A GRADIENT-TYPE ALGORITHM FOR
CONSTRAINED OPTIMIZATION WITH APPLICATION TO
MICROSTRUCTURE OPTIMIZATION

Cristian Barbarosie* and Anca-Maria Toader
CMAF-CIO, Faculdade de Ciências, Universidade de Lisboa
1749-016 Lisboa, Portugal

Sérgio Lopes
CMAF-CIO, Faculdade de Ciências, Universidade de Lisboa
1749-016 Lisboa, Portugal
and Dep. Area of Mathematics, ISEL, Instituto Politécnico de Lisboa
Rua Conselheiro Emídio Navarro, 1959-007 Lisboa, Portugal

Abstract. We propose a method to optimize periodic microstructures for obtaining homogenized materials with negative Poisson ratio, using shape and/or topology variations in the model hole. The proposed approach employs worst case design in order to minimize the Poisson ratio of the (possibly anisotropic) homogenized elastic tensor in several prescribed directions. We use a minimization algorithm for inequality constraints based on an active set strategy and on a new algorithm for solving minimization problems with equality constraints, belonging to the class of null-space gradient methods. It uses first order derivatives of both the objective function and the constraints. The step is computed as a sum between a steepest descent step (minimizing the objective functional) and a correction step related to the Newton method (aiming to solve the equality constraints). The linear combination between these two steps involves coefficients similar to Lagrange multipliers which are computed in a natural way based on the Newton method. The algorithm uses no projection and thus the iterates are not feasible; the constraints are only satisfied in the limit (after convergence). A local convergence result is proven for a general nonlinear setting, where both the objective functional and the constraints are not necessarily convex functions.

1. Introduction. Structural optimization is the field that deals with designing mechanical structures aiming at improving their performance, in a sense to be defined, by varying a number of variables, usually called “structural parameters”. See e.g. [14], [27], [26]. Different structural parameters can be considered, like the shape of the object, or the size and location in space of pieces composing the structure (and the way they are linked to each other), or even the mechanical properties of the constituent material(s).

Structural optimization problems have some common properties. These problems usually involve a large number of variables (structural parameters) and also
constraints, e.g. on the production cost of the structure, or on its weight, or on its natural frequencies. But the most striking property is the complicated dependence of the functionals (objective functional and constraint functionals) with respect to the optimization variables (structural parameters). These functionals depend on a state variable, which in turn depends on the structural parameters by means of a partial differential equation (or a system of such equations). In Section 2, an example is studied which involves the Poisson ratio of the homogenized elastic tensor associated to a periodic microscopic perforation of a body.

A consequence of this complicated dependence of the functionals is that their derivatives, with respect to the structural parameters, are difficult to compute and their evaluation is numerically expensive. First order derivatives can be computed (this is a challenging exercise from the theoretical point of view), but second order derivatives are much too expensive or even impossible to compute.

Due to these difficulties in computing the derivatives of the functionals, the choice of the numerical algorithm for optimization is quite limited. Either one uses derivative free algorithms or an algorithm based on first order derivatives, like the gradient method or some slight improvement of it. One cannot rely on second order derivatives, and this makes the treatment of the constraints quite difficult. In particular, applying Newton’s method to the Karush-Kuhn-Tucker conditions is not affordable. Also, projection methods are often not appropriate, again due to the complicated dependence of the functional on the optimization variables.

For the reasons described above, numerical approaches to structural optimization problems deal with constraints in several ways. One solution is to add a penalization term for the constraint, with a constant Lagrange multiplier, see [3, Section 6]. This way, one has no control on the final value of the constraint functional, other than repeating the optimization process with different values of the multiplier. Another solution is to tune the corresponding multiplier at each step of the optimization process, using a formula based on numerical intuition rather than on mathematical arguments, see e.g. [1, paragraph 5.2]. Still another solution is the method of moving asymptotes, which ends up being almost equivalent to the previously mentioned tuning of the Lagrange multiplier; see e.g. [28] and [32]. Both are conceptually unclear and require the user to set values for certain parameters. Other methods based on gradients are penalty methods, (augmented) lagrangean methods, interior point and trust region methods, sequential linear programming and sequential quadratic programming, see [12, 23, 34, 16].

In the present paper, we propose an algorithm directed at optimization applications where second derivatives are either unavailable or prohibitively expensive, and first derivatives themselves are challenging (but obtainable). The method, which is quite natural, easy to implement and conceptually clear, aims at the minimization (or maximization) of a functional, subject to constraints, and it is based on gradients only.

Section 2 shows an application on a large-scale example, involving the minimization of the Poisson ratio(s) of a composite material (see [22, 20]), in the context of homogenization theory (see e.g. [2]). To achieve this goal, we perform shape and/or topology variations in the model hole that characterizes the microstructure. We use the minimax algorithm (introduced in Section 3) in order to minimize simultaneously the Poisson coefficient of the (possibly anisotropic) homogenized material along many directions of the plane.
Section 3 is devoted to the description of the algorithm and to a convergence result, along with extensions to accommodate inequality constraints and to deal with minimax problems. The method seeks for local solutions and works in the general case, for nonlinear and nonconvex objective functional and constraints. Smoothness of these functions is however required, since their gradients are used. The method is twofold, involving a sum between a steepest descent step (which minimizes the objective functional) and a correction step related to the Newton method (which aims to solve the equality constraints). The linear combination between these two steps uses certain coefficients similar to Lagrange multipliers, which are computed in a natural way based on the Newton method. The algorithm uses no projection and thus the iterates are not feasible; the constraints are satisfied only in the limit (after convergence). Convergence is proven under the hypothesis that a certain Hessian-like matrix is positive definite at the solution. Inequality constraints are dealt with by using an active-set methodology. Special attention is given to activation and deactivation strategies; the deactivation criterion is entirely based on the sign of the associated Lagrange multiplier. The case of an infinite (continuous) family of inequality constraints is discussed, as well as the extension of the method for treating minimax problems.

Section 4 presents numerical results obtained by applying this algorithm to the problem described in Section 2, and some closing comments are made in Section 5.

2. Optimization of auxetic materials. The main motivation of the present paper comes from the study of periodic microstructures and optimization of their macroscopic properties, in the context of linearized elasticity.

This section describes an optimization problem whose goal is to obtain a composite material having negative Poisson ratio along all directions. Materials having a negative Poisson ratio are called auxetic materials.

A composite material will be described as a periodic microstructure, that is, a nonlinearly elastic body whose material coefficients vary at a microscopic scale, according to a periodic pattern. Homogenization theory allows one to accurately describe the macroscopic behaviour of such a microstructure by means of so-called cellular problems, which are elliptic PDEs subject to periodicity conditions, see e.g. [2]. Porous materials, that is, bodies with periodic infinitesimal perforations, can be described in a similar manner.

The present study focuses on minimizing the Poisson ratio of the homogenized material, in the spirit of [7, Section 7] and [8, Subsection 6.6].

2.1. The cellular problem. We shall consider a model hole, which is a compact set $T \subset Y$ (see Figure 1), where $Y$ is the periodicity cell. Usually, $Y$ is the unit cube in $\mathbb{R}^N$; see, however, [7] for a general notion of periodicity. In the sequel we shall discuss only the two-dimensional case $N = 2$; however, most of the theoretical considerations hold for higher dimension, too.

The perforated body is obtained by removing from $\mathbb{R}^2$ translations of the model hole. For a cubic cell $Y$, one has (see Figure 2)

$$\mathbb{R}^2_{\text{perf}} = \mathbb{R}^2 \setminus \bigcup_{\vec{k} \in \mathbb{Z}^n} (T + \vec{k})$$

(1)

The cellular problem describing the behaviour of such a porous material is:
The above problem models the mechanical behaviour of a microstructure with elastic tensor $\mathbf{C}$, occupying the domain $\mathbb{R}_\text{perf}^2$ and subject to the macroscopic strain $\mathbf{A}$. In (2), $\mathbf{u}_\mathbf{A}$ denotes the displacement of the body, $\varepsilon$ denotes the symmetric part of the gradient and $\mathbf{n}$ is the unit vector normal to the boundary of the hole. The third equation in (2) states the mechanical equilibrium, while the fourth condition states that the holes are free of tractions.

The homogenized elastic tensor $\mathbf{C}^H$, describing the effective (macroscopic) behaviour of this microstructure, is given by

$$\mathbf{C}^H \mathbf{A} = \frac{1}{|Y|} \int_{Y \setminus T} \mathbf{C} \varepsilon(\mathbf{u}_\mathbf{A})$$

or

$$\langle \mathbf{C}^H \mathbf{A}, \mathbf{B} \rangle = \frac{1}{|Y|} \int_{Y \setminus T} \langle \mathbf{C} \varepsilon(\mathbf{u}_\mathbf{A}), \varepsilon(\mathbf{u}_\mathbf{B}) \rangle,$$

where $\mathbf{A}$ and $\mathbf{B}$ are given macroscopic strains.

The cellular problem (2) can be reformulated in stress, as follows (see [7]):

$$\begin{align*}
\mathbf{w}_\sigma &\in L^p_{\text{perf}}, \\
-\text{div}(\mathbf{C}\varepsilon(\mathbf{w}_\sigma)) &= 0 \text{ in } \mathbb{R}_\text{perf}^2 \\
\mathbf{C}\varepsilon(\mathbf{w}_\sigma)\mathbf{n} &= 0 \text{ on } \partial T \\
\frac{1}{|Y|} \int_{Y \setminus T} \mathbf{C}\varepsilon(\mathbf{w}_\sigma) &= \mathbf{\sigma},
\end{align*}$$

where $\mathbf{\sigma}$ represents an applied macroscopic stress and $L^p_{\text{perf}}$ is the set of vector fields which are sums of a linear map and a periodic field.

