



LEVEL SET-BASED TOPOLOGY OPTIMIZATION USING NONLINEAR PROGRAMMING: A PRELIMINARY STUDY

Suzana L. C. Castro

suzana.castro@metrocamp.edu.br

Metrocamp

R. Dr. Sales de Oliveira, 1661, 13035-270, Campinas, SP, Brazil

Francisco A. M. Gomes

Sandra A. Santos

chico@ime.unicamp.br

sandra@ime.unicamp.br

University of Campinas

R. Sergio Buarque de Holanda, 651, 13083-859, Campinas, SP, Brazil

Abstract. *We investigate the numerical performance of a structural optimization method based on the level-set approach and using nonlinear programming. Besides the method of the moving asymptotes (MMA) and a globally convergent modification of it with a spectral correction, we have also considered sequential quadratic programming and interior-point methods as the nonlinear programming driving tools. We remark that the original formulation of the topology optimization problem may be addressed, without any need to penalize the constraints. The method was applied to the solution of some classical problems from the literature and the results obtained are encouraging.*

Keywords: *Nonlinear programming, Level-set methods, Topology optimization, Structural optimization*

1 INTRODUCTION

The level-set approach has been successfully used to address structural topology optimization. Recent reviews (Deaton and Grandhi (2014), Sigmund and Maute (2013)) include such an approach as an alternative to the traditional material distribution density-based methods. Level-set methods can be categorized with respect to the level-set-function parameterization, the geometry mapping, the physical/mechanical model, the information and the procedure to update the design and the applied regularization. Important issues that affect the convergence behavior of the optimization process in the level-set scenery include the control over the slope and smoothness of the level-set function as well as the hole nucleation. These ideas are fully discussed in the review article of Dijk et al. (2013).

Matlab codes are current available for level set-based topology optimization methods, that provide enlightening and concrete implementation details and were designed with educational purposes. The code of Challis (2010) minimizes the compliance of statically loaded structures. An evolution equation with an additional forcing term is used to update the level-set function, so that the nucleation of new holes within the structure is allowed, but the standard Hamilton-Jacobi evolution equation can be employed as well by setting the parameter of the forcing term to zero. The code of Otomori et al. (2015) also addresses the compliance minimization problem, but the level-set function is updated using a reaction-diffusion equation based on the topological derivative of the objective functional (see also Yamada et al. (2010)). The regularizing diffusive term ensures the smoothness of the level-set function and a relaxation in the volume constraint stabilizes the convergence. The finite element method (FEM) is used to solve the equilibrium equations and the reaction-diffusion equation when updating the level-set function.

Pursuing a direct steepest-descent update of the design variables in a level-set method – the level-set nodal values – Dijk et al. (2012) use an exact Heaviside formulation to relate the level-set function to element densities. As a result, they tackle the numerical consistency of the sensitivity analysis and propose alternative parameterizations to avoid artifacts in the final results. In their conclusions, they mention that preliminary tests combining the proposed method with the method of the moving asymptotes (MMA) of Svanberg (1987) show promising results.

Otomori et al. (2011) applied level-set based topology optimization to the design of compliant mechanisms. The level-set function is updated using mathematical programming, more specifically MMA, to facilitate the treatment of constraints. To assess its capability, the authors have applied their method to compliant mechanism design problems that include displacement constraints and stress constraints. It is worth mentioning that level set-based topology optimization methods are immune to the problem of grayscales since the boundaries of the optimal configuration are implicitly represented using the level set function. Therefore, within the level-set approach it is no longer necessary to circumvent the obtaining of structures with grayscale areas.

In this paper we also employ mathematical programming techniques for updating the level-set-function. We have prepared a numerical investigation of the performance of a structural optimization method based on the level-set approach in the solution of benchmarking compliance problems. The nonlinear programming driving tools are MMA in its original version; a globally convergent modification of MMA with a spectral correction (Gomes-Ruggiero et al. (2010)); the general purpose constrained minimization solver of Matlab (routine `fmincon`)

with either the sequential quadratic programming algorithm (cf. chapter 18 of Nocedal and Wright (2006)) or the interior-point algorithm (Byrd et al., 2000).

This work is organized as follows. In Section 2 we define the level-set function and the general structural topology optimization compliance problem, with all the necessary notation. The algorithmic framework adopted in our investigation is also established. Section 3 starts with the methodology employed to prepare our code, followed by a brief explanation about the nonlinear programming (NLP) methods used in the updating of the level-set function, together with the details and choices that define the set of test problems. It also contains the numerical results, depicted as images, with comparative data. In Section 4 we provide the final remarks and comments on our work in progress.

2 THE ALGORITHMIC FRAMEWORK

2.1 The level-set function and the general compliance problem

Categorized as a boundary variation method, with roots in shape optimization techniques, the level-set based methods rely on implicit functions that define the structural boundaries, instead of an explicit parameterization of the design domain. Indeed, the structural boundary is implicitly specified as a contour line of the field Φ , which is a function of x (see Fig. 1). Therefore,

$$\begin{cases} \Phi(x) > 0, & x \in \Omega \\ \Phi(x) = 0, & x \in \Gamma \\ \Phi(x) < 0, & x \notin \Omega, \end{cases}$$

where Ω is the domain of the structure and Γ is the boundary of such a domain.

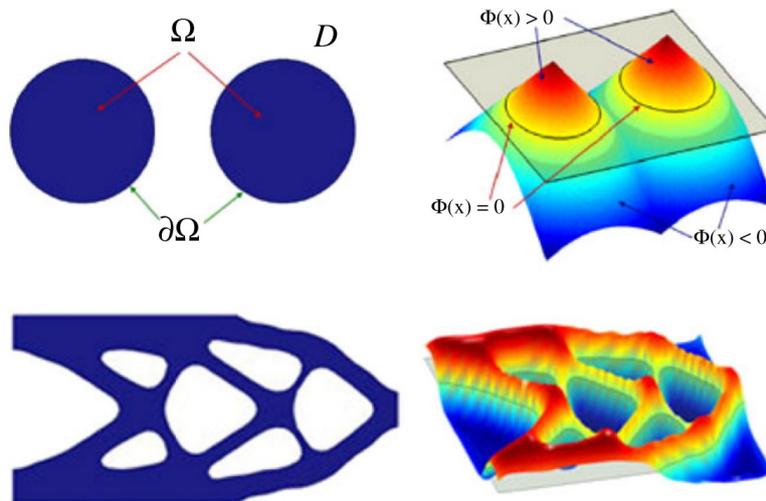


Figure 1: Level set representations: (a) 2D topology with (b) corresponding level set function along with a more complicated representation of benchmark structure (c) and (d), from Deaton and Grandhi (2014).

The level-set formulation is characterized by letting the level set model vary in time, i.e. $\{x(t) \mid \Phi(x(t), t) = 0\}$, taking its time derivative and applying the chain rule to obtain the

Hamilton-Jacobi-type equation

$$\frac{\partial \Phi(x, t)}{\partial t} + \nabla \Phi(x, t) \frac{dx}{dt} = 0, \quad \Phi(x, 0) = \Phi_0(x),$$

that defines an initial value problem for the time-dependent function Φ .

Consequently, the optimal structural boundary turns into the solution of a partial differential equation on Φ , in which $\frac{dx}{dt} \equiv \Upsilon(x, \Phi)$ is the velocity vector of the level set and depends on the desired objective to be optimized. An appropriate vector may be obtained as a descent direction of the objective via sensitivity analysis.

