

Toptimiz3D: a topology optimization software using unstructured meshes

Ernesto Aranda, José Carlos Bellido, Alberto Donoso

*Departamento de Matemáticas
E.T.S. de Ingeniería Industrial
Universidad de Castilla - La Mancha
13071 Ciudad Real, Spain*

Abstract

Toptimiz3D is a graphical user interface for setting up and solving structural topology optimization problems which can handle with general geometries and unstructured meshes. And it is precisely here where the emphasis is placed on. Unlike other existing educational softwares, Toptimiz3D is intended to go a bit further as the aforementioned strengths allow us to face more realistic situations and therefore optimize structures coming for real-world problems. It solves five types of topology optimization benchmark problems (single and multiple loads compliance problem, the volume problem, compliant mechanism design and the stress constrained problem) by means of the SIMP method. Moreover, three optimization algorithms are included (MMA, IPOPT, OC), and two types of filters are incorporated (density filter with a conic function and a Helmholtz-type PDE-based filter). This application offers a user-friendly environment to specify all relevant data in order to define a topology optimization problem and solve it directly from the interface. Toptimiz3D is a free-distribution software open to future development.

Keywords: topology optimization, SIMP method, finite element software, unstructured meshes

1. Introduction

Thirty years after the publication of the pioneering work [1], there is no doubt in claiming that topology optimization can be considered nowadays as one of the major conceptual tools for structural design. In that remarkable work, M.P. Bendsøe and N. Kikuchi, starting out from homogenization theory, were the first to come up with a well-founded numerical way of optimizing the topology of general structures. Two nice and comprehensive books regarding topology optimization from this homogenized perspective are the monographs [2] and [3]. Just one year later, in 1989, the Solid Isotropic Material with Penalization (SIMP) method was conceived in [4] as an alternative to the homogenization approach, in order to avoid the presence of composite materials in the final layouts and therefore converge to black-and-white designs. In short, these two aforementioned contributions ([1, 4]) were the germ of the topology optimization method as it is understood nowadays. Since then, the development of the area has been overwhelming and topology optimization as a tool is very consolidated today in engineering companies, and not only that, it is still a matter of study in academia.

The way to proceed on using SIMP is typically called a density-based approach, and it is explained later on. However, there exist other ways to do topology optimization, being the second more extended the so-called level-set method ([5, 6, 7]), which it is gaining in popularity and followers as well. Although these optimization techniques were initially conceived for structural design only, the idea behind the method has been successfully spread to multiple physical contexts and consequently the number of references regarding applications of the method is on the rise as well. For the purpose of getting an overall perspective of both the method and the applications, the reader is referred to the book [8], as well as the surveys ([9, 10, 11]).

The interest for these problems has resulted in many software products, both academic and commercial, using different techniques and perspectives. But in many cases, it can be tricky to adapt them when we want to solve realistic

problems defined in general geometries. Two existing proposals in 3d that use structured meshes are the educational reference [12] and the code using PETSc [13] that lets obtain finer results using a parallel framework with a lot of cores. In 2d, there are however far more proposals (some of the most well-known educational are [14, 15, 16, 17]). We just point up here PolyTop [18] because as far as the authors' knowledge it is the only one that uses unstructured polygonal finite element meshes.

The aim of this work is to release a free-distribution software to solve topology optimization problems for structures and mechanisms in the framework of linear elasticity, placing emphasis on the fact that it may handle with general geometries and, more importantly, unstructured meshes, both in 2d and 3d, without any extra difficulty. Another main characteristic is that it allows the user to define meshes and set up the problem selecting different options to solve it through a Graphical User Interface (GUI). The interface is coded in Python and, once all information is entered, a C++ code is built, compiled and executed. The MFEM [19] library for finite element analysis is used.

The software provides the possibility of running from the GUI or exporting the generated code for running outside the application. We have incorporated different solvers and options which are very easy to select and it is also possible to export the results in VTK format (The Visualization Toolkit, [20]) for post-processing with ParaView [21]). There are other functionalities that can be consulted in the manual (`Help` menu), distributed with the application.