We shall denote by $\mathbf{D}^H$ the homogenized compliance tensor, that is, the inverse of $\mathbf{C}^H$. 
2.2. **Shape and topology derivatives.** The effective elastic properties of the above described porous body can be optimized by varying the size and shape of existing holes in the periodicity cell $Y$, and also by creating new, infinitesimal, holes.

The first approach is called shape optimization (here applied at the cellular level). The shape derivative describes the variation of a certain objective functional when an infinitesimal deformation is applied to a given geometry, see [29, 17, 18]. Consider $\vec{\theta} : \mathbb{R}^2 \to \mathbb{R}^2$ a vector field defining the deformation; note that $\vec{\theta}$ itself should be periodic in order to preserve the periodic character of the microstructure under study. Then the variation induced by this deformation in the quantity $(C^H A, B)$ is (see [7, Section 6] and [6])

$$D_S(C^H A, B) = \frac{1}{|Y|} \int_{\partial T} \langle C \varepsilon(\vec{u}_A) , \varepsilon(\vec{u}_B) \rangle \vec{\theta} \cdot \vec{n}$$

where $\vec{n}$ is the unit vector normal to the boundary of the hole $T$ and pointing inside $T$. Assuming that $C$ is a linear isotropic elastic tensor, $C\xi = 2\mu\xi + \lambda(\text{tr}\xi)I$, the above formula becomes

$$D_S(C^H A, B) = \frac{1}{|Y|} \int_{\partial T} [2\mu(\varepsilon(\vec{u}_A) , \varepsilon(\vec{u}_B)) + \lambda\text{tr}(\varepsilon(\vec{u}_A))\text{tr}(\varepsilon(\vec{u}_B))] \vec{\theta} \cdot \vec{n}$$

In particular, this gives the shape derivative of the homogenized coefficients:

$$D_S C^H_{ij} = D_S \langle C^H f_i , f_j \rangle = \frac{1}{|Y|} \int_{\partial T} [2\mu(\varepsilon(\vec{u}_{f_i}) , \varepsilon(\vec{u}_{f_j})) + \lambda\text{tr}(\varepsilon(\vec{u}_{f_i}))\text{tr}(\varepsilon(\vec{u}_{f_j}))] \vec{\theta} \cdot \vec{n}$$

(6)

where $(f_i)_{i=1,2,3}$ is the following basis in the space of symmetric matrices

$$f_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} , \ f_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} , \ f_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

and $\vec{u}_{f_i}$ are the corresponding solutions of the cellular problem (2) with effective strain $f_i$.

A second approach for the optimization of a structure is topology variation (here applied at the cellular level). It consists in drilling an infinitesimal circular hole and imposing zero Neumann condition on the newly created boundary, see [24, 25]. The topological derivative describes the infinitesimal variation thus induced in the functional $(C^H A, B)$, and depends on the location $x$ of the new hole. It can be proven (see [7, Section 5] and [30]) that the topological derivative is given by

$$D_T(C^H A, B)(x) = -\frac{\pi}{|Y|} \frac{\lambda + 2\mu}{\lambda + \mu} \left[ 4\mu\varepsilon(\vec{u}_A)\varepsilon(\vec{u}_B) + \frac{\lambda^2 + 2\lambda\mu - \mu^2}{\mu} \text{tr} \varepsilon(\vec{u}_A) \text{tr} \varepsilon(\vec{u}_B) \right](x)$$

(7)

In [8], an algorithm was proposed for optimizing the microgeometry of the hole(s) in the cellular problem, with the goal of improving certain macroscopic properties of the porous microstructure (which is a body with periodically distributed infinitesimal perforations). The algorithm alternates shape variations with topology variations until a certain convergence criterion is fulfilled. The properties to be optimized include the effective bulk modulus, the effective response to shear and the effective Poisson coefficient (see [8, Section 6]).
Both shape and topology derivatives of the homogenized compliance tensor $D^H$ are obtained from the derivatives of the homogenized tensor $C^H$ by:

$$
\begin{align*}
D_S D^H_{ijkl} &= -D^H_{ij\alpha\beta} D_S C^H_{\alpha\beta\gamma\delta} D^H_{\gamma\delta kl}, \\
D_T D^H_{ijkl} &= -D^H_{ij\alpha\beta} D_T C^H_{\alpha\beta\gamma\delta} D^H_{\gamma\delta kl}.
\end{align*}
$$

(8)

Note that formula (8) uses the coordinate notation for the fourth-order tensors $C^H$ and $D^H$, while in (6) the indices $i$ and $j$ are relative to the basis $(f_i)_{i=1,2,3}$ in the space of symmetric matrices.

2.3. Poisson ratios and the minimax technique. This work focuses on the search of two-dimensional periodic microstructures exhibiting negative Poisson ratio at the macroscopic level (so-called auxetic materials). In previous works from the same authors, [7] and [8], anisotropic effective elastic tensors have been obtained which exhibit negative Poisson ratio in a prescribed direction of the plane (the horizontal direction), see Figure 3 while taking into account Remarks 6 and 7 in Section 4. In the present work, we look for periodic microstructures with the same negative Poisson ratio among all directions in the plane. This is done by combining the techniques described in the above Subsection 2.2 (for shape optimization at the cellular level) with the minimax algorithm described in Subsection 3.7 which ensures that the largest Poisson ratio among many directions in the plane is being minimized.

Note that the effective elastic tensor resulting from the homogenization technique, defined by (3) or (4), is not isotropic in general. Thus, the notion of Poisson coefficient must be defined with care: it is minus the ratio between the transverse strain and the axial strain when the material is stretched or compressed along the axial direction, see [13]. In the two-dimensional case under consideration, a Poisson ratio can be associated to each unit vector $\vec{v} = (v_1, v_2)$ arbitrarily chosen in the plane. Consider a stretching stress applied along the direction of $\vec{v}$; in the frame $\{\vec{v}, \vec{v}^\perp\}$ the stress writes $\sigma = \begin{bmatrix} 1 & \nu \nu \end{bmatrix}$. Then the Poisson ratio $\nu_v$ in the direction $\vec{v}$ is defined as $\nu_v = -\varepsilon_{\perp\perp}/\varepsilon_{vv}$, where $\varepsilon_{\perp\perp}$ is the strain in the direction $\vec{v}^\perp$ and $\varepsilon_{vv}$ is the strain in the direction $\vec{v}$. The above defined stress $\sigma$ expressed in cartesian coordinates has the form

$$
\sigma = \begin{bmatrix}
\nu_1^2 & \nu_1 \nu_2 \\
\nu_1 \nu_2 & \nu_2^2
\end{bmatrix}
$$

(9)
and the associated strain matrix is $\varepsilon = D^H \sigma$ (recall that $D^H$ is the homogenized compliance tensor). Then the axial strain is $\varepsilon_{\nu \nu} = D^H \sigma \cdot \hat{v}$ and the transverse strain is $\varepsilon_{\perp \perp} = D^H \sigma \cdot \hat{v}^\perp / D^H \sigma \cdot \hat{v}$ and introducing the stress corresponding to a stretch in the direction $\hat{v}$, denoted by

$$\sigma^\perp = \begin{bmatrix} v_2^2 & -v_1v_2 \\ -v_1v_2 & v_1^2 \end{bmatrix},$$

(10)

it becomes:

$$\nu = -\frac{\langle D^H \sigma, \sigma^\perp \rangle}{\langle D^H \sigma, \sigma \rangle}.$$  

(11)

The derivative of $\nu$ with respect to $D^H$ is given by:

$$\frac{\partial \nu}{\partial D^H_{ijkl}} = -\frac{\sigma_{ij}^\perp \sigma_{kl}}{\langle D^H \sigma, \sigma \rangle} + \frac{\langle D^H \sigma, \sigma^\perp \rangle}{\langle D^H \sigma, \sigma \rangle^2} \sigma_{ij} \sigma_{kl}$$  

(12)

The above formulae (11) and (12) are suitable for implementation since they can actually be seen as depending on the homogenized tensor $C^H$, see (8).

Summing up the above, we are dealing with a problem of minimizing several objective functionals (the Poisson ratios along different directions in the plane) in simultaneous.

The problem of finding materials having negative Poisson ratios along several directions in the plane will be numerically solved (in Section 4) by applying a new minimax algorithm introduced in Section 3 below.

3. The minimization algorithm. In this section, we propose an algorithm for the minimization of a functional subject to constraints. The algorithm seeks for local solutions (as usual for gradient-based methods). It deals with nonessential constraints, that is, with constraints whose violation does not render the problem ill-posed. In the method here proposed, the constraints are usually violated during the optimization process, and become satisfied only in the limit (after convergence); see, however, Subsection 3.4 for an exception.

The algorithm can be seen as a null-space gradient method, see e.g. [33, 4]. It has been described in [6] without proof of convergence; its local convergence has been proven in [9].

We begin by describing the case of equality constraints (Subsection 3.1). The convergence of the algorithm is proven under the hypothesis that a certain Hessian-like matrix is positive definite at the solution (Theorem 3.2 in Subsection 3.2). Our methodology can be regarded as a gradient method applied in the direction tangent to the manifold determined by the constraints, together with a Newton method applied in the orthogonal direction. This method, although not very fast (it has linear convergence) is quite natural, easy to implement, and has the advantage of requiring solely the first derivatives of the objective and of the constraint functions.

A generalization is proposed in Subsections 3.3 and 3.5 which deals with inequality constraints, based on an active-set strategy. During the optimization process, an inequality constraint is activated as soon as it is violated. Its deactivation depends on the sign of the associated Lagrange multiplier. To some extent, this procedure can be seen as a generalization of the simplex method to nonlinear functions. Based on these ideas, in Subsection 3.7 we further extend the algorithm with minimax problems in sight.
Remark 1. We would like to stress beforehand that, although these methods can act as general purpose nonlinear optimization solvers, they were not designed with such a purpose in mind. In fact, benchmarking these algorithms with existing databases of test problems would be an unnecessary venture, as they would no doubt underperform against any serious contender in the field. The proposed methods are to be regarded in light of the very specific context in which they will be applied, namely that of Section 2 or, more generally, to optimization problems where only first derivatives are obtainable.