From this perspective and following Dunning and Kim (2013), a general minimum compliance optimization problem may be stated as

$$\begin{aligned} & \text{minimize} && \int_D E \varepsilon(u) \varepsilon(u) H(\Phi) d\Omega \equiv C(u, \Phi) \\ & \text{subject to} && \int_D H(\Phi) d\Omega \leq V_{\max} \\ & && \int_D E \varepsilon(u) \varepsilon(v) H(\Phi) d\Omega = \int_D b v H(\Phi) d\Omega + \int_{\Gamma_t} t v d\Gamma \\ & && u|_{\Gamma_u} = 0, \quad \forall v \in U \end{aligned} \tag{1}$$

where D is the design domain, Ω is the material domain, such that $\Omega \subset D$, V_{\max} is an upper bound on the material volume, E is the material property tensor, $\varepsilon(u)$ is the strain tensor for the displacement field u , U is the space of kinematically permissible displacement fields, v is any permissible displacement field, b are body forces, t are surface tractions and $H(\Phi)$ is the Heaviside function

$$H(\Phi) = \begin{cases} 1, & \Phi \geq 0 \\ 0, & \Phi < 0. \end{cases}$$

Figure 2 illustrates the fixed design domain D , the material domain Ω and boundary conditions for a mean compliance minimization problem.

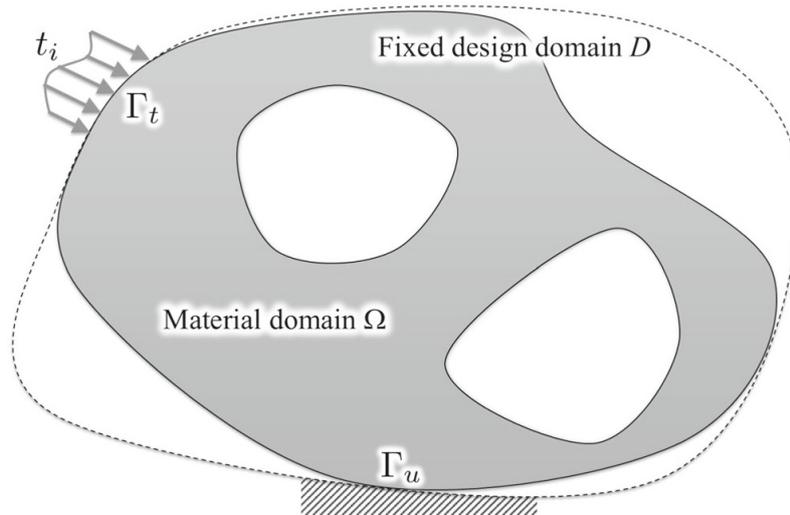


Figure 2: Design domain D , material domain Ω and boundary conditions, from Otomori et al. (2015).

2.2 The optimization scheme

The equilibrium equality constraints of problem (1) are evaluated by the FEM, together with the corresponding boundary conditions, and turned into $Ku = f$, where the stiffness matrix is given by

$$K = \sum_e \rho_e(\Phi) K_e$$

with

$$\rho_e(\Phi) = \varepsilon + (1 - \varepsilon) \frac{\int_{D_e} H(\Phi) d\Omega}{\int_{D_e} d\Omega},$$

where ε is a lower bound to avoid a singular structural problem, $H(\Phi)$ is the Heaviside function of the level-set function and D_e is the domain of an element. As a result, we reach the following discrete formulation (cf. Dijk et al. (2012)) for the compliance problem

$$\begin{aligned} \min_{\Phi} \quad & c(\Phi) \equiv u^T K u \\ \text{s.t.} \quad & \frac{1}{N_e} \sum_e \rho_e(\Phi) \leq V_{\max} \\ & K u = f \\ & -1 \leq \Phi \leq 1, \end{aligned} \tag{2}$$

where N_e is the number of elements of the discretization.

Schematically, the procedure adopted in this study for solving (2) is provided in Fig. 3. Our flow chart was motivated by those of Otomori et al. (2011) and Yamada et al. (2011).

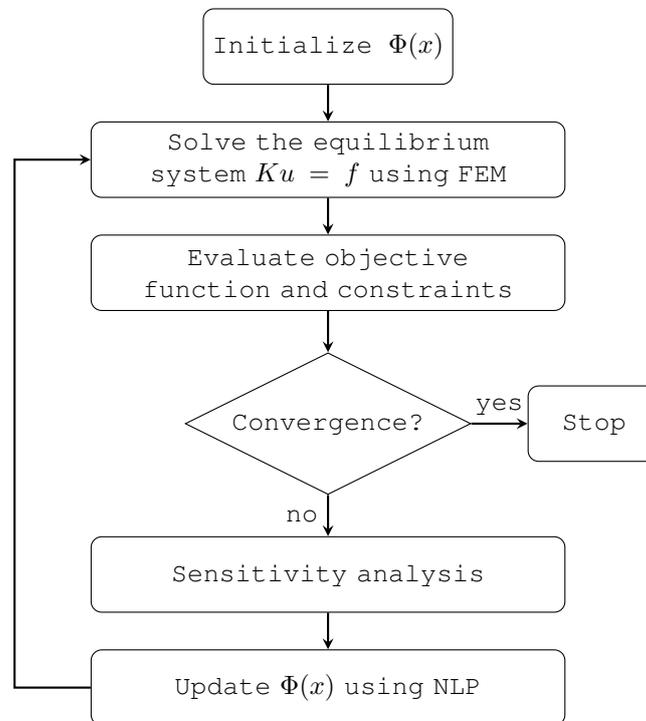


Figure 3: Flow chart of the optimization scheme.

3 NUMERICAL EXPERIMENTS

3.1 Methodology

We have prepared a Matlab implementation to address the solution of the problem (2). We have used an iMac, with a 2.4 GHz Intel Core 2 Duo processor and MatlabR2012a 64-bits. Our code rests upon elements from both Dijk et al. (2012) and Otomori et al. (2015). From the former, we have inherited the computation of the densities $\rho(\Phi)$, their derivatives, the perimeter of the structure and its derivative. From the latter, we have adopted the FEM computation (Andreassen et al. (2011)), together with the stopping criteria (budget on number of iterations and lack of progress). As suggested in Dijk et al. (2012), we have controlled the occurrence of gray areas by adding a perimeter penalty to the objective function, with penalty factor γ .

3.2 The NLP methods

We have chosen four nonlinear programming methods to update the level-set function: (i) the original version of MMA (Svanberg (1987), denoted by `MMAorig`); (ii) the modified version of MMA of Gomes-Ruggiero et. al (2010), that is based on the spectral parameter to update a key parameter of the model, so that the second-order information present in the spectral parameter is included in the model functions that define the rational approximations, denoted by `MMAspec`; (iii) the sequential quadratic programming algorithm, cf. Chapter 18 of Nocedal and Wright (2006), coded as the internal option `sqp` of the Matlab routine `fmincon`, denoted by `fmincon/sqp`; (iv) the interior-point algorithm (Byrd et al., 2000), coded as the option `interior-point` of `fmincon`, denoted by `fmincon/ip`.

