compliance minimization problems, we propose the approximation of the problem by a SPLP model, so some information about the curvature is considered but the subproblems solved at each iteration remain linear. The new algorithm has the same global convergence properties of the SLP algorithm of Gomes and Senne [6].

The cost of computing the second derivatives is mitigated using a diagonal approximation for the Hessian of the objective function, that is computed according to a strategy proposed by Groenwold and Etman [11]. The numerical instabilities of Newton's method are addressed rescaling the density variables, that are shifted to the interval $[0.1, 1]$, allowing the solution of problems with large external loads.

The performance of the SPLP algorithm was compared to a SLP and a SQP method. The numerical tests suggest that the new algorithm is promising. In fact, it seems to combine relatively cheap steps, as done by the SLP method, with a moderate number of iterations, as in the SQP method.

As a future work, we plan to improve the efficiency of the algorithm accelerating the solution of the systems of equations by means of the approximate method proposed by Amir, Bendsøe and Sigmund [44].

REFERENCES

1. Svanberg K. The Method of Moving Asymptotes - A new method for structural optimization. *International Journal for Numerical Methods in Engineering* 1987; **24**:359–373, DOI: 10.1002/nme.1620240207.
2. Svanberg K. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization* 2002; **12(2)**:555–573, DOI: 10.1137/S1052623499362822.
3. Bruyneel M, Duysinx P, Fleury C. A family of MMA approximations for structural optimization. *Structural and Multidisciplinary Optimization* 2004; **24(4)**:263–276, DOI: 10.1007/s00158-002-0238-7.
4. Zillober Ch. Global convergence of a nonlinear programming method using convex approximations. *Numerical Algorithms* 2001; **27**:256–289, DOI: 10.1023/A:1011841821203.
5. Zillober Ch, Schittkowski K, Moritzen K. Very Large Scale Optimization by Sequential Convex Programming. *Optimization Methods and Software* 2004; **19(1)**:103–120, DOI: 10.1080/10556780410001647195.
6. Gomes FAM, Senne TA. An SLP algorithm and its application to topology optimization. *Computational and Applied Mathematics* 2011; **30**:53–89, DOI: 10.1590/S1807-03022011000100004.
7. Sigmund O. On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines* 1997; **25**:493–524, DOI: 10.1080/08905459708945415.
8. Kikuchi N, Nishiwaki S, Ono JSF, Silva ECS. Design optimization method for compliant mechanisms and material microstructure. *Computer Methods in Applied Mechanics and Engineering* 1998; **151**:401–417, DOI: 10.1016/S0045-7825(97)00161-8.
9. Nishiwaki S, Frecker MI, Seungjae M, Kikuchi N. Topology optimization of compliant mechanisms using the homogenization method. *International Journal for Numerical Methods in Engineering* 1998; **42**:535–559, DOI: 10.1002/(SICI)1097-0207(19980615)42:3¡535::AID-NME372¿3.0.CO;2-J.
10. Etman LFP, Groenwold AA, Rooda JE. First-order sequential convex programming using approximate diagonal QP subproblems. *Structural and Multidisciplinary Optimization* 2012; **45**:479–488, DOI: 10.1007/s00158-011-0739-3.
11. Groenwold AA, Etman LFP. A quadratic approximation for structural topology optimization. *International Journal for Numerical Methods in Engineering* 2010; **82**:505–524, DOI: 10.1002/nme.2774.
12. Jog C. Distributed-parameter optimization and topology design for nonlinear thermoelasticity. *Computer Methods in Applied Mechanics and Engineering* 1996; **132**:117–134, DOI: 10.1016/0045-7825(95)00990-6.
13. Buhl T, Pedersen CBW, Sigmund O. Stiffness design of geometrically nonlinear structures using topology optimization. *Structural and Multidisciplinary Optimization* 2000; **19**:93–104, DOI: 10.1007/s001580050089.
14. Bruns TE, Tortorelli D. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3443–3459, DOI: 10.1016/S0045-7825(00)00278-4.
15. Gea CH, Luo J. Topology optimization of structures with geometrical nonlinearities. *Computers and Structures* 2001; **79**:1977–1985, DOI: 10.1016/S0045-7949(01)00117-1.
16. Bruns TE, Sigmund O, Tortorelli DA. Numerical methods for the topology optimization of structures that exhibit snap-through. *International Journal for Numerical Methods in Engineering* 2002; **55**:1215–1237, DOI:

10.1002/nme.544.
17. Ohsaki M, Nishiwaki S. Shape design of pin-jointed multistable compliant mechanisms using snapthrough behavior. *Structural and Multidisciplinary Optimization* 2005; **30**:327-334, DOI: 10.1007/s00158-005-0532-2.
18. Luo Z, Tong L. A level set method for shape and topology optimization of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering* 2008; **76**:862–892, DOI: 10.1002/nme.2352.
19. Lazarov BS, Schevenels M, Sigmund O. Robust design of large-displacement compliant mechanisms. *Mechanical Sciences* 2001; **2**:175–182, DOI: 10.5194/ms-2-175-2011.
20. Lahuerta RD, Simões ET, Campello EMB, Pimenta PM, Silva ECN. Towards the stabilization of the low density elements in topology optimization with large deformation. *Computational Mechanics* 2013. Published online: 27 March 2013. DOI: 10.1007/s00466-013-0843-x.
21. Lee H, Park G. Topology Optimization for Structures With Nonlinear Behavior Using the Equivalent Static Loads Method. *Journal of Mechanical Design* 2012; **134(3)**:031004:1–14, DOI: 10.1115/1.4005600.
22. Bendsøe MP. Optimal shape design as a material distribution problem. *Structural Optimization* 1989; **1**:193–202, DOI: 10.1007/BF01650949.
23. Ciarlet PG. *Mathematical Elasticity - Vol. 1: Three-Dimensional Elasticity.* North-Holland: Netherlands, 1988.
24. Simo JC, Hughes TJR. *Computational Inelasticity.* Springer-Verlag: New York, 1998, DOI: 10.1007/b98904.
25. Ball JM. Convexity Conditions and Existence Theorems in Nonlinear Elasticity. *Archive for Rational Mechanics and Analysis* 1977; **63**:337–403, DOI: 10.1007/BF00279992.
26. Crisfield MA. *Non-linear Finite Element Analysis of Solids and Structures*, Vol. 1, John Wiley & Sons: Chichester, 2001.
27. Gomes FAM, Maciel MC, Martínez JM. Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters. *Mathematical Programming* 1999; **84**:161–200, DOI: 10.1007/s10107980014a.
28. Senne, TA. Topology optimization of structures under geometric nonlinearity (in portuguese). PhD thesis, State University of Campinas, Campinas, Brazil, 2013.
29. Byrd, R.; Nocedal, J.; Waltz, R.; Wu, Y.; On the use of piecewise linear models in nonlinear programming. *Mathematical Programming A* 2013; **137**:289–324, DOI: 10.1007/s10107-011-0492-9.
30. Byrd, R.; Nocedal, J.; Waltz, R.; Wu, Y.; An Implementation of an Algorithm for Nonlinear Programming Based on Piecewise Linear Models. Technical Report, Optimization Center, Northwestern University, Evanston, Illinois, USA, 2011.
31. Goulart E, Herskovits J. Sparse Quasi-Newton Matrices for Large Scale Nonlinear Optimization. *Proceedings of the* 6^{th} *World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, 2005; 1–11.
32. Fadel GM, Riley MF, Barthelemy JM. Two point exponential approximation method for structural optimization. *Structural Optimization* 1990; **2**:117–124, DOI: 10.1007/BF01745459.
33. Diaz AR, Sigmund O. Checkerboard patterns in layout optimization. *Structural and Multidisciplinary Optimization* 1995; **10**:40–45, DOI: 10.1007/BF01743693.
34. Bruns TE, Tortorelli DA. An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms. *International Journal for Numerical Methods in Engineering* 2003; **57**:1413–1430, DOI: 10.1002/nme.783.
35. Guest JK, Prevost JH, Belytschko T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering* 2004; **61**:238–254, DOI: 10.1002/nme.1064.
36. Sigmund O. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization* 2007; **33**:401–424, DOI: 10.1007/s00158-006-0087-x.
37. Wempner GA. Discrete Approximation Related to Nonlinear Theories of Solids. *International Journal of Solids and Structures* 1971; **17**:1581–1599, DOI: 10.1016/0020-7683(71)90038-2.
38. Riks E. An Incremental Approach to the Solution of Snapping and Buckling Problems. *International Journal of Solids and Structures* 1979; **15**:524–551, DOI: 10.1016/0020-7683(79)90081-7.
39. Batoz JL, Dhatt G. Incremental Displacement Algorithms for Nonlinear Problems. *International Journal for Numerical Methods in Engineering* 1979; **14**:1262–1267, DOI: 10.1002/nme.1620140811.
40. Crisfield MA. A Fast Incremental/Iterative Solution Procedure That Handles Snap-Through. *Computers and Structures* 1981; **13**:55–62, DOI: 10.1016/0045-7949(81)90108-5.
41. Chen Y, Davis TA, Hager WW, Rajamanickam S. Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software* 2008; **35(3)**:22:1–14. DOI: 10.1145/1391989.1391995.
42. Jung D, Gea HC. Topology optimization of nonlinear structures. *Finite Element in Analysis and Design* 2004; **40**:1417–1427, DOI: 10.1016/j.engstruct.2008.01.009.
43. Dolan ED, Moré JJ. Benchmarking optimization software with performance profiles. *Mathematical Programming A* 2002; **91**:201–213, DOI: 10.1007/s101070100263.
44. Amir O, Bendsøe MP, Sigmund O. Approximate reanalysis in topology optimization. *International Journal for Numerical Methods in Engineering* 2009; **78**:1474–1491, DOI: 10.1002/nme.2536.