The following notation will be used: $x \in \mathbb{R}^n$ is the vector of variables (also called unknowns); $x_i$ will denote the components of $x$ while $x^{(k)}$ will denote a sequence of vectors; $f$ is the objective function, a scalar function of $x$ that we want to minimize or maximize; the constraints will be modelled by a vector function $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$. The Jacobian matrix of a vector function $g$ will be denoted by $Dg$ while its transpose will be denoted by $\nabla g$. In particular, for a scalar function $f$, $Df$ will be the row gradient while $\nabla f$ will be the usual (column) gradient. The Hessian matrix of $f$ will be denoted by $D^2 f$. Partial derivatives will be denoted by a comma, e.g. $g_{i,j} = \frac{\partial g}{\partial x_j}$.

3.1. A gradient algorithm for equality constrained problems. Consider the minimization problem

$$\min_{x \in C} f(x), \quad C = \{x \in \mathbb{R}^n : g(x) = 0\}. \quad (13)$$

A typical case in structural design arises when engineers adjust the variables to optimize the performance of a structure while keeping a prescribed cost. In such a framework, the constraint function $g$ appearing in (13) will be the difference between the cost of the structure and a prescribed “price”. For presentation purposes, the discussion will be initially restricted to this case of only one constraint ($m = 1$) and subsequently extended to account for multiple constraints.

The algorithm should pursue two goals simultaneously: decrease the value of $f$ while solving the equation $g = 0$. Our approach sets up a direction that targets both goals at once, as described below.

Given an iterate $x^{(k)}$, the next iterate will be defined by an increment $\delta^{(k)}$, that is, $x^{(k+1)} = x^{(k)} + \delta^{(k)}$. The increment $\delta^{(k)}$ will be the sum of two components: one of them is the vector $-\eta \nabla f(x^{(k)})$ (with $\eta > 0$ fixed) corresponding to the steepest descent algorithm; the other one aims at fulfilling the constraint equation $g = 0$ and has the form $-\lambda^{(k)} \nabla g(x^{(k)})$, where $\lambda^{(k)} \in \mathbb{R}$ is a sort of Lagrange multiplier:

$$\delta^{(k)} = -\eta \nabla f(x^{(k)}) - \lambda^{(k)} \nabla g(x^{(k)}).$$

The multiplier $\lambda^{(k)}$ is defined adaptively in a natural way, inspired in Proposition 1, Appendix B. It suffices to impose the Newton-type condition, relative to the equation $g = 0$,

$$\langle \nabla g(x^{(k)}), \delta^{(k)} \rangle = -g(x^{(k)})$$

which is immediately solvable:

$$\lambda^{(k)} = \frac{g(x^{(k)}) - \eta \langle \nabla g(x^{(k)}), \nabla f(x^{(k)}) \rangle}{\| \nabla g(x^{(k)}) \|^2}. \quad (14)$$

With this choice of the multiplier, the whole procedure amounts to performing a “tangential gradient method” to minimize $f$, together with a unidimensional Newton method to solve the constraint equation $g = 0$. 
To better understand the last assertion, consider the following reasoning. In the neighborhood of a solution \( x^* \) there are two main directions to consider from \( x^{(k)} \): the direction \( \nabla g(x^{(k)}) \), orthogonal to the level set \( C_k = \{ x \in \mathbb{R}^n : g(x) = g(x^{(k)}) \} \), and the subspace orthogonal to it (whose vectors are tangent to \( C_k \) at \( x^{(k)} \)). In this latter subspace we have to minimize \( f \) (note that, since the solution \( x^* \) should minimize \( f \) in a level set of \( g, C \), there is no point in decreasing \( f \) along directions other than tangent ones); in the direction of \( \nabla g(x^{(k)}) \) we want to solve the equation \( g = 0 \), moving the next iterate closer to \( C \). A very simple method is obtained which, to the best of our knowledge, is not mentioned in the literature.

**Remark 2.** The algorithm here proposed is somewhat similar to the Newton method described in [12, Section 12.1], with the major difference that we use information related to the first derivatives only (of both the objective function and the constraints). We once again underline that there are many practical problems in which second derivatives are impossible (or very expensive) to compute, see Section 2.

The algorithm generalizes naturally to vector-valued constraint functions \( g : \mathbb{R}^n \to \mathbb{R}^m \) (with \( m < n \)). In this case \( \lambda^{(k)} \in \mathbb{R}^m \) but the iterates are defined in a similar fashion by

\[
x^{(k+1)} = x^{(k)} - \eta \nabla f(x^{(k)}) - \nabla g(x^{(k)}) \lambda^{(k)}.
\]

By imposing the Newton-type condition (again, inspired in Proposition 1, Appendix B)

\[
Dg(x^{(k)}) \delta^{(k)} = -g(x^{(k)}),
\]

we obtain

\[
Dg(x^{(k)}) \nabla g(x^{(k)}) \lambda^{(k)} = g(x^{(k)}) - \eta Dg(x^{(k)}) \nabla f(x^{(k)}).
\]  

(15)

In coordinate notation:

\[
x^{(k+1)}_j = x^{(k)}_j - \eta f_{j,j}(x^{(k)}) - \sum_{i=1}^{m} \lambda^{(k)}_i g_{i,j}(x^{(k)}), \quad 1 \leq j \leq n,
\]

where

\[
\sum_{i=1}^{m} \sum_{j=1}^{n} g_{i,j}(x^{(k)}) g_{i,j}(x^{(k)}) \lambda^{(k)}_i = g(x^{(k)}) - \sum_{j=1}^{n} \eta g_{l,j}(x^{(k)}) f_{j,j}(x^{(k)}), \quad 1 \leq l \leq m.
\]

This linear system of equations uniquely determines \( \lambda^{(k)} \) if \( Dg(x^{(k)}) \) has full rank (equal to \( m \)), see Definition 3.1 in Subsection 3.2 and the comments following it. Even in the case of vector-valued constraints, the method can be interpreted geometrically as a steepest descent method in the directions tangent to \( C_k \), combined with a Newton method in the directions normal to \( C_k \).

**Algorithm 1 (for equality constraints).**

INPUT: initial guess \( x^{(0)} \), step size \( \eta > 0 \), tolerance \( \varepsilon > 0 \), maximum number of iterations \( N \).

OUTPUT: approximate solution \( x \) or message of failure.

**Step 1** With \( k \) from 1 to \( N \), do Steps 2–5.

**Step 2** Compute \( \lambda \) by solving \( Dg(x^{(0)}) \nabla g(x^{(0)}) \lambda = g(x^{(0)}) - \eta Dg(x^{(0)}) \nabla f(x^{(0)}) \).

**Step 3** Set \( x = x^{(0)} - \eta \nabla f(x^{(0)}) - \nabla g(x^{(0)}) \lambda \).

**Step 4** If \( \| x - x^{(0)} \| < \varepsilon \) then OUTPUT(\( x \));

STOP.
Step 5 Set $x^{(0)} = x$.
Step 6 OUTPUT('The method failed after $N$ iterations.');// STOP.

This type of algorithm has already been used in [6] to solve large-scale optimization problems.

3.2. Convergence results. In the present Subsection, the convergence of Algorithm 1 is proven, under suitable hypotheses.

As usual, some kind of "constraint qualification" will be assumed to hold at a solution $x^* \in \mathcal{C}$ of problem (13); namely the Linear Independence Constraint Qualification condition, which is the subject of the following definition.

Definition 3.1. A point $x \in \mathbb{R}^n$, satisfying the constraint $g(x) = 0$, is said to be a regular point if the gradient vectors $\nabla g_1(x), \nabla g_2(x), \ldots, \nabla g_m(x)$ are linearly independent. In other words, the Jacobian matrix $Dg(x)$ should have full rank (equal to $m$).

At a regular point $x$, the constraint function $g$ is a submersion, giving $\mathcal{C}$ the appropriate geometrical concept: that of a manifold in $\mathbb{R}^n$; the tangent subspace to $\mathcal{C}$ is known to be $T_x = \{ \tau \in \mathbb{R}^n : Dg(x) \tau = 0 \}$. Note also that $m < n$; in fact, $m \geq n$ usually yields a discrete set of feasible points, a situation which is outside the scope of the present paper.

We now state and prove the main theorem regarding the method proposed in Subsection 3.1 above.

Theorem 3.2. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ($m < n$) be twice continuously differentiable functions. Let $x^* \in \mathcal{C}$ be a regular point satisfying the Karush-Kuhn-Tucker (KKT for short) conditions

$$
\begin{align*}
\nabla f(x^*) + \nabla g(x^*) \lambda^* &= 0, \\
\lambda^*(x^*) &= 0,
\end{align*}
$$

and such that the matrix $H^* = D^2 f(x^*) + \sum_{i=1}^{m} \lambda^*_i D^2 g_i(x^*)$ is positive definite on $T_{x^*}$, that is, for any vector $\tau \neq 0$ tangent to $\mathcal{C}$ at $x^*$, there holds $\langle H^* \tau, \tau \rangle > 0$. Then there exists $r > 0$ such that, given $x^{(0)} \in B_r(x^*)$, the sequence of iterates defined by

$$
x^{(k+1)} = x^{(k)} - \eta \nabla f(x^{(k)}) - \nabla g(x^{(k)}) \lambda^{(k)}, \quad k \in \mathbb{N}_0,
$$

with $\lambda^{(k)}$ determined by (15), converges linearly to $x^*$ for sufficiently small step lengths $\eta > 0$.