For updating the level-set function $\Phi(x)$, instead of actually solving the current NLP problem, we let the solver to perform at most a pre-established and fixed number of iterations. As shown in the numerical results, this policy has generated good results.

3.3 Test problems

Figure 4 shows the fixed design domains and the boundary conditions for the test problems solved in this study. The first one models a cantilever beam whereas the second models a half wheel. Due to the symmetrical features of the second model, only its right half portion is computed, as suggested by Otomori et al. (2015), and this is the portion of the structure that is depicted in the results.

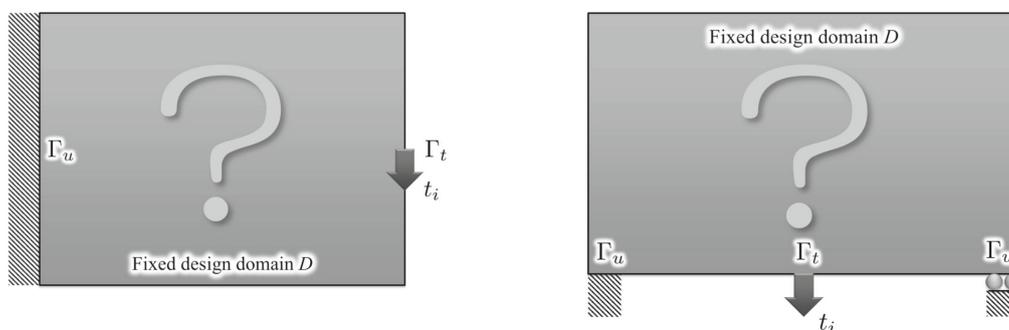


Figure 4: Design domains and boundary conditions for Problem 1 (left) and Problem 2 (right), from Otomori et al. (2015).

3.4 Results

The Young's modulus of the material was set as $E = 1$ and its Poisson's ratio was set to $\nu = 0.3$. Figure 5 depicts the initial configurations for both problems. We remark that the option `fmincon/sqp` is only able to address medium-scale problems, so the coarser meshes were used solely with this option of solver. Indeed, the dimensions of the squared stiffness matrices were as follows: 1680 for the 40×20 mesh, 6560 for the 80×40 mesh, 1890 for the 30×30 mesh and 7380 for the 60×60 mesh.

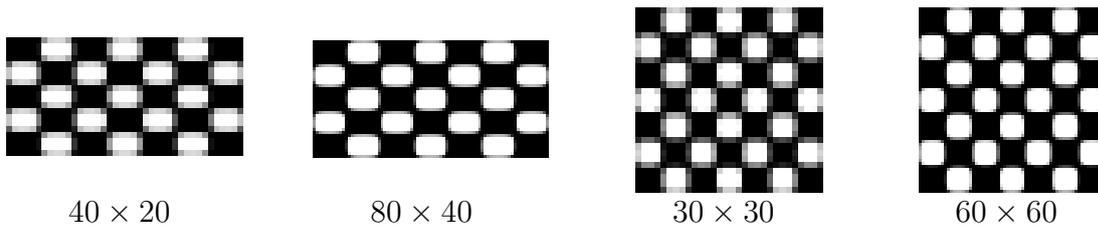


Figure 5: Initial configurations for Problem 1 (40×20 and 80×40 meshes) and Problem 2 (30×30 and 60×60 meshes)

When it comes to the stopping criteria, we proceed as Otomori et al. (2015) (see also Chalis (2010)), by reaching at most 0.5% of error in the volume constraint, i.e.,

$$\left| \frac{1}{N_e} \sum_e \rho_e(\Phi) - V_{\max} \right| \leq 0.005,$$

together with relative measure of lack of progress within the objective function value in the last 5 iterations, with tolerance ε_c , as follows

$$|c(\Phi_k) - c(\Phi_{k-p})| \leq \varepsilon_c |c(\Phi_k)|, \quad \forall p \in \{1, 2, \dots, 5\}. \quad (3)$$

We have also imposed a maximum number of 300 outer iterations for all the solvers. Nevertheless, in case `fmincon` produces an unexpected output, the code may exit without performing the budget of iterations.

To assess the influence of the perimeter control, we have run `fmincon/ip` for Problems 1 and 2, without such a control, as well as with $\gamma \in \{10^{-3}, 10^{-2}, 10^{-1}\}$ for two options for the maximum fraction of volume, namely $V_{\max} \in \{0.40, 0.50\}$.

The results can be seen at Figs. 6 and 7 for Problem 1 and at Figs. 8 and 9 for Problem 2. Based on the obtained results, we have decided to further investigate, for each problem and each solver, the effect of the choices $\gamma = 10^{-1}$ and $\gamma = 10^{-2}$.

Figure 10 depicts a study concerning the influence, in the final structure of Problem 1, of the number of performed iterations of the solver `MMAorig` in the updating of the level-set function Φ . Such a number will be addressed as the *inner iterations* of the corresponding NLP solver, whereas the *outer iterations* denotes the number of times the convergence criteria are checked.

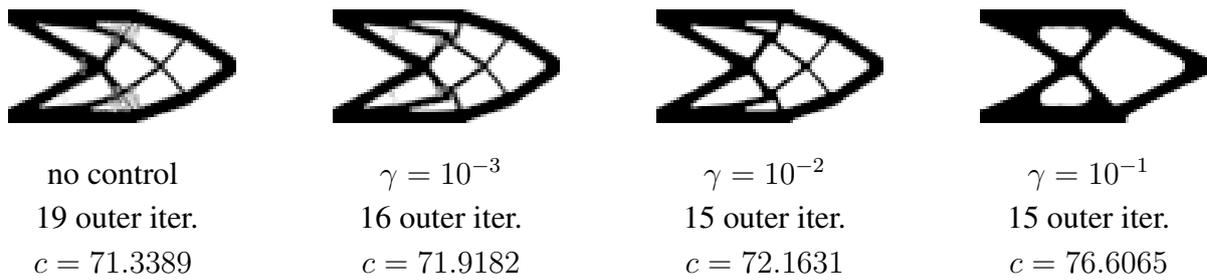


Figure 6: Solver `fmincon/ip`, Problem 1, 80×40 mesh, $V_{\max} = 0.40$, $\varepsilon_c = 10^{-2}$, 5 inner iterations.

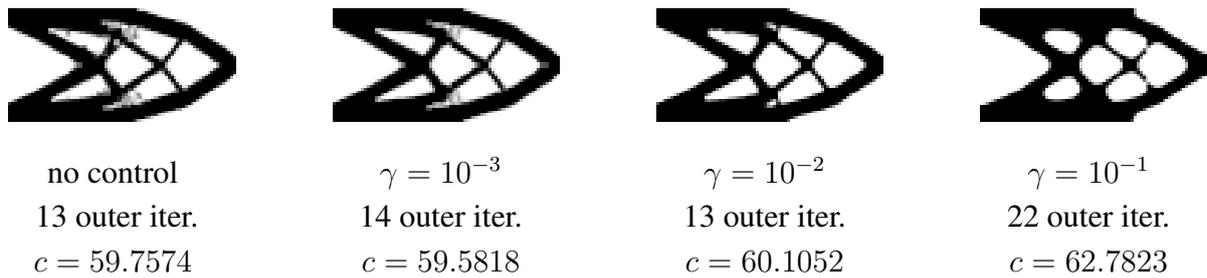


Figure 7: Solver `fmincon/ip`, Problem 1, 80×40 mesh, $V_{\max} = 0.50$, $\varepsilon_c = 10^{-2}$, 5 inner iterations.