`Toptimiz3D` is open to everybody for free use and future code extensions in a repository at GitLab¹. There is also available a Live CD downloadable from Sourceforge² for quick use without installation (notice that `Toptimiz3D` has dependencies on certain libraries that are required for a proper installation).

In summary, our software tries to meet a twofold requirement: on the one hand and from educational purposes, `Toptimiz3D` is intended to go a bit further

¹<https://gitlab.com/e-aranda/topt-mfem>

²<https://sourceforge.net/projects/live-toptimiz3d/>

than existing educational softwares on using unstructured meshes in both 2d and
60 3d situations; on the other hand, it also pretends to face more realistic examples
on optimizing structures coming from real-world problems.

The layout of the paper is as follows. Section 2 is devoted to show a brief
overview on the topology optimization method via SIMP as well as the type of
problems `Optimiz3D` solves. Section 3 describes the main characteristic of the
65 software and section 4 includes several examples of application. Finally, some
conclusions and comments are provided in the last section.

2. Structural topology optimization problems

The most paradigmatic example of a topology optimization problem consists
of finding the best way of distributing a limited amount of material in a design
domain in order to maximize the stiffness of the resulting structure, or equiva-
lently, to minimize the structure flexibility or compliance. This problem is pop-
ularly known as the *compliance minimization problem*. To set it up, let Ω be the
reference configuration of a linearly elastic material, with $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_R \cup \Gamma_F$
(see fig. 1). If $\sigma(\mathbf{u})$ is the stress given by Hooke's law:

$$\sigma(\mathbf{u}) = 2\mu\varepsilon(\mathbf{u}) + \lambda \operatorname{tr}(\varepsilon(\mathbf{u}))I$$

where $\varepsilon_{ij}(\mathbf{u}) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ is the linearized strain tensor, and λ and μ are
the Lamé coefficients (defined below), then the displacement \mathbf{u} of the structure
is the solution of the linear elasticity system:

$$\left\{ \begin{array}{l} -\operatorname{div}(\sigma(\mathbf{u})) = \mathbf{f} \quad \text{in } \Omega \end{array} \right. \quad (1)$$

$$\left\{ \begin{array}{l} \mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_D \end{array} \right. \quad (2)$$

$$\left\{ \begin{array}{l} \sigma(\mathbf{u}) \cdot \mathbf{n} = \mathbf{t} \quad \text{on } \Gamma_N \end{array} \right. \quad (3)$$

$$\left\{ \begin{array}{l} M\mathbf{u} + \sigma(\mathbf{u}) \cdot \mathbf{n} = \mathbf{h} \quad \text{on } \Gamma_R \end{array} \right. \quad (4)$$

Here \mathbf{f} stands for a distributed body force acting on Ω (or on a part of it).
Regarding boundary conditions, \mathbf{t} is a traction applied on Γ_N , with \mathbf{n} the outer
70 unit normal, $M\mathbf{u} + \sigma(\mathbf{u}) \cdot \mathbf{n} = \mathbf{h}$ are mixed (Robin) boundary conditions, with

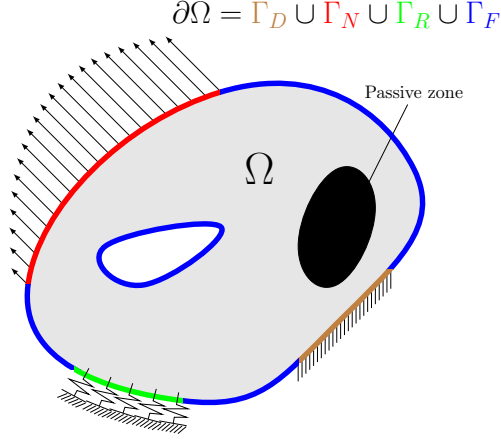


Figure 1: Design domain

M the spring stiffness matrix and \mathbf{h} the applied traction, on the part of the boundary Γ_R . The displacement is fixed on Γ_D . Finally, we have free boundary conditions on Γ_F .