In the above, $B_r(x^*)$ is the closed ball of radius $r$, centered at $x^*$.

The main ingredient in the proof of Theorem 3.2 is to regard (17) as a fixed-point iteration, so that Banach’s fixed-point theorem applies. First of all, an auxiliary result is established.

Lemma 3.3. Let $P \neq 0$ be an orthogonal projection on $\mathbb{R}^n$. If $A \neq 0$ is a self-adjoint linear operator on $\mathbb{R}^n$, then $v \neq 0$ is an eigenvector of $PA$, associated with the eigenvalue $\mu \neq 0$, if and only if

(i) $v \in \text{Ran}(P),$  
(ii) $(A - \mu I)v \in \text{Ker}(P).$

Hence, the following estimate of the spectral radius holds: $\rho(PA) \leq \rho(A|_{\text{Ran}(P)})$. 


Proof. The “if” part of the assertion is trivial. The “only if” part follows basically from the fact that, \( P \) being an orthogonal projection, one has the direct sum decomposition \( \mathbb{R}^n = \text{Ker}(P) \oplus \text{Ran}(P) \). Hence, given an eigenpair \( \mu \neq 0 \) and \( \mu \neq 0 \) of \( PA \), there are unique \( v \in \text{Ker}(P) \) and \( w \in \text{Ran}(P) \) such that \( Au = v + w \); but then, \( PAu = \mu u \) reads \( w = \mu u \). Therefore, it must be \( u \in \text{Ran}(P) \) and \( Au - \mu u = v \in \text{Ker}(P) \).

The last estimate is now obvious, since \( \rho(PA) = \rho(PA|_{\text{Ran}(P)}) \) and the spectral radius of an operator is dominated by the \( \ell^2 \) norm of that same operator (recall also that \( \| P \|_2 = 1 \) and that the spectral radius of a self-adjoint operator equals its \( \ell^2 \) norm).

Remark 3. Another useful result regarding spectral radii and matrix norms (whose proof can be found in [15, Section 1.4]), is that for any square matrix \( A \) and \( \varepsilon > 0 \), there exists a natural norm \(^{1}\) with the property that \( \| A \| < \rho(A) + \varepsilon \). One then concludes that contractivity properties of continuously differentiable maps \( S : \mathbb{R}^n \to \mathbb{R}^n \) are essentially governed by the spectral radius of their Jacobian matrices: if \( \rho(DS(x)) < 1 \), it always exists a vector norm for which \( S \) is locally contractive around \( x \).

Proof of Theorem 3.2. We begin by rewriting the algorithm to display its fixed point nature. Because \( x^* \) is a regular point, \( Dg(x^*) \) has full rank and the same is true for \( Dg(x) \) with \( x \) nearby \( x^* \). Thus, equation (15) has a unique solution

\[
\lambda^{(k)} = \frac{\vert Dg(x^{(k)}) \nabla g(x^{(k)}) \vert^{-1}}{\| g(x^{(k)}) \|} = \eta \frac{\nabla f(x^{(k)}) - \nabla g(x^{(k)})}{\| \nabla g(x^{(k)}) \|}.
\]

Putting this expression into (17) yields

\[
x^{(k+1)} = x^{(k)} - \eta \left( I - \nabla g(x^{(k)}) \left[ Dg(x^{(k)}) \nabla g(x^{(k)}) \right]^{-1} \right) \nabla f(x^{(k)})
\]

so \( x^{(k+1)} = S(x^{(k)}) \), upon defining \( S(x) = x - \eta P(x) \nabla f(x) - K(x) g(x) \). Because \( x^* \) is a regular point, the operator \( S \) is well defined locally around \( x^* \). Harking back to Remark 3, one is left to establish that \( \rho(DS(x^*)) < 1 \). The conclusion will then follow by the Banach fixed-point theorem applied on \( \hat{B}_r(x^*) \), for some sufficiently small \( r > 0 \) (see e.g. [19, Chapter 5] or [5, Chapter 5]).

\( K(x) \) is clearly a right inverse of \( Dg(x) \) and it is not difficult to prove that \( P(x) \) is the matrix of the orthogonal projection onto the tangent subspace \( T_x \) to \( C_x \) \( \{ y \in \mathbb{R}^n : g(y) = g(x) \} \) at \( x \). There are some trivial relations involving \( P(x) \), \( K(x) \) and \( Dg(x) \), namely: \( K(x) Dg(x) = I - P(x) \), \( P(x) K(x) = 0 \) and \( P(x) \nabla g(x) = 0 \); in view of this last equality, one can write

\[
S(x) = x - \eta P(x) [\nabla f(x) + \nabla g(x) \lambda] - K(x) g(x),
\]

and it is now easy to see, due to the KKT conditions (16), that the Jacobian matrix of \( S \) at \( x^* \) is given by

\[
DS(x^*) = I - \eta P(x^*) H^* - K(x^*) Dg(x^*)
\]

\[
= I - \eta P(x^*) H^* - [I - P(x^*)] = P(x^*) (I - \eta H^*).
\]

\(^{1}\)A matrix norm that is associated with a vector norm is called a natural norm.
Since $I - \eta H^*$ is a symmetric matrix and $P(x^*)$ is the orthogonal projection’s matrix onto $\mathcal{T}_x^*$, precisely the subspace where $H^*$ is positive definite, the eigenvalues of $(I - \eta H^*)|_{\mathcal{T}_x^*}$ take the form $1 - \eta \mu_i^*$ ($1 \leq i \leq n - m \equiv k$), where $\mu_i^* \geq \mu_2^* \geq \cdots \geq \mu_k^*$ are the positive eigenvalues of $H^*|_{\mathcal{T}_x^*}$ (note that we loosely identified a matrix with the linear operator it represents); therefore, we have $1 - \eta \mu_i^* \in [1 - \eta \mu_1^*, 1 - \eta \mu_k^*]$ ($1 \leq i \leq k$) and the choice $0 < \eta < \frac{2}{\mu_1^*}$ implies that $[1 - \eta \mu_1^*, 1 - \eta \mu_k^*] \subset ] - 1, 1]$. Hence, recalling Lemma 3.3, one gets indeed $\rho(DS(x^*)) < 1$. 

Remark 4. The “true” Lagrange multiplier $\lambda^*$ can be easily approximated because the functional expression defining $\lambda^{(k)}$, using either (14) or (18) depending on the number of constraints, evaluates to $\eta \lambda^*$ at $x^*$. More precisely, the function

$$\Lambda(x) = [Dg(x) \nabla g(x)]^{-1}[g(x) - \eta Dg(x) \nabla f(x)]$$

is well defined around $x^*$ and $\Lambda(x^*) = \eta \lambda^*$ in view of the KKT conditions (16); since $\Lambda$ is continuous (because $g$, $Dg$ and $\nabla f$ are all continuous), for $x^{(k)}$ near $x^*$ we have $\lambda^{(k)} = \Lambda(x^{(k)}) \approx \Lambda(x^*)$, that is $\lambda^* \approx -\eta \lambda^{(k)}$.

Note that the constraints $g(x^{(k)})$ converge to zero faster than the iterates, see Remark 9 in Appendix B. This, together with Remark 10, implies that the distance between $x^{(k)}$ and the manifold $\mathcal{C}$ defined by the constraints converges to zero faster than the distance $\|x^{(k)} - x^*\|$.

It is interesting to observe that the proposed algorithm converges even on certain minimization problems which do not satisfy the hypotheses of Theorem 3.2. This is the case of Example 12.1 in [12], brought by the authors as an evidence that a good minimization algorithm should take into account the curvature of the level set defined by the constraints, that is, information from the second order derivatives of the constraints. Algorithm 1 shows good convergence on this example, although it uses information solely from the first derivatives.

3.3. Extension to inequality constraints. We now consider the problem

$$\min_{x \in \mathcal{C}} f(x), \quad \mathcal{C} = \{x \in \mathbb{R}^n : g_i(x) \leq 0\},$$

where the inequality is to be understood componentwise:

$$\mathcal{C} = \{x \in \mathbb{R}^n : g_i(x) \leq 0, 1 \leq i \leq m\}.$$

The necessary optimality conditions for this sort of problem are better expressed in terms of the active constraints at a solution $x^* \in \mathcal{C}$, that is, those constraints which attain equality:

$$\mathcal{A}^* = \{i \in \mathbb{N} : 1 \leq i \leq m, g_i(x^*) = 0\}.$$

The KKT conditions can then be written as follows:

$$\begin{cases}
\nabla f(x^*) + \sum_{i \in \mathcal{A}^*} \lambda_i^* \nabla g_i(x^*) = 0, \\
g_i(x^*) = 0, \quad i \in \mathcal{A}^*, \\
g_i(x^*) < 0, \quad i \notin \mathcal{A}^*, \\
\lambda_i^* > 0, \quad i \in \mathcal{A}^*, \\
\lambda_i^* = 0, \quad i \notin \mathcal{A}^*.
\end{cases}$$

These conditions hold necessarily at a solution $x^*$, provided the constraints satisfy a qualification condition; see e.g. [12, page 160], [11, Chapter 3] or [23, Chapter 12]. The first two equations are simply the optimality conditions for the equality constrained problem obtained by requiring the active constraints to be zero. The
third condition ensures that inactive constraints are satisfied. The last condition specifies that inactive constraints have null Lagrange multipliers attached; however, this condition is usually imposed in the KKT conditions for mere convenience; the values of those multipliers have no relevance whatsoever. The fourth condition is most important for practical purposes: Lagrange multipliers associated with active constraints must be nonnegative. This will be useful in order to decide when to deactivate constraints along the iterations.