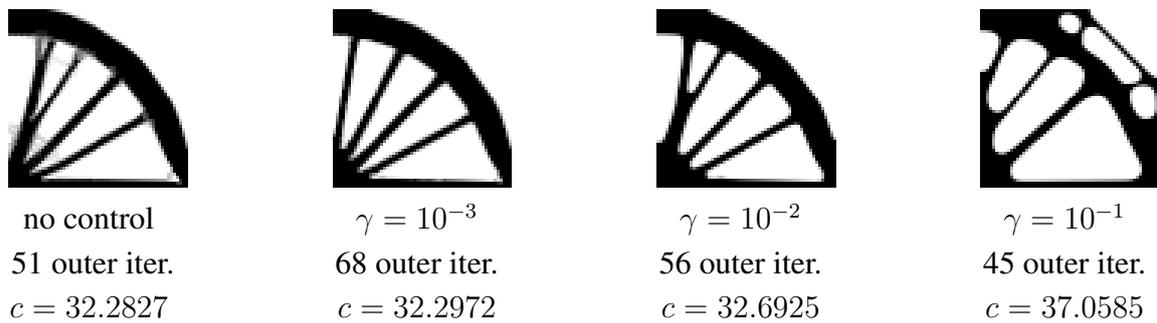


Figure 8: Solver `fmincon/ip`, Problem 2, 60×60 mesh, $V_{\max} = 0.40$, $\varepsilon_c = 10^{-3}$, 5 inner iterations.

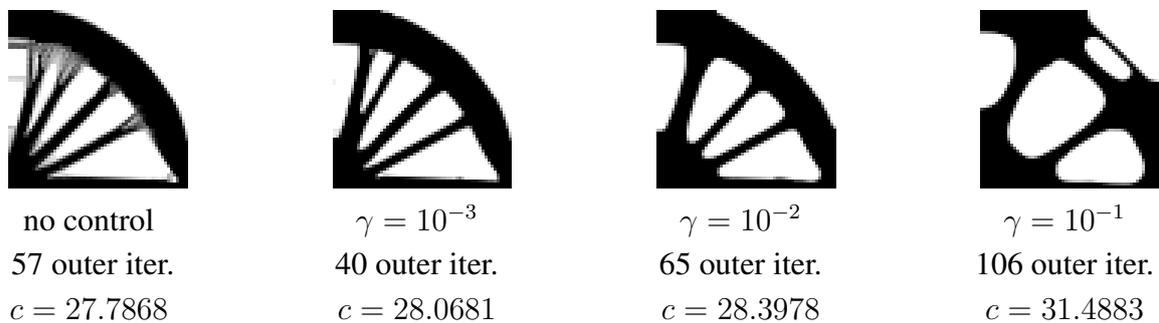


Figure 9: Solver `fmincon/ip`, Problem 2, 60×60 mesh, $V_{\max} = 0.50$, $\varepsilon_c = 10^{-3}$, 5 inner iterations.

$V_{\max} = 0.40$		$V_{\max} = 0.50$	
1 inner iteration			
			
21 outer iter. $\gamma = 10^{-1}$ $c = 76.5102$	20 outer iter. $\gamma = 10^{-2}$ $c = 74.5767$	42 outer iter. $\gamma = 10^{-1}$ $c = 63.6847$	22 outer iter. $\gamma = 10^{-2}$ $c = 60.4948$
2 inner iterations			
			
14 outer iter. $\gamma = 10^{-1}$ $c = 76.1292$	17 outer iter. $\gamma = 10^{-2}$ $c = 73.3600$	22 outer iter. $\gamma = 10^{-1}$ $c = 63.6309$	15 outer iter. $\gamma = 10^{-2}$ $c = 60.1998$
3 inner iterations			
			
13 outer iter. $\gamma = 10^{-1}$ $c = 76.0803$	13 outer iter. $\gamma = 10^{-2}$ $c = 72.1767$	12 outer iter. $\gamma = 10^{-1}$ $c = 62.3294$	15 outer iter. $\gamma = 10^{-2}$ $c = 60.0650$
4 inner iterations			
			
15 outer iter. $\gamma = 10^{-1}$ $c = 75.2610$	13 outer iter. $\gamma = 10^{-2}$ $c = 71.9319$	13 outer iter. $\gamma = 10^{-1}$ $c = 62.2747$	11 outer iter. $\gamma = 10^{-2}$ $c = 60.0617$

Figure 10: Solver MMA_{orig}, Problem 1, 80×40 mesh, $\varepsilon_c = 10^{-2}$.

Based on these results, we decided to adopt 4 inner iterations for the subsequent tests involving MMA_{orig}. A similar study has been prepared for the other solvers, and 4 inner iterations turned to be the best choice for MMA_{spec} as well, whereas for fmincon/ip and fmincon/sqp the most favourable results were obtained with 5 inner iterations.

It is worth mentioning that such an idea came in hand with the evolutionary approaches based on partial differential equations (eg. Challis (2010), Otomori et. al (2015)), in which steepest descent steps for the augmented Lagrangian are taken to update the level-set function (see also Dijk et al. (2012)). Consequently, our updating for Φ also has a local and *inexact* flavor, and it is obtained upon the corresponding sufficient decrease condition of the adopted solver.

Figure 11 summarizes the results of the solver `fmincon/ip` for Problem 1, for two choices of the percentage of volume of material as well as two options for the penalty factor γ .

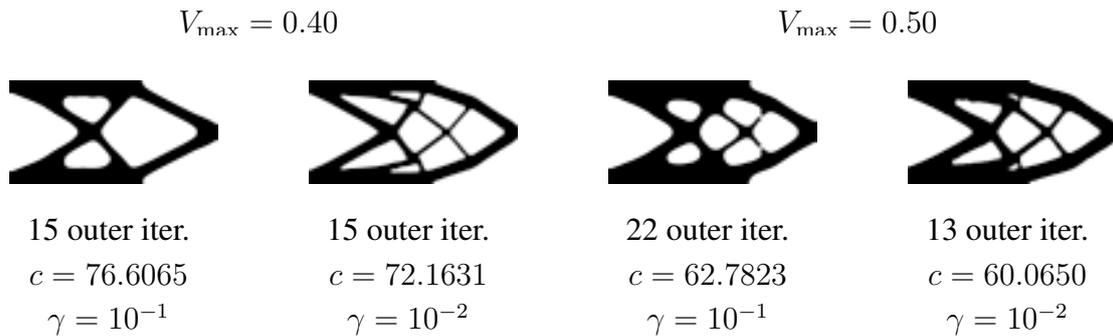


Figure 11: Solver `fmincon/ip`, Problem 1, 80×40 mesh, $\varepsilon_c = 10^{-2}$, 5 inner iterations.

In Fig. 12 we can observe the effect of tightening the tolerance of the relative stopping criterion of Eq. (3), with an improvement in the compliance at the expense of additional effort.