For isotropic elastic bodies, the Lamé parameters λ , μ are defined as

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}, \quad \mu = \frac{E}{2(1 + \nu)} \quad (5)$$

where E is the Young's modulus and ν the Poisson's ratio of the material. For 2d problems it is possible to consider *plain stress* or *plain strain* assumptions, which means that, in the former, fictitious Lamé coefficients λ and μ are computed as:

$$\lambda = \frac{\nu E}{1 - \nu^2}, \quad \mu = \frac{E}{2(1 + \nu)}. \quad (6)$$

In order to set the optimal design problem a characteristic function χ is introduced as the unknown, corresponding to the subset of the domain where we put the material. Therefore we assume that the material has a Young's modulus which corresponds to a mixture between a solid material and void. To avoid singularity problems the void has to be modeled as a material with a very

small Young's modulus E_{\min} so that

$$E = E_0\chi + E_{\min}(1 - \chi),$$

where E_0 is the Young's modulus of solid material. As this problem is ill-posed, the SIMP method [4, 17] substitutes the characteristic function χ by a density function $\rho \in [0, 1]$ with a penalization in the form:

$$E \equiv E(\rho) = E_0\rho^p + E_{\min}(1 - \rho^p) = \rho^p(E_0 - E_{\min}) + E_{\min} \quad (7)$$

where p is the SIMP penalization parameter.

Then, maximizing the stiffness of the structure is equivalent to minimizing the flexibility or compliance, which can be measured by the functional

$$J(\rho) = \int_{\Omega} \mathbf{f} \cdot \mathbf{u} + \int_{\Gamma_N} \mathbf{t} \cdot \mathbf{u} + \int_{\Gamma_R} \mathbf{h} \cdot \mathbf{u} \quad (8)$$

where \mathbf{u} is the solution of the elasticity system (1)–(4). We will refer to this problem as *Compliance*. Finally, the volume constraint is expressed by

$$V(\rho) = \frac{1}{|\Omega|} \int_{\Omega} \rho \leq V^*$$

75 where $0 < V^* < 1$ is a prescribed volume fraction.

In the same framework, it is possible to set up two related problems. The first one, consisting of finding the material distribution with minimum volume under a compliance constraint, that we call *Volume* problem, and the second one, a compliant mechanism problem, where the displacement of the structure in some part of the boundary is minimized under a volume constraint (*Mechanism* problem). This corresponds to minimize the functional

$$J_M(\rho) = \int_{\Gamma_M} \mathbf{L} \cdot \mathbf{u}, \quad (9)$$

where Γ_M is the part of the boundary where the displacement in the direction of some prescribed vector \mathbf{L} is measured. Obviously, $\Gamma_M \cap \Gamma_D = \emptyset$. Note that, eventually, Γ_M can be a part of the rest of the boundary.

It is well known that SIMP approach requires of regularization in order to
80 admit optimal solutions, what it is evidenced with the appearance of numerical

instabilities, mainly mesh dependency and checkerboard pattern formation, in the absence of regularization. The most widely used regularization is the filtering process, what can be viewed as some kind of regularization of the density field.

Our software implements two different filters: a density filter with a conic function and a filter based on the solution of a Helmholtz-type differential equation. The first one is the result of the convolution of the density with a smooth function K ([22]):

$$\hat{\rho} = (\rho * K)(x) = \int_{\Omega} \rho(y)K(x - y) dy, \quad x, y \in \Omega \quad (10)$$

where K is defined by $K(z) = C \max\{0, 1 - \frac{|z|}{r_h}\}$, with r_h the length scale and C is a normalization constant such that $\int_{\Omega} K(z) dz = 1$. From a numerical point of view, this is equivalent to a matrix multiplication. The second one, initially proposed in [23], replace the density ρ by the solution of the following Helmholtz PDE:

$$\begin{cases} -r^2 \Delta \hat{\rho} + \hat{\rho} = \rho & \text{in } \Omega \\ \frac{\partial \hat{\rho}}{\partial \mathbf{n}} = 0 & \text{on } \partial \Omega \end{cases} \quad (11)$$

being r a length scale.³

On the other hand, in order to have “black and white” designs, that is, designs with density values close to 1/0, (meaning that density values of 1 are associated with black and that density values of 0 are associated with white), filtered densities are usually projected using a smooth function that produces high contrast designs. (cf. [24, 25, 26]). The projection function used here is an approximation of a Heaviside function centered at $\frac{1}{2}$:

$$\Psi(\rho_e) = \frac{1}{2} + \frac{\tanh(\beta(\rho_e - \frac{1}{2}))}{2 \tanh(\frac{1}{2}\beta)}. \quad (12)$$

⁸⁵ Here, ρ_e stands for the value of the density in the element e of the triangulation and β is the parameter of the projection, which is usually implemented through a continuation method.

³Note that the length scales r and r_h are different. According to [23], the estimation between the relationship between those parameters is given by $r_h = 2\sqrt{3}r$.

Therefore, the evaluation of objective functions for the three problems, *Compliance*, *Volume* and *Mechanism*, is carried on solving the elasticity problem with a filtered and projected density, that is, E in (7) has to be written as $E(\Psi(\hat{\rho}))$ in the elasticity system. Next, we explain some technical details on how the software proceeds to solve a problem.

3. Software description

`TopOptim3D` is a software that provides a GUI where the user can solve, after introducing all relevant data, a problem among five kinds of topology optimization problems: the usual **compliance** minimization problem with a volume constraint, a **multiload** compliance problem with volume constraint, the minimization of **volume** under a compliance constraint, a compliant **mechanism** problem under a volume constraint and the minimization of the volume under a *P-norm* measure of the **stress** constraint (as it was described in [27]).

Once the problem has been set up, the software automatically generates a C++ code, compiles and runs it in a transparent way. To solve any of these problems, a density-based approach (the SIMP method) is implemented with three optimization methods: the widely used Method of Moving Asymptotes (MMA, cf. [28]), the Interior Point Optimizer (IPOPT, see. [29]), and the usual Optimality Conditions (OC) as it was described in [8, Eq.(1.12)]. The last one has only been implemented for the compliance and multiload problems.

The interface is divided in four panels and a menu bar (see fig. 2) where the user can load and visualize the mesh, introduce the necessary information to set up the problem, do the compilation, execution and visualization of the results Figure 3 shows cuts of different panels corresponding to the data of some examples in the next section.

The various input parameters for the problem, from the elastic constants of the material, the volume fraction, tolerance, etc., to the boundary conditions, minimization method or filter type are introduced in several panels or selecting options in the menu bar. A detailed description of all menus and panels are in

the manual that it is distributed with the software.

One key aspect of the software is that the user must provide a suitable mesh for the problem which has to be built in an external tool. Different mesh
120 formats are allowed: mainly, the MFEM mesh format, GMSH mesh files, GMF
(the Gamma Mesh Format, see [30]) and COMSOL text mesh files (`.mpltxt`
extension). Triangular and quadrilateral meshes in 2D and tetrahedral and
hexahedral meshes in 3D are allowed and the software uses linear finite elements
interpolation.⁴ Namely, the mesh has to be properly labeled in order to impose
125 the boundary conditions on the problem. Once the mesh is loaded, it can be
visualized and handled at the *Graphic Panel* (rotated, moved, ...) and the
boundaries can be highlighted by right clicking on the corresponding line or
surface. Note that the selected label's numbers associated to specific boundaries
are shown at the bottom left corner of the panel. There are also a couple of
130 drop-down buttons to highlight the inner and boundary labels at the *Mesh*
Information Panel.