We propose a generalization of Algorithm 1 which can handle inequality constraints. As in Algorithm 1, the iterates are not necessarily feasible; see, however, Subsection 3.4 for an exception. The strategy is based on the concept of active set; this means that, at each iteration, the constraints are partitioned in two separate groups. Those inequalities considered active will be treated much in the same manner as the equality constraints are treated in Algorithm 1. The inequalities considered inactive are essentially ignored. Obviously, the set of active indices is not constant along the optimization process. Activating and deactivating inequality constraints is the central (and difficult) point of Algorithm 2.

There is a large literature body on active set strategies, see e.g. [23, Chapter 18] or [21, Chapter 12]. In the proposed algorithm, an inequality is activated as soon as it is violated (step 3 in Algorithm 2). The deactivation criterion is not as straightforward. It is certainly not a good idea to deactivate a constraint as soon as it is fulfilled again (i.e., when the value of $g_i$ becomes negative again). Recall that an active inequality constraint is treated essentially as an equality constraint. Recall also that in our approach the constraints are not fulfilled along the optimization process (they are satisfied only in the limit). So, activating and deactivating an inequality constraint on the sole criterion of it being fulfilled or violated would often produce a zigzagging phenomenon (the same constraint being activated and deactivated repeatedly).

We propose that a constraint should be kept active as long as the process of minimization of $f$ has the tendency of violating that particular constraint. In order to measure this tendency, we use the sign of the respective Lagrange multiplier as a criterion. Lagrange multipliers associated to active constraints should be positive (see the above KKT conditions). Thus, we choose to deactivate a constraint when the associated Lagrange multiplier becomes negative (step 6 of Algorithm 2). To some extent, this procedure can be seen as a generalization of the simplex method to nonlinear functions. See [23, Section 16.5]) for a somewhat similar strategy; note that in [23] a distinction is made between active constraints and a working set of constraints, a terminology that we do not use.

The question arises as to what to do when more than one Lagrange multiplier becomes negative at the same iteration. Should we deactivate all the constraints corresponding to negative multipliers? Note that, if we deactivate one constraint, the remaining Lagrange multipliers should be computed again, and they may change signs. Should we deactivate only the constraint corresponding to the most negative multiplier? Does it make sense to compare the value of one Lagrange multiplier to another? In order to fix ideas, in Algorithm 2 we choose to deactivate the constraint corresponding to the most negative Lagrange multiplier, then compute again the remaining multipliers (steps 5 and 6). A more detailed discussion of the deactivation criterion is postponed to Subsection 3.5.

**Algorithm 2 (for inequality constraints).**

**INPUT:** initial guess $x^{(0)}$, step size $\eta > 0$, tolerance $\varepsilon > 0$, maximum number of
iterations $N$.
OUTPUT: approximate solution $x$ or message of failure.

**Step 1** Set $A = \emptyset$. *(no active constraints)*

**Step 2** With $k$ from 1 to $N$, do Steps 3–9.

**Step 3** With $i$ from 1 to $m$, do

If $g_i(x^{(0)}) > 0$ then set $A = A \cup \{i\}$.
*(constraint $g_i \leq 0$ is being violated, thus we set it active)*

**Step 4** Compute $\lambda_j (j \in A)$ by solving

$$
\sum_{j \in A} \lambda_j \langle \nabla g_i(x^{(0)}), \nabla g_j(x^{(0)}) \rangle = g_i(x^{(0)}) - \eta \langle \nabla g_i(x^{(0)}), \nabla f(x^{(0)}) \rangle, i \in A.
$$

**Step 5** Set $i = \arg \min_{j \in A} \lambda_j$.

**Step 6** If $\lambda_i < 0$ then set $A = A \setminus \{i\}$; *(constraint $g_i \leq 0$ is set inactive)*

GOTO Step 4.

**Step 7** Set $x = x^{(0)} - \eta \nabla f(x^{(0)}) - \sum_{i \in A} \lambda_i \nabla g_i(x^{(0)})$.

**Step 8** If $\|x - x^{(0)}\| < \varepsilon$ then OUTPUT($x$);

STOP.

**Step 9** Set $x^{(0)} = x$.

**Step 10** OUTPUT(‘The method failed after $N$ iterations.’);

STOP.

Convergence proofs for such methods usually assume some idealized procedure that is hardly employed in practice. We prefer not to state any kind of convergence result. In general, convergence cannot be guaranteed and zigzagging can sometimes occur, although experience shows it to be a rare phenomenon.

Step 4 of Algorithm 2 can become very heavy if many constraints are active (and thus many Lagrange multipliers must be computed). Note that the number of active constraints cannot exceed the number of variables, so this can only happen for a large number of variables. In Subsection 3.4 below, we describe how the computational burden associated with many active constraints can be significantly alleviated in a particular case.

Finally, let us note that it is not difficult to combine Algorithms 1 and 2 in order to treat the case when equality constraints are present together with inequality constraints. Simply, the equality constraints should always be kept active.

### 3.4. The case of box-like constraints.

We now turn our attention to constraints of the simple form $g_i(x) = a_i - x_i$ or $g_i(x) = x_i - b_i$; they confine the vector variable $x$ to a rectangular box in $\mathbb{R}^n$. Due to their particular form, constraints of this type deserve a special treatment. First, it is very easy to make a projection for such inequalities. So, they can be treated as essential constraints by performing a projection as soon as they are violated (in step 3), that is, by setting $x_i = a_i$ or $x_i = b_i$ respectively.

Second, their gradient has only one nonzero component (in the variable $x_i$). Because of this, the corresponding Lagrange multipliers can be eliminated from the linear system in step 4 of Algorithm 2. Thus, it suffices to compute the other Lagrange multipliers (if any) by solving a reduced system of linear equations. Then, the Lagrange multipliers associated to box-like constraints can be recovered one by one without computational effort. This process is explained in some detail in Appendix A.

Based on the above considerations, Algorithm 2 can be reformulated as Algorithm 3. We denote by $\mathcal{B}$ the set of box-like constraints. The symbol $\nabla^*$ denotes the gradient with respect to the “free” variables only, i.e. ruling out the, let’s say
Algorithm 3 (for box-like constraints).

**INPUT:** initial guess \( x^{(0)} \), step size \( \eta > 0 \), tolerance \( \varepsilon > 0 \), maximum number of iterations \( N \).

**OUTPUT:** approximate solution \( x \) or message of failure.

**Step 1** Set \( A = \emptyset \). (no active constraints)

**Step 2** With \( k \) from 1 to \( N \), do Steps 3–10.

**Step 3** With \( i \) from 1 to \( m \), do

- If \( g_i(x^{(0)}) > 0 \) then set \( A = A \cup \{i\} \). (constraint \( g_i \leq 0 \) is being violated, thus we set it active)

- If \( i \in B \) then set \( x_i^{(0)} = a_i \) or \( b_i \); (we project)

**Step 4** Compute \( \lambda_j \) (\( j \in A \setminus B \)) by solving

\[
\sum_{j \in A \setminus B} \lambda_j \langle \nabla g_i(x^{(0)}), \nabla g_j(x^{(0)}) \rangle = g_i(x^{(0)}) - \eta \langle \nabla^* g_i(x^{(0)}), \nabla^* f(x^{(0)}) \rangle, \quad i \in A \setminus B.
\]

**Step 5** Compute \( \lambda_j \) (\( j \in A \cap B \)) as

\( \lambda_j = -\eta f_{i,j}(x^{(0)}) - \sum_{k \in A \setminus B} \lambda_k g_{k,j}(x^{(0)}) \)

**Step 6** Set \( i = \arg \min_{j \in A} \lambda_j \).

**Step 7** If \( \lambda_i < 0 \) then set \( A = A \setminus \{i\} \); (constraint \( g_i \leq 0 \) is set inactive)

GOTO Step 4.

**Step 8** Set \( x = x^{(0)} - \eta \nabla f(x^{(0)}) - \sum_{i \in A} \lambda_i \nabla g_i(x^{(0)}) \).

**Step 9** If \( \|x - x^{(0)}\| < \varepsilon \) then OUTPUT(\( x \));

STOP.

**Step 10** Set \( x^{(0)} = x \).

**Step 11** OUTPUT(‘The method failed after \( N \) iterations.’);

STOP.

Observe the tricky detail that “blocked” variables \( x_i \) with \( i \in A \cap B \) should be ignored when computing the scalar products between gradients in step 4 of Algorithm 3; they should also be left unchanged in step 8 since a projection has been performed previously (in step 3). In a word, those “blocked” variables must be treated as if they were no longer variables but mere parameters, equal to \( a_i \) or to \( b_i \). However, Lagrange multipliers associated to constraints in \( A \cap B \), computed in step 5, are meaningful (and are used in steps 6 and 7 to decide deactivation).

The technique described above has been used to produce the numerical results in [10].

3.5. **Activation and deactivation strategies.** Different criteria for deactivating constraints (steps 5 and 6 of Algorithm 2, steps 6 and 7 of Algorithm 3) may be considered when more than one Lagrange multiplier becomes negative at a certain step of the algorithm. For instance, one could deactivate at once all the constraints with negative multipliers instead of deactivating only the most negative one.

In our opinion, two different situations should be distinguished. In the first one, which we shall describe as *discrete constraints*, there is a relatively small number of inequality constraints. These constraints may be very different of each other. They may have different physical nature, perhaps different physical units and different orders of magnitude. It makes no sense to compare their values, and it makes no sense to compare their associated Lagrange multipliers. Thus, there is no point in choosing the “most negative” multiplier, as done in step 5 of Algorithm 2 (step 6 of Algorithm 3). We consider that in this case at most one Lagrange multiplier should become negative at each step of the algorithm. The event of more than
one Lagrange multiplier becoming negative at a certain step should be interpreted as a warning that the optimization process is going too fast. Perhaps the value of the parameter \( \eta \) should be decreased. Thus, in this case, steps 5 and 6 of the Algorithm 2 (steps 6 and 7 of Algorithm 3) should be reformulated in order to test whether more than one multiplier is negative, and to take appropriate measures if this happens.