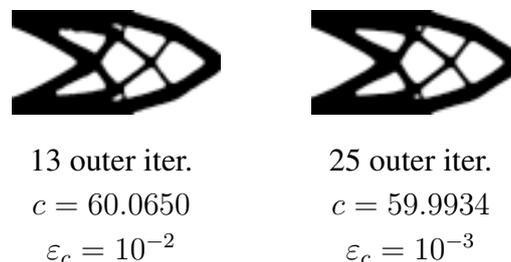


Figure 12: Solver `fmincon/ip`, Problem 1, 80×40 mesh, $V_{\max} = 0.50$, $\gamma = 10^{-2}$, 5 inner iterations.

The results of the solver `fmincon/sqp` for Problem 1 are depicted in Fig. 13, being qualitatively very much alike those of Fig. 11.

Concerning the solver `MMAspec`, the results of Problem 1 are shown in Fig. 14. After tightening the tolerance ε_c , we have refined a poor result of an instance, as can be seen in Fig. 15.

When it comes to Problem 2, although the outcomes of the solver `fmincon/ip` have already been presented along the analysis of the influence of the perimeter control parameter (Figs. 8 and 9), they are compiled in Fig 16 for the sake of better visualization and comparative purposes. The results are similar for those obtained for the solver `fmincon/sqp`, shown in Fig. 17, but only for the choice $\gamma = 10^{-2}$.

For the solver `MMAorig`, as can be seen in Fig. 18, the instances with $\gamma = 10^{-1}$ reached the budget of maximum iterations, stopping with a poorly attendance of the volume constraint.

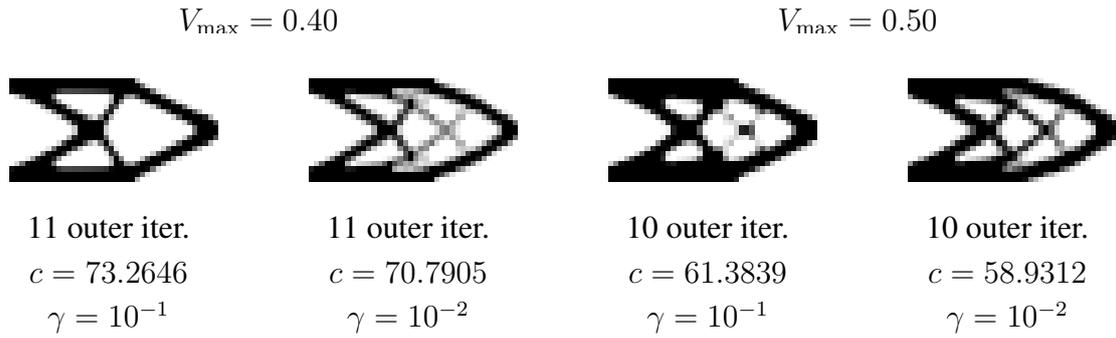


Figure 13: Solver `fmincon/sqp`, Problem 1, 40×20 mesh, $\varepsilon_c = 10^{-2}$, 5 inner iterations.

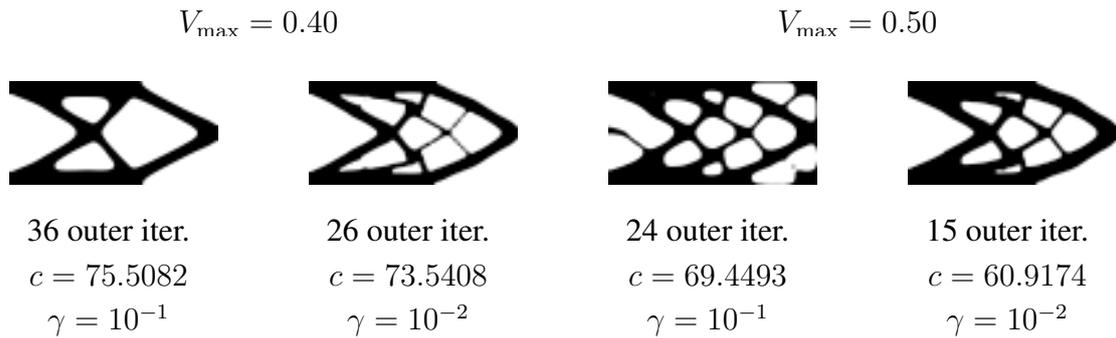


Figure 14: Solver `MMAspec`, Problem 1, 80×40 mesh, $\varepsilon_c = 10^{-2}$, 4 inner iterations.

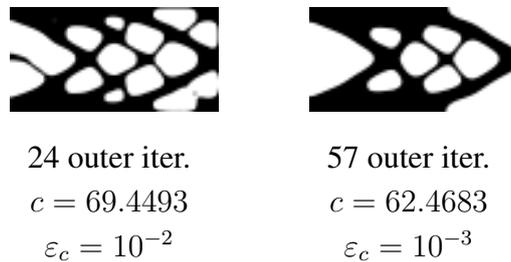


Figure 15: Solver `MMAspec`, Problem 1, 80×40 mesh, $V_{\max} = 0.50$, $\gamma = 10^{-1}$, 4 inner iterations.

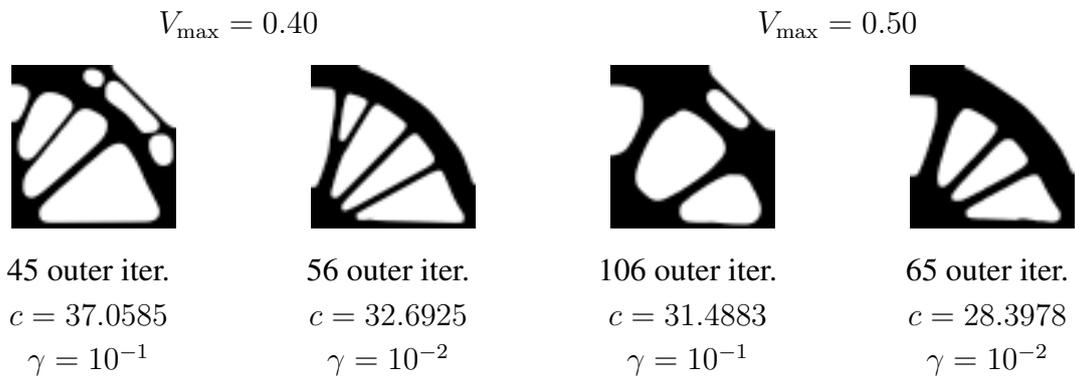


Figure 16: Solver `fmincon/ip`, Problem 2, 60×60 mesh, $\varepsilon_c = 10^{-3}$, 5 inner iterations.

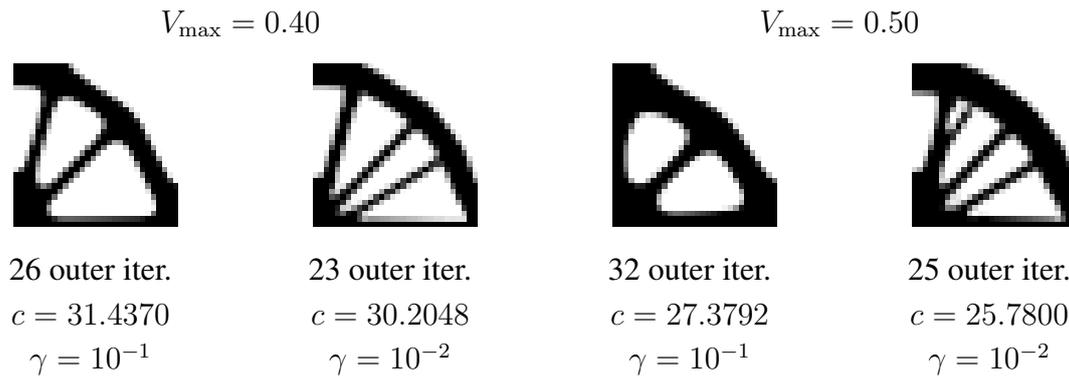


Figure 17: Solver `fmincon/sqp`, Problem 2, 30×30 mesh, $\varepsilon_c = 10^{-3}$, 5 inner iterations.