Once the problem has been set up, the code is generated by selecting **Pro-**
cessing from the `Process` menu⁵, the C++ code is built up and compiled, and
a terminal-like panel is showed where it can be run from. Any subsequent mod-
135 ification of parameters needs **Processing** to be selected again before running.⁶

Pressing `Run` button starts the minimization process and a GLVIS⁷ graphic
window is shown (as a separated window) with the result of each iteration.
Once the computation is finished, the obtained design is shown in the Graphic
Panel. Density, stress and deformed configuration can be seen selecting the
140 corresponding button at the top bar of this panel.

⁴Future versions of the code will incorporate the possibility of use of higher order finite elements.

⁵At this point, validation of input data is performed, and a message error pops up if those are incongruent.

⁶There are some parameters (volume fraction, tolerance, iterations, filter radio and others) passed as command line arguments which don't need to rebuild and compile the code, while other changes in data problem involving code modification carries on a new compilation.

⁷<https://glvis.org>

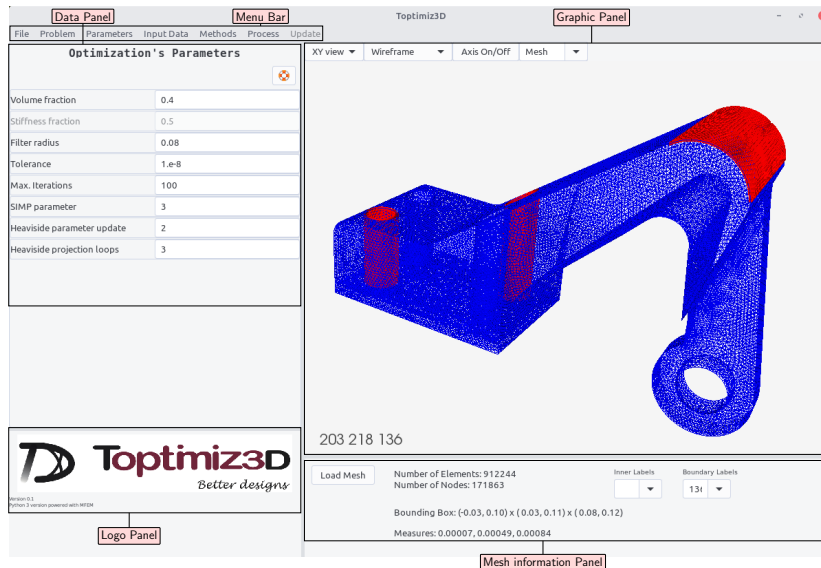


Figure 2: Toptimiz3D window

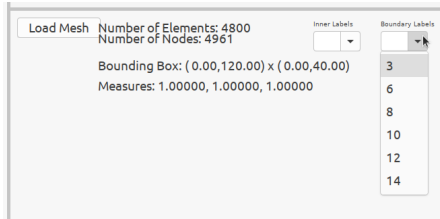
The software is able to export the results in VTK format for post processing
 or in MFEM format (.gf file), that could be eventually used as an initialization
 of subsequent simulations. Also, it is possible to export the generated C++
 code for running outside from the application, but notice that the code relies
 on several dependencies.

4. Examples

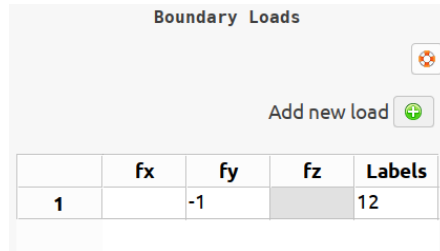
In this section we will show how to set up and solve some topology optimization
 problems. All files used in these examples, except ex. 4.7, are distributed
 with the software in the `examples` folder.