Actually, the above considerations apply also to the activation of constraints. If more than one constraint is violated at one step of the algorithm, this again should be viewed as a warning that the optimization process is going too fast.

In the second situation, there are many inequality constraints, very similar to each other. They share the same physical units and have the same order of magnitude. In a word, they are comparable. We shall call such a set of constraints an *almost-continuum of constraints* because this may appear, for instance, as a discretization of the continuum case described in Subsection 3.6. Note that this is not the same as the situation described in Subsection 3.4, where there are many constraints but they are very different from each other (they constrain different variables \( x_i \)).

In the case of an almost-continuum of constraints, even before discussing the deactivation strategy, we should take a look at how activation is done, that is, at how constraints are violated. Since there are many constraints, close to each other, if one of them is violated we expect the “neighbour” ones, that is, ones which are similar to it, to be violated too. So, “group violations” are to be expected in this second situation. If this happens, there is no point in activating the whole group of constraints. We propose that only the “worst offender” should be activated, that is, the one which becomes more positive (recall that the constraints are of the form \( g_i \leq 0 \)). In subsequent steps, the algorithm will push the value of that “worse” \( g_i \) towards zero, and this will have the side effect of pushing also its “neighbour” constraints towards zero.

It is not an easy task to implement the above described ideas into a computer program. Clusters of violated constraints must be identified and monitored along the optimization process. Each cluster should have a “leader” (the most violated constraint in that cluster) which is active. By pushing the “leader” towards zero one hopes to control the behaviour of the whole cluster. Along the optimization process, clusters may merge or split, which makes this programming task really challenging.

The implementation of the above ideas is only possible if the (almost-continuous) set of constraints has some internal organization which allows us to identify the closest neighbours of a given constraint, like the one described in Remark 5 for instance. Using this notion of vicinity, we suggest the following activation strategy. Among the set of all violated constraints, one should only activate those which are “more violated” than all of its neighbours (these will be the “leaders”). Along the optimization process, one should keep checking if the neighbours of the “leaders” become “more violated” than the “leader” and switch the activation flag towards the “most violated” one.

Going back to the deactivation issue, we see that the case of an almost-continuum of constraints is actually not very different from the case of discrete constraints, since only the “leaders” of the clusters of constraints have been activated. Thus, the active constraints continue to be few and “far” from each other (that is, different), and the same deactivation strategy should be employed as in the first situation.
Finally, note that a combination of the two situations (discrete constraints and almost-continuum constraints) may appear in some examples. It is not difficult to adapt the algorithm in order to deal with such problems. Even several almost-continua of constraints can be treated in the same manner.

3.6. A continuum of constraints. A true continuum of constraints may also be of interest for certain problems. In the sequel, we shall use the notation $g_\xi$ instead of $g_i$, $\xi$ being a parameter indexing the family of constraints.

For instance, in the example studied in Section 2 it would be interesting to allow for any angle between 0 and 180 degrees (this would give rise to a one-dimensional continuum of constraints). Another example is the optimization of a structure subject to an incoming acoustic or electromagnetic wave; the wave may come from any direction of the plane and may have any frequency within a certain range (this would be a two-dimensional continuum of constraints). Often, the optimization of a structure subject to multiple loads also falls under this category.

This case of a continuum of constraints can be dealt with, more or less in the same manner as described in Subsections 3.3 and 3.5. The main novelty is that we should now actively seek for the worst case (the most violated constraint) within a continuum, and this should be done by means of a maximization algorithm. This may be viewed as a dual approach: on one hand we minimize (in $x$) a function $f(x)$, while on the other hand we look for the worst case by maximizing (in $\xi$) $g_\xi(x)$.

Note that we have not implemented these ideas yet; the results in Section 4 have been obtained using the discrete approach described in Subsections 3.3 and 3.5.

3.7. Extension to minimax problems. We now turn to the problem of minimizing simultaneously a family of functionals. More precisely, the goal is to minimize the maximum of several different functionals:

$$\min_x \max \{f_1(x), f_2(x), \ldots, f_m(x)\},$$

where all of the $f_i$ are smooth functions. Instead of trying to apply a minimization algorithm to the nonsmooth inner max-function, we apply a classical technique in order to bypass this nonsmoothness (see e.g. [35]). The problem is written into a form appropriate for treatment by the method described in Subsections 3.3 and 3.5 by introducing a new (dummy) variable $z$ and then minimizing (in $x$ and $z$) the function $F(x, z) = z$ subject to the constraints $f_i(x) \leq z$, $i = 1, 2, \ldots, m$. See Section 4, namely Figure 4, for a flowchart of this algorithm in the specific case of designing auxetic materials.

The case of a continuous minimax problem can be dealt with in the same way. The problem

$$\min_{x, \xi} \max f_\xi(x)$$

can be reformulated as the minimization of $F(x, z) = z$ subject to $f_\xi(x) \leq z$, $\forall \xi$. Thus, the method described in Subsection 3.6 can be applied.

4. Numerical implementation and results. The minimax algorithm proposed in Subsection 3.7 is applied to the problem described in Section 2. The family of functionals to minimize simultaneously consists of all Poisson ratios associated to the effective elasticity tensor $C^H$ in many different directions of the plane. Specifically, we consider a (large but finite) set of directions $\vec{v}$ in the plane and for each direction we associate the respective Poisson ratio $\nu_\vec{v}$ defined by (11), (9) and (10). Each Poisson ratio $\nu_\vec{v}$ is a functional of $D^H$, which is the inverse tensor of $C^H$, which
in turn is a function of the shape and topology of the perforations denoted by $T$ in Subsection 2.1. The respective derivatives of these dependencies are given by formulas (8) and (12).

![Algorithm Flowchart](image-url)

**Figure 4. Algorithm**

Figure 4 shows a flow-chart of the minimax algorithm. The objective functionals are denoted by $\nu_i$ (they are the Poisson ratios of the microstructure along different
directions of the plane). The design variables are denoted by $x$; they define the shape of the microscopic hole(s). There is also a dummy variable $z$ (its role is explained in Subsection 3.7).

The algorithm will minimize the largest Poisson ratios; this means that, after convergence, the effective elastic tensor thus obtained will have roughly the same Poisson ratio in all directions of the plane; also, if the process is successful, this Poisson ratio will be negative. Note, however, that this does not ensure that $C^H$ is isotropic.

**Remark 5.** We use a (finite) family of directions $\vec{v}$ indexed by an angle varying from $0^\circ$ to $180^\circ$. Using the terminology from Subsection 3.5, this is an almost-continuous family of functionals organized as a ring (0 degrees actually gives the same direction as 180 degrees). Within this organization, each constraint has two closest neighbours.

We use our homemade code whose parts treating the cellular problem and the shape and topology derivatives have been described in [8, Sections 4 and 5]. As explained there, in order to discretize problem (2), the microstructure is meshed with triangular finite elements of Lagrange type of degree two. Some of the triangles are marked as “full”, corresponding to the elastic material, while other triangles are marked as “empty”, corresponding to the hole $T$. The interface between material and hole is marked in red (see Figures 3, 5 and 7).

**Remark 6.** Although for graphical purposes several contiguous cells are represented, the mesh covers only one cell $Y$ and is “closed” in itself, having no boundary. It can be described as a mesh on the two-dimensional torus. The graphical representation in Figures 3, 5 and 7 refers to an unfolded mesh where vertices, segments and triangles are drawn more than once.

**Remark 7.** Triangles drawn in light gray, representing the hole, play no role in the finite element analysis. We use no ersatz material. These triangles exist only because the part of the code dealing with meshing and remeshing treats in a unified manner the case of mixtures between two or more materials and the (present) case of perforations.

In order to implement the periodicity condition in (2), linear+periodic functions are considered on this mesh (they can be identified with multifunctions on the torus). This is done by keeping track of segments crossing the boundary of the cell $Y$ and by taking into account the jump of the function along those segments.

Along the optimization process, the mesh deforms in order for the holes to change their shape. The deformation of the mesh is accomplished by simply moving the vertices. However, this implies a gradual loss in the quality of the mesh: sharp angles eventually appear, as well as too long or too short segments. At some point, certain triangles may even become flat or reversed. Of course this must be prevented, since it turns the process of solving problem (2) by the finite element method ill-posed and consequently unstable. With this end in view, the program improves frequently the quality of the mesh, either by moving the vertices (equilibrating the mesh) or by changing the elements of the mesh (flipping segments, adding/eliminating vertices).

For shape optimization, the integrands in (6) are computed. These are scalar functions defined on the boundary of the holes and depending on the solutions $u_{f_j}$ of three cellular problems ($j = 1, 2, 3$). A functional $J$ is chosen which depends on the homogenized coefficients $C^H_{ij}$ (as explained in Subsection 2.3, here we actually
consider several functionals to be minimized simultaneously: the Poisson ratios). The shape derivative of $J$ is computed as

$$D_S J = \sum_{ij} \frac{\partial J}{\partial C_{ij}} D_S C_{ij} = \int_{\partial T} \gamma \vec{\theta} \cdot \vec{n}$$

where $\gamma$ is a scalar function computed as a linear combination of the integrands in (6). If a steepest descent method were used, one should choose a deformation of the interface $\partial T$ equal to $\vec{\theta} = -\gamma \vec{n}$ (multiplied by some positive constant $\eta$ which controls the speed of the process) in order to decrease the value of $J$. Here, we use instead the method described in Subsection 3.7 in order to minimize simultaneously several functionals, and the shape derivative of each of the Poisson ratios is used in Algorithm 3 accordingly. The algorithm provides a desired deformation of the interface $\partial T$, which is then propagated into the whole mesh by means of an averaging process.