Tightening to $\varepsilon_c = 10^{-4}$ the relative tolerance for the instances with $\gamma = 10^{-2}$, we obtained better (cleaner) structures, cf. Fig. 19. In this case, although it reached the maximum allowed number of iterations, the structure that demanded 40% of the volume practically used all the available material.

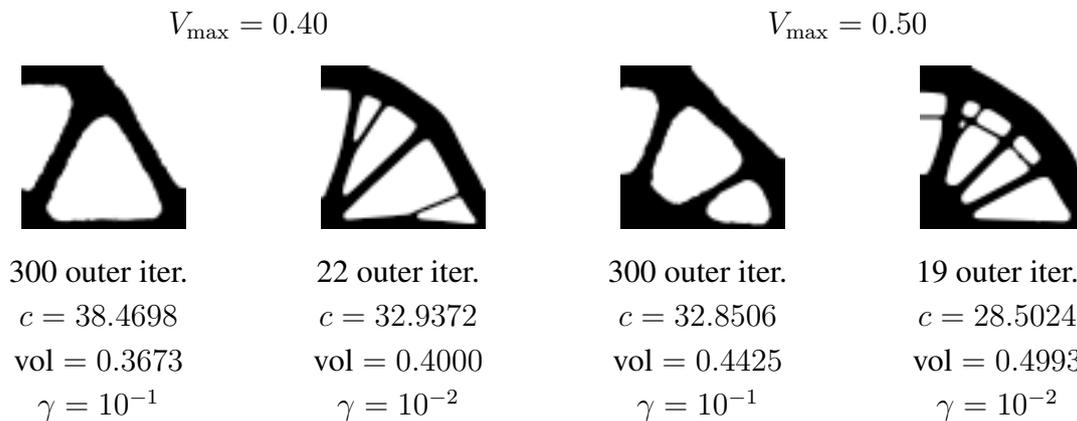


Figure 18: Solver `MMAorig`, Problem 2, 60×60 mesh, $\varepsilon_c = 10^{-3}$, 4 inner iterations.

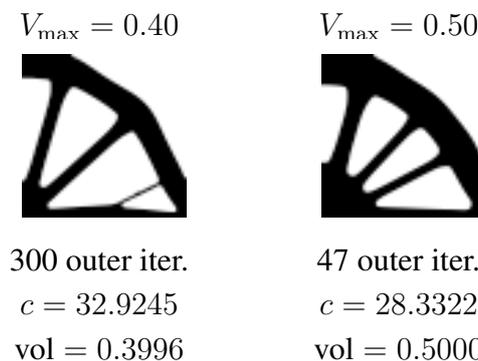


Figure 19: Solver `MMAorig`, Problem 2, 60×60 mesh, $\varepsilon_c = 10^{-4}$, $\gamma = 10^{-2}$, 4 inner iterations.

Figure 20 shows the outcomes of the solver `MMAspec` for Problem 2 for the combination of choices of γ and the percentage of volume previously considered.

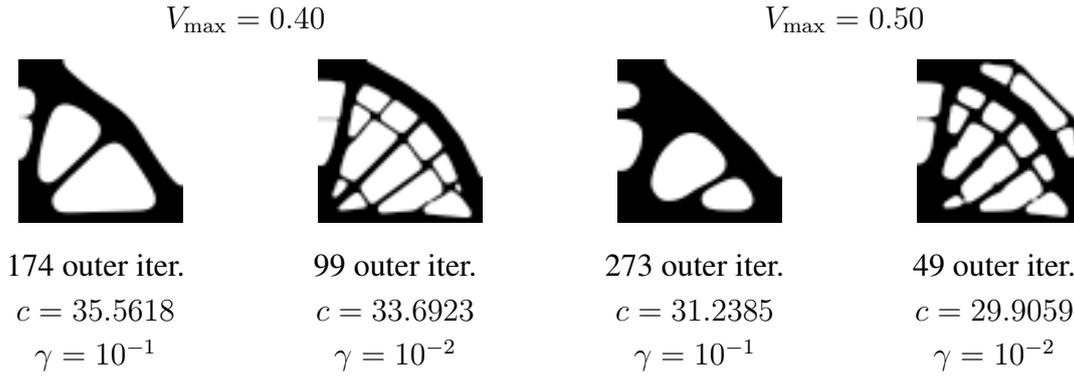


Figure 20: Solver MMAspec, Problem 2, 60×60 mesh, $\varepsilon_c = 10^{-4}$, 4 inner iterations.

An important aspect that deserves attention is the influence of the initialization on the final structure. To illustrate such a dependence, we depict in Fig. 21 two initial configurations for the 60×60 mesh of Problem 2, and the corresponding structures that were obtained by the solver fmincon/ip, with $\gamma = 10^{-1}$ and $V_{\max} = 0.40$.

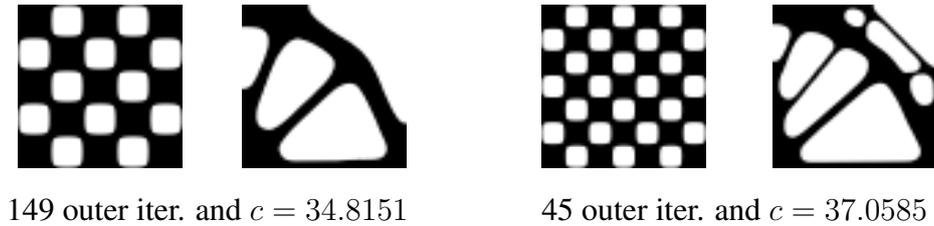


Figure 21: Initial and final structures obtained by fmincon/ip: Problem 2, 60×60 mesh, $\gamma = 10^{-1}$, $V_{\max} = 0.40$, $\varepsilon_c = 10^{-3}$.

A comparative view of the evolution of the compliance and the average volume of the structure against the iterations for the solvers under analysis is provided in Fig. 22 for the instance of Problem 1 with $V_{\max} = 0.40$ and the perimeter control parameter set as $\gamma = 10^{-2}$. The corresponding structures are summarized in Fig. 23

To offer the reader an idea of the average CPU time demanded per each call of the solvers under consideration, we have summarized in Table 1 the results corresponding to the aforementioned instance. It is worth mentioning that our code is not completely optimized for Matlab, nevertheless the basic framework is common for all the solvers, so the comparison is consistent. Despite dealing with linear systems that have four times smaller dimensions, the solver fmincon/sqp costs, per iteration, much more than the other three solvers.

Table 1: Average demanded CPU time for Problem 1, with $V_{\max} = 0.40$ and $\gamma = 10^{-2}$.