Topology optimization can be performed by simply changing triangles, neighbour to a certain vertex, from material to void. However, in the present paper we focus only on shape optimization, which means that we begin the optimization process with a certain number of holes in the periodicity cell, and this number is going to remain constant up to the end of the optimization process.

In the first example we optimize the Poisson ratios along ten directions in the plane, at angles uniformly distributed between 0 and 180 degrees. The largest Poisson ratio among those ten directions is minimized. The initial microstructure (initial guess), see Figure 5, has a square periodicity, property that does not vary during the optimization process. It presents two model holes that repeat periodically. The algorithm, after 54 iterations, produced the microstructure on the right in Figure 5, still with square periodicity and still presenting two model holes. At iteration 47 the Poisson ratios became negative and the final design presents Poisson ratios less then -0.7. In the history of convergence, see Figure 6, one can observe that in the first iterations the algorithm makes the ten Poisson ratios as close as possible and afterwards it decreases them all together.

**Figure 5.** Initial guess and final microstructure for square periodicity
In the second example, eighteen directions in the plane are chosen, at angles uniformly distributed between 0 and 180 degrees. The largest Poisson ratio among those directions is minimized. The initial guess has a hexagonal periodicity and presents one model hole repeated periodically (with respect to the hexagonal periodicity), see Figure 7. After 60 iterations the algorithm produces the microstructure presented in Figure 7, on the right. The history of convergence is shown in Figure 8, with two zoom-in views. The final design has Poisson ratios close to -0.9 in all eighteen directions.
5. Conclusions and future development. Optimization of periodic microstructures is performed in order to obtain homogenized elastic tensors with negative Poisson ratio, by varying the shape and/or topology of the model hole characterizing the microstructure. Worst case design is employed in order to minimize the Poisson ratio of the homogenized elastic tensor in several prescribed directions of the plane, in order to obtain a material having roughly the same negative Poisson ratio in all directions. This approach may be generalized in order to minimize the largest Poisson ratio of the homogenized elastic tensor among all possible directions of the plane.

It is possible to obtain periodic 2D microstructures with Poisson ratio close to $-1$ and roughly the same in all directions of the plane. The examples show that the algorithm tends to “cut” the structure, so the bulk modulus approaches zero.

An algorithm for optimization with equality constraints is presented and proven to be convergent. It can be regarded as a gradient method applied in the direction tangent to the manifold determined by the constraints, together with a Newton method applied in the orthogonal direction. This method, although not very fast (it has linear convergence) is quite natural, easy to implement, and has the advantage of requiring solely the first derivatives of the objective and of the constraint functions – often a decisive feature in the framework of structural optimization.

Generalizations of this algorithm are described in order to address inequality constrained problems also, as well as minimax problems. Criteria for activation and deactivation of constraints are discussed in some detail, as well as the special case of box-like constraints. An approach to solve problems involving a (continuous) infinite family of inequality constraints is also discussed.

Future work includes the study of three-dimensional microstructures, the treatment of a continuum of constraints and its extension to continuous minimax problems. When put in the context of optimization of microstructures, the capacity to
deal with a continuum of constraints would allow one to impose as a constraint that the composite material be truly isotropic.
Appendix A. Box-like constraint multipliers. We now resume the course of action postponed in Subsection 3.4.

Suppose there are \( m_1 + m_2 \) active constraints; suppose that the first \( m_1 \) of them are of box-type, either of the form \( x_i \geq a_i \) or \( x_i \leq b_i \). Note that this implies that each of these box-type constraints corresponds to a certain variable; of course it is impossible for both \( x_i \geq a_i \) and \( x_i \leq b_i \) to become active simultaneously for the same variable \( x_i \).

The system of linear equations defining the Lagrange multipliers \( \lambda_j \) in step 4 of Algorithm 2 writes

\[
\sum_{j=1}^{m_1+m_2} \lambda_j \langle \nabla g_i, \nabla g_j \rangle = g_i - \eta \langle \nabla g_i, \nabla f \rangle, \quad 1 \leq i \leq m_1 + m_2
\]

We treat differently the first \( m_1 \) unknowns and the first \( m_1 \) equations:

\[
\sum_{j=1}^{m_1} \lambda_j \langle \nabla g_i, \nabla g_j \rangle + \sum_{k=m_1+1}^{m_1+m_2} \lambda_k \langle \nabla g_i, \nabla g_k \rangle = g_i - \eta \langle \nabla g_i, \nabla f \rangle, \quad 1 \leq i \leq m_1;
\]

\[
\sum_{j=1}^{m_1} \lambda_j \langle \nabla g_l, \nabla g_j \rangle + \sum_{k=m_1+1}^{m_1+m_2} \lambda_k \langle \nabla g_l, \nabla g_k \rangle = g_l - \eta \langle \nabla g_l, \nabla f \rangle, \quad m_1 + 1 \leq l \leq m_1 + m_2.
\]

Since \( 1 \leq i \leq m_1 \), we have that \( \nabla g_i = \pm e_i \), \( e_i \) being the canonical basis in \( \mathbb{R}^n \). To fix ideas, we suppose that \( \nabla g_i = e_i \); since \( 1 \leq j \leq m_1 \), we have \( \nabla g_j = e_j \). Also, note that \( g_i = 0 \) (when the box-like constraints are activated, a projection operation is performed, thus they are satisfied exactly). Hence, the linear system writes

\[
\lambda_i + \sum_{k=m_1+1}^{m_1+m_2} \lambda_k g_{k,i} = -\eta f_{i,i}, \quad 1 \leq i \leq m_1
\]

\[
\sum_{j=1}^{m_1} \lambda_j g_{l,j} + \sum_{k=m_1+1}^{m_1+m_2} \lambda_k \langle \nabla g_l, \nabla g_k \rangle = g_l - \eta \langle \nabla g_l, \nabla f \rangle, \quad m_1 + 1 \leq l \leq m_1 + m_2.
\]

By using the first \( m_1 \) equations, we easily express each \( \lambda_i \) in terms of \( \lambda_k \) \((m_1 + 1 \leq k \leq m_1 + m_2)\). By plugging these expressions into the second part of the system, we get

\[
-\sum_{j=1}^{m_1} \left( \sum_{k=m_1+1}^{m_1+m_2} \lambda_k g_{k,j} + \eta f_{j,j} \right) g_{l,j} + \sum_{k=m_1+1}^{m_1+m_2} \lambda_k \langle \nabla g_l, \nabla g_k \rangle = g_l - \eta \langle \nabla g_l, \nabla f \rangle, \quad m_1 + 1 \leq l \leq m_1 + m_2.
\]

and therefore

\[
\sum_{k=m_1+1}^{m_1+m_2} \lambda_k \langle \nabla g_l, \nabla g_k \rangle - \sum_{k=m_1+1}^{m_1+m_2} \sum_{j=1}^{m_1} \lambda_k g_{k,j} g_{l,j} = g_l - \eta \langle \nabla g_l, \nabla f \rangle + \sum_{j=1}^{m_1} \eta f_{j,j} g_{l,j}, \quad m_1 + 1 \leq l \leq m_1 + m_2.
\]

By expanding the inner products between gradients, we rewrite the above as

\[
\sum_{k=m_1+1}^{m_1+m_2} \sum_{j=m_1+1}^{n} \lambda_k g_{k,j} g_{j,j} = g_l - \eta \sum_{j=m_1+1}^{n} f_{j,j} g_{j,j}, \quad m_1 + 1 \leq l \leq m_1 + m_2
\]
Theorem B.1. The following result describes the procedure in detail. Furthermore, for $\eta > 0$, let $\{x \in \mathbb{R}^n : g(x) = 0\}$ be well defined and converge to a point $x^*$ such that $g(x^*) = 0$. This can be done, roughly speaking, by building a step orthogonal to the level set of $g$ and obeying within that orthogonal subspace, to the basic idea of the Newton method. Actually, this procedure can be used as a projection algorithm onto the manifold $C = \{x \in \mathbb{R}^n : g(x) = 0\}$. The following result describes the procedure in detail.

**Appendix B. Newton’s method for underdetermined systems.** Let $g : \mathbb{R}^n \to \mathbb{R}^m$, with $m \leq n$, be a nonlinear function and consider the problem of finding a root of $g$, that is, of solving the equation $g(x) = 0$. If $m < n$, this system of equations is underdetermined; the set of solutions will be a manifold $\mathcal{C}$ in $\mathbb{R}^n$ of dimension $n - m$. However, it may be of interest to solve numerically this system, that is, to start with some $x^{(0)} \in \mathbb{R}^n$ and to build iteratively a sequence $x^{(k)}$ which converges to some $x^*$ such that $g(x^*) = 0$. This can be done, roughly speaking, by building a step orthogonal to the level set of $g$ and obeying within that orthogonal subspace, to the basic idea of the Newton method. Actually, this procedure can be used as a projection algorithm onto the manifold $C = \{x \in \mathbb{R}^n : g(x) = 0\}$. The following result describes the procedure in detail.

**Theorem B.1.** Let $g : \mathbb{R}^n \to \mathbb{R}^m$ be differentiable, $Dg$ be of full rank $m$ in an open convex set $D$ and let the following hold:

(i) there exists $K \geq 0$ and $\alpha \in [0, 1]$ such that $\|Dg(y) - Dg(x)\| \leq K \|y - x\|^\alpha$ for all $x, y \in D$;

(ii) there is a constant $B$ for which $\|Dg(x)^+\| \leq B$ for all $x \in D$, where $Dg(x)^+ = \nabla g(x) [Dg(x) \nabla g(x)]^{-1}$ is the Moore-Penrose inverse of $Dg(x)$.

Furthermore, for $\eta > 0$, let $D\eta = \{x \in D : \|y - x\| < \eta \Rightarrow y \in D\}$. Then there exists an $\epsilon > 0$ depending only on $K$, $\alpha$, $B$ and $\eta$ such that if $x^{(0)} \in D\eta$ and $\|Dg(x^{(0)})\| < \epsilon$, then the iterates $x^{(k)}$ determined by

$$x^{(k+1)} = x^{(k)} - Dg(x^{(k)})^+ g(x^{(k)})$$

(19)

are well defined and converge to a point $x^* \in D$ such that $g(x^*) = 0$. Moreover, there is a constant $\beta$ for which

$$\|x^{(k+1)} - x^*\| \leq \beta \|x^{(k)} - x^*\|^{1+\alpha}, \quad k = 0, 1, 2, \ldots$$

Note that, if $m = n$, iteration (19) above reduces to the usual Newton iteration for determined systems of equations. For $m < n$, the rate of convergence is still quadratic if $Dg$ is Lipschitz continuous. Convergence theory on such Newton-like methods, for underdetermined systems of equations, can be found in [31] and references therein.

Since the considered system of equations has many solutions (a manifold of them), it is legitimate to question whether the sequence $x^{(k)}$ must be so rigidly defined. That is, one may ask whether the step $\delta^{(k)}$ must be necessarily orthogonal to the level set of $g$. The following result (whose proof is immediate) shows that the philosophy behind Newton’s method can be applied while allowing for uncertainties in the definition of the sequence $x^{(k)}$.

**Proposition 1.** Let $g : \mathbb{R}^n \to \mathbb{R}^m$ be continuously differentiable and consider a sequence $x^{(k)}$ which converges to some $x^* \in \mathbb{R}^n$ and which satisfies

$$Dg(x^{(k)}) \delta^{(k)} = -g(x^{(k)})$$

(20)
where $\delta^{(k)} = x^{(k+1)} - x^{(k)}$; in coordinate notation,

$$
\sum_{j=1}^{n} g_{i,j}(x^{(k)}) \delta^{(k)}_j = -g_i(x^{(k)}), \quad 1 \leq i \leq m.
$$

Then $x^*$ is a solution, that is, $g(x^*) = 0$.

Note that condition (19) implies (20). Note also that, if $m = n$, then (19) and (20) are equivalent. However, for $m < n$, unlike (19), equation (20) does not define uniquely the sequence $x^{(k)}$; it simply states a property of the sequence. The user has the freedom to choose $n - m$ components of $\delta^{(k)}$: those orthogonal to $\nabla g_1(x^{(k)})$, $\nabla g_2(x^{(k)})$, \ldots, $\nabla g_m(x^{(k)})$, that is, components which are tangent to the level set of $g$. The user may use this freedom in order to solve other equation(s) or to minimize some functional.

**Remark 9.** The quantity $g(x^{(k)})$ converges to zero faster than the convergence of $x^{(k)} \to x^*$. For instance, suppose $x^{(k)}$ converges linearly, meaning there is a positive constant $L < 1$ such that $\|x^{(k)} - x^*\|$ is of order $O(L^k)$. Then $\delta^{(k)} = x^{(k+1)} - x^{(k)}$ is also of order $O(L^k)$ and a simple Taylor expansion about $x^{(k)}$ yields

$$
g(x^{(k+1)}) = g(x^{(k)}) + Dg(x^{(k)}) \delta^{(k)} + O(\|\delta^{(k)}\|^2);
$$

thus, $g(x^{(k)})$ is of order $O(L^{2k})$. Note, however, that this is not quadratic convergence but simply an improved linear one. Nor should quadratic convergence be expected because, unlike in Theorem B.1, here the step $\delta^{(k)} = x^{(k+1)} - x^{(k)}$ is not a full Newton-type step.

**Remark 10.** Locally around a regular point $x^* \in \mathcal{C} = \{x \in \mathbb{R}^n : g(x) = 0\}$ (see Definition 3.1 in Subsection 3.2), it can be proven that the quantity $\|g(x^{(k)})\|$ is of the same order of magnitude as the distance $\text{dist}(x^{(k)}, \mathcal{C})$ of $x^{(k)}$ to the manifold $\mathcal{C}$. This geometric property, taken together with Remark 9 above, implies that $\text{dist}(x^{(k)}, \mathcal{C})$ converges to zero faster than the convergence of $x^{(k)} \to x^*$.

**REFERENCES**

[1] G. Allaire, E. Bonnetier, G. Francfort and F. Jouve, Shape optimization by the homogenization method, *Numer. Math.*, **76** (1997), 27–68.

[2] G. Allaire, *Shape Optimization by the Homogenization Method*, Applied Mathematical Sciences, 146, Springer-Verlag, New York, 2002.

[3] G. Allaire, F. Jouve and A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, *J. Comput. Phys.*, **194** (2004), 363–393.

[4] F. Feppon, G. Allaire and C. Dapogny, Null space gradient flows for constrained optimization with applications to shape optimization, preprint, 2019.

[5] K. Atkinson and W. Han, *Theoretical Numerical Analysis: A Functional Analysis Framework*, Texts in Applied Mathematics, 39, Springer, Dordrecht, 2009.

[6] C. Barbarosie, Shape optimization of periodic structures, *Computational Mechanics*, **30** (2003), 235–246.

[7] C. Barbarosie and A.-M. Toader, Shape and topology optimization for periodic problems, part I: The shape and the topological derivative, *Struct. Multidiscip. Optim.*, **40** (2010), 381–391.

[8] C. Barbarosie and A.-M. Toader, Shape and topology optimization for periodic problems, part II: Optimization algorithm and numerical examples, *Struct. Multidiscip. Optim.*, **40** (2010), 393–408.

[9] C. Barbarosie and S. Lopes, A gradient-type algorithm for optimization with constraints, preprint Pre-2011-001, available from http://cmaf.fc.ul.pt/preprints.html, 2011.

[10] C. Barbarosie and S. Lopes, A generalized notion of compliance, *Comptes Rendus Mécanique*, **339** (2011), 641–648.
[11] D. Bertsekas, *Nonlinear Programming*, 2nd edition, Athena Scientific Optimization and Computation Series, Athena Scientific, Belmont, MA 1999.

[12] J. Bonnans, J. Gilbert, C. Lemaréchal and C. Sagastizábal, *Numerical Optimization – Theoretical and Practical Aspects*, Universitext, Springer-Verlag, Berlin, 2003.

[13] A. Borei, R. Schmidt and O. Sidebottom, *Advanced Mechanics of Materials*, Wiley, 1993.

[14] P. W. Christensen and A. Klarbring, *An Introduction to Structural Optimization*, Solid Mechanics and Its Applications, Springer, New York, 2009.

[15] P. G. Ciarlet, *Introduction à l’Analyse Numérique Matricielle et à l’Optimisation*, Masson, Paris, 1990.

[16] R. Fletcher, *Practical Methods of Optimization*, Constrained optimization, John Wiley & Sons, Chichester, 2013.

[17] J. Haslinger and R. A. E. Mäkinen, *Introduction to Shape Optimization: Theory, Approximation and Computation*, Advances in Design and Control, 7, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003.

[18] A. Henrot and M. Pierre, *Shape Variation and Optimization: A Geometrical Analysis*, European Mathematical Society (EMS), Zürich, 2018.

[19] E. Kreyszig, *Introductory Functional Analysis with Applications*, John Wiley & Sons, 1978.

[20] T. C. Lim, *Axiomatic Materials and Structures*, Engineering Materials, Springer, 2015.

[21] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd edition, International Series in Operations Research & Management Science, 116, Springer, New York, 2008.

[22] G. Milton, *The Theory of Composites*, Cambridge Monographs on Applied and Computational Mathematics, 6, Cambridge University Press, Cambridge, 2002.

[23] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd edition, Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006.

[24] A. A. Novotny and J. Sokolowski, *Topological Derivatives in Shape Optimization*, Interaction of Mechanics and Mathematics, Springer, Heidelberg, 2013.

[25] A. A. Novotny, J. Sokolowski and A. Żochowski, *Applications of the Topological Derivative Method*, Studies in Systems, Decision and Control, 188, Springer, 2019.

[26] A. Rothwell, *Optimization Methods in Structural Design*, Solid Mechanics and its Applications, 242, Springer, 2017.

[27] W. R. Spillers and K. M. MacBain, *Structural Optimization*, in *Computers & Structures*, Springer, Dordrecht, 2009.

[28] C. Van Hooricks, O. Sigmund, M. Schevenels, B. S. Lazarov and G. Lombaert, *Topology optimization of two-dimensional elastic wave barriers*, Journal of Sound and Vibration, 376 (2016), 95–111.

[29] J. Sokolowski and J.P. Zolesio, *Introduction to Shape Optimization: Shape Sensitivity Analysis*, Springer Series in Computational Mathematics, 16, Springer-Verlag, Berlin, 1992.

[30] A.-M. Toader, *The topological derivative of homogenized elastic coefficients of periodic microstructures*, J. Control Optim., 49 (2011), 1607–1628.

[31] H. Walker and L. Watson, *Least-change secant update methods for underdetermined systems*, SIAM J. Numer. Anal., 27 (1990), 1227–1262.

[32] M. Wormser, F. Wein, M. Stingl and C. Kerker, *Design and additive manufacturing of 3D phononic band gap structures based on gradient based optimization*, Materials, 10 (2017).

[33] H. Yamashita, *A differential equation approach to nonlinear programming*, Math. Programming, 18 (1980), 155–168.

[34] Y. Yuan, *A review of trust region algorithms for optimization*, ICIAM, Oxford Univ. Press, Oxford, 2000, 271–282. Available from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.9964.

[35] Z. Zhu, X. Cai and J. Jian, *An improved SQP algorithm for solving minimax problems*, Appl. Math. Lett., 22 (2009), 464–469.

Received May 2019; revised June 2019.

E-mail address: cabarbarosie@fc.ul.pt
E-mail address: atoader@fc.ul.pt
E-mail address: slopes@adm.isel.pt