Solver	fmincon/ip	fmincon/sqp	MMAorig	MMAspec
Demanded CPU (s)	390.56	1611.03	170.45	350.85
Total calls of solver	$15 \times 5 = 75$	$11 \times 5 = 55$	$13 \times 4 = 52$	$26 \times 4 = 102$
Average CPU/call	5.21	29.29	3.28	3.37

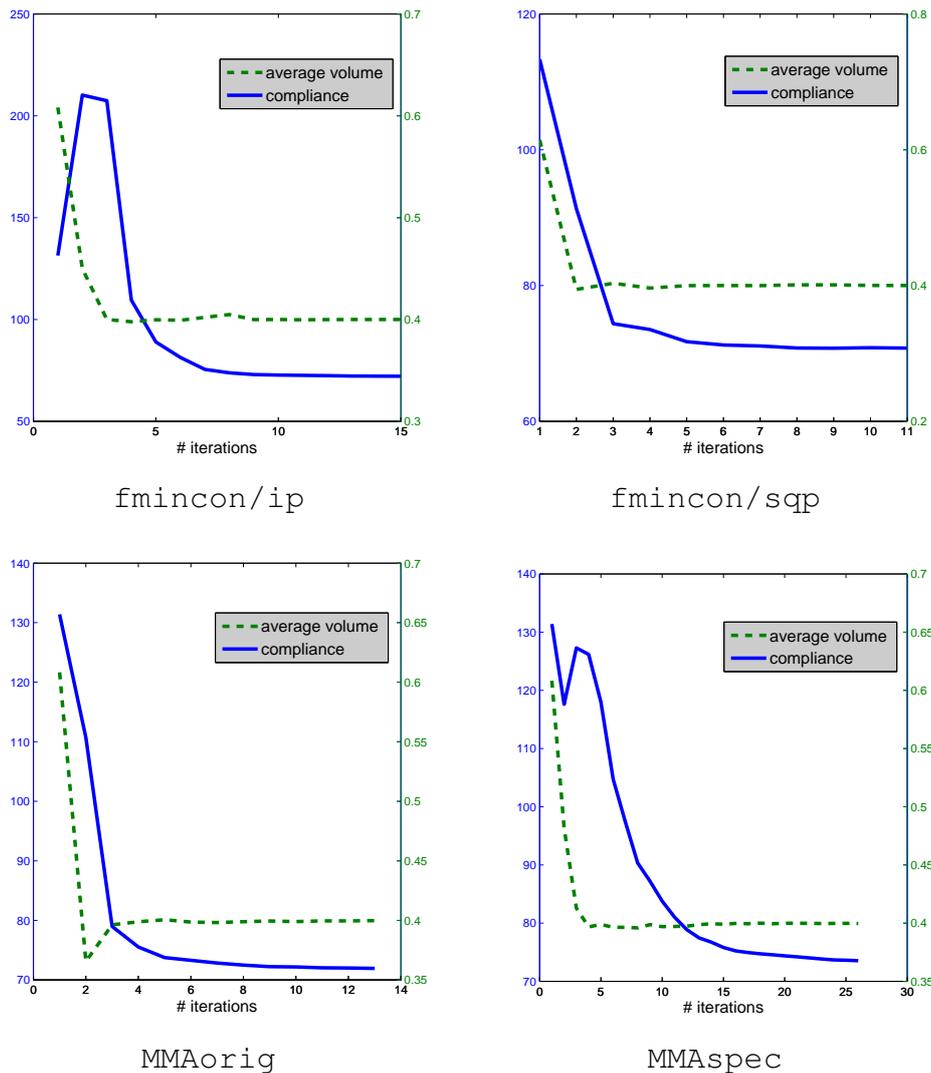


Figure 22: Compliance (blue continuous line) and average volume (green dashed line) against iterations for the distinct solvers – Problem 1, with $V_{\max} = 0.40$ and $\gamma = 10^{-2}$.

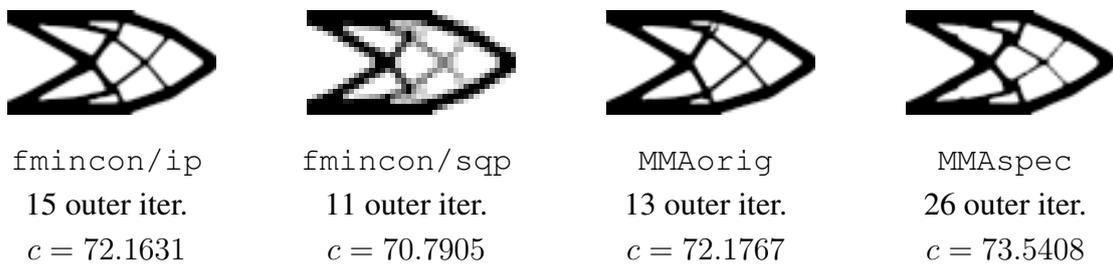


Figure 23: Problem 1, $\gamma = 10^{-2}$, $V_{\max} = 0.40$.

Concerning Problem 2, a similar analysis was prepared for the instance with $V_{\max} = 0.40$ and $\gamma = 10^{-2}$. The comparative evolution of compliance and average volume along the iterations for each solver is presented in Fig. 24, with the corresponding structures summarized in Fig. 25. Table 2 contains the average CPU time demanded per each iteration of the solvers

under consideration.

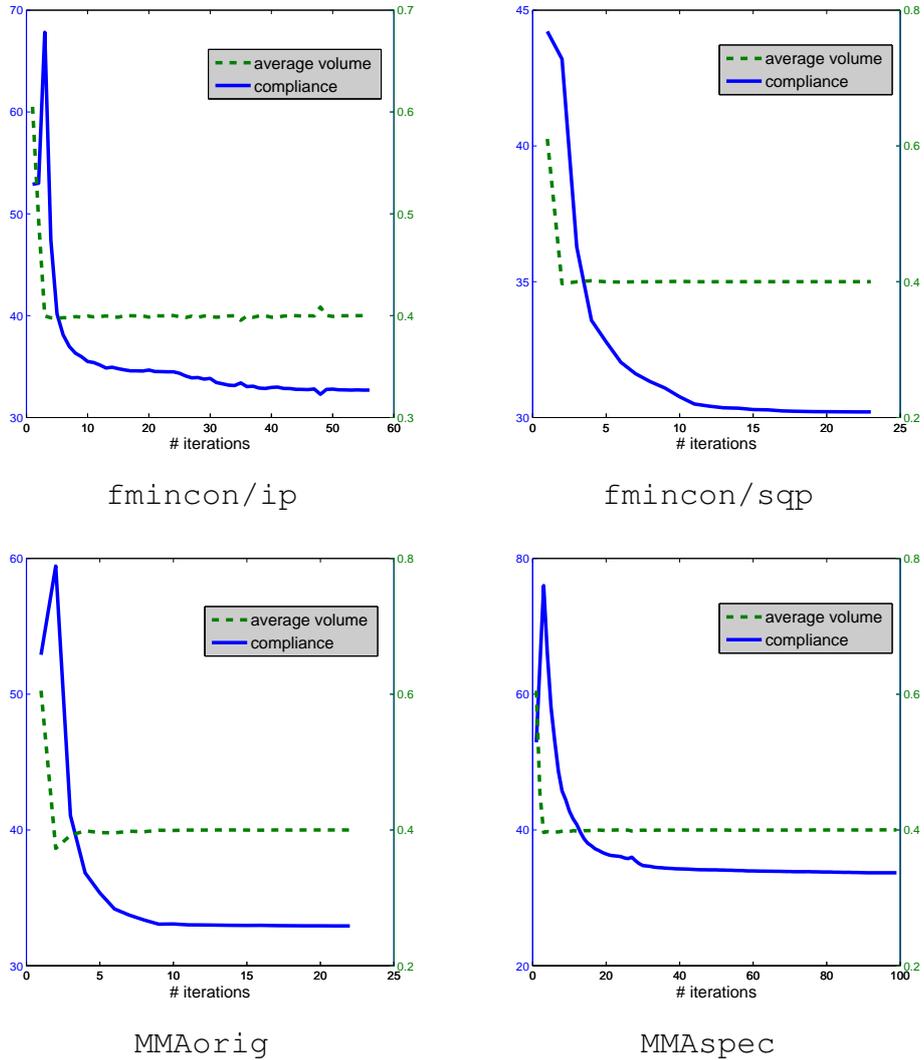


Figure 24: Compliance (blue continuous line) and average volume (green dashed line) against iterations for the distinct solvers – Problem 2, with $V_{\max} = 0.40$ and $\gamma = 10^{-2}$.

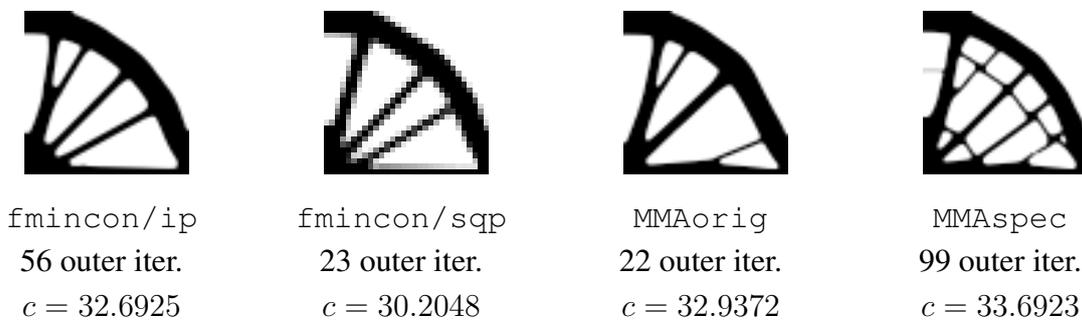


Figure 25: Problem 2, $\gamma = 10^{-2}$, $V_{\max} = 0.40$.

Table 2: Average demanded CPU time for Problem 2, with $V_{\max} = 0.40$ and $\gamma = 10^{-2}$.

Solver	fmincon/ip	fmincon/sqp	MMAorig	MMAspec
Demanded CPU (s)	2000.64	4385.62	363.75	1372.54
Total calls of solver	$56 \times 5 = 280$	$23 \times 5 = 115$	$22 \times 4 = 88$	$99 \times 4 = 396$
Average CPU/call	7.15	38.14	4.13	3.47

4 PERSPECTIVES

The nonlinear programming approach has shown to be an effective strategy for updating the level-set function, especially when the obtained results are compared, qualitatively speaking, with those of the codes of Challis (2010) and Otomori et. al (2015) for the same instances. Nevertheless, it is worth mentioning that the demanded CPU time of our code, which is not completely optimized, is not comparable with the demanded CPU time of the optimized codes of Challis (2010) and Otomori et. al (2015). We have addressed two benchmark structures from the literature.

Four solvers were analyzed, namely `fmincon` with the interior point strategy, `fmincon` with the sequential quadratic programming option, the original MMA of Svanberg (1987) and a recent modification of the MMA with a spectral updating (Gomes-Ruggiero et. al. (2010)). The features and peculiarities of each solver were translated into the obtained results. The solver `fmincon/sqp` cannot handle large scale problems, so the meshes had to be reduced. Despite of that, this solver demanded the largest amount of time per iteration among the four solvers, for both problems under consideration.

In our investigation, as in Dijk et. al. (2012), we have also observed that not only the final obtained structure depends on the initial configuration, but also that nucleation does not occur. Another similarity with Dijk et. al. (2012) was the need to prevent gray areas by adopting some sort of control. We have adopted the perimeter control by means of an extra penalization term added to the compliance in the objective function.

We have noticed that, at the beginning, the generated sequences have good progresses, but eventually they take very short steps. It is possible that some preconditioning could accelerate the iterates (see e.g. Dijk et. al. (2012)) and this might be a line for further investigation.

Our work in progress includes addressing problems with additional constraints, such as compliant mechanisms, as our approach allows dealing with the original constraints, without any need of employing a penalization strategy. Following ideas from Otomori et. al. (2011), we intend to consider a fictitious term at the objective function, to allow nucleation of the structure as well.

ACKNOWLEDGEMENTS

The authors are thankful to Prof. Krister Svanberg and to Dr. Mael Sachine for gently providing us their Matlab routines for running the original MMA and its modified spectral version, respectively. This work was partially supported by CNPq (Grant 304032/2010-7) and FAPESP (Grants 2013/05475-7 and 2013/07375-0).

REFERENCES

- Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., & Sigmund, O., 2011. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, vol. 43, pp. 1–16.
- Byrd, R. H., Gilbert, J. C., & Nocedal, J., 2000. A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming. *Mathematical Programming*, vol. 89, n. 1, pp. 149–185.
- Challis, V. J., 2010. A discrete level-set topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, vol. 41, pp. 453–464.
- Deaton, J. D., & Grandhi, R. V., 2014. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, vol. 49, pp. 1–38.
- Dijk, N. P., Langelaar, M., & Keulen, F., 2012. Explicit level-set-based topology optimization using an exact Heaviside function and consistent sensitivity analysis. *International Journal for Numerical Methods in Engineering*, vol. 91, pp. 67–97.
- Dijk, N. P., Maute, K., Langelaar, M., & Keulen, F., 2013. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, vol. 48, pp. 437–472.
- Dunning, P. D. & Kim, A., 2013. A new hole insertion method for level set based structural topology optimization. *International Journal for Numerical Methods in Engineering*, vol. 93, pp. 118–134.
- Gomes-Ruggiero, M. A., Sachine, M., & Santos, S. A., 2010. A spectral updating for the method of moving asymptotes. *Optimization Methods & Software*, vol. 25, n. 6, pp. 883–893.
- Nocedal, J. & Wright, S. J., 2006. *Numerical Optimization*, 2nd ed. Springer Series in Operations Research, Springer Verlag.
- Otomori, M., Yamada, T., Izui, K., & Nishiwaki, S., 2011. Level set-based topology optimisation of a compliant mechanism using mathematical programming. *Mechanical Sciences*, vol. 2, pp. 91–98.
- Otomori, M., Yamada, T., Izui, K. & Nishiwaki, S., 2015. Matlab code for a level set-based topology optimization method using a reaction diffusion equation. *Structural and Multidisciplinary Optimization*, vol. 51, pp. 1159–1172.
- Sigmund, O., & Maute, K., 2013. Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*, vol. 48, pp. 1031–1055.
- Svanberg, K., 1987. The method of moving asymptotes – a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 359–373.
- Yamada, T., Izui, K., Nishiwaki, S., & Takezawa, A., 2010. A topology optimization method based on the level set method incorporating a fictitious interface energy. *Computer Methods in Applied Mechanics and Engineering*, vol. 199, pp. 2876–2891.