4.1. The Messerschmitt-Bölkow-Blohm Beam

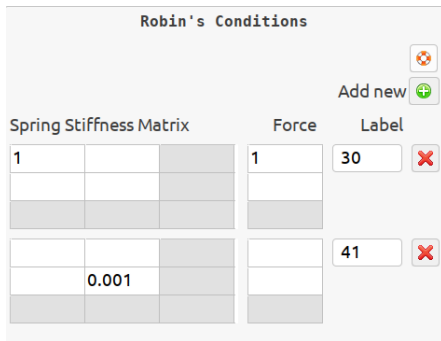
We start with the classic MBB-beam example where the design domain and
 dimensions are shown in fig. 4. We will consider the same parameters as in [31],
 that is, $E_0 = 1$, $E_{\min} = 10^{-9}$, $\nu = 0.3$, $p = 3$, $f_{in} = 1$, the filter radius $r_h = 3.5$



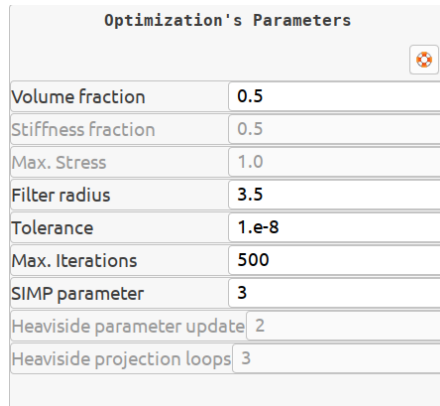
(a) Mesh information



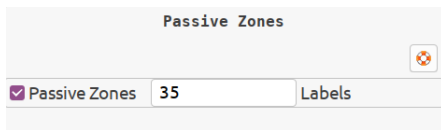
(b) Boundary loads



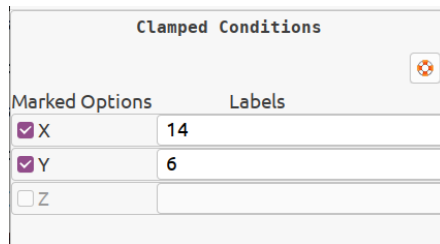
(c) Robin boundary conditions



(d) Optimization's parameters



(e) Passive zones



(f) Fixed displacements

Figure 3: Different panels for introducing data problem

for the conic filter and the volume fraction $V^* = 0.5$. Also, as in [31], we model
 155 only half the domain (see fig. 5).

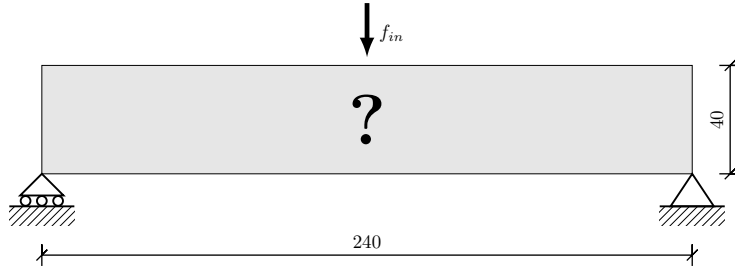


Figure 4: Original domain for the MBB problem

As MFEM relies on variational formulations, it can be tricky to impose
 point loads or fixed conditions at a point, so in order to reproduce this typical
 benchmark we have built a mesh such that the vertical force f_{in} and the right
 bottom support can be imposed on a very small part of the boundary. Figure 6
 160 shows the mesh we have used for computations (file `mbb-quad.mesh`). The
 parts of the boundary where the force and the right support are applied are
 highlighted in red and the vertical roller support is highlighted in green. Figure 6
 also shows the assigned labels to these boundaries. Note that the mesh must
 be created in order to have those relevant parts. This mesh has been built with
 165 the SALOME platform ([32]), and exported in GMF format which can be read

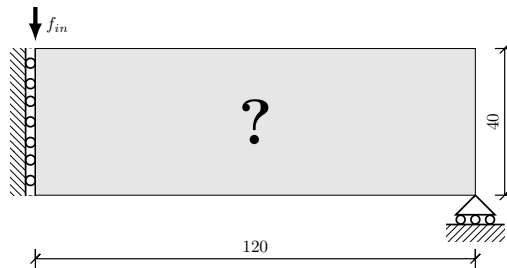


Figure 5: Effective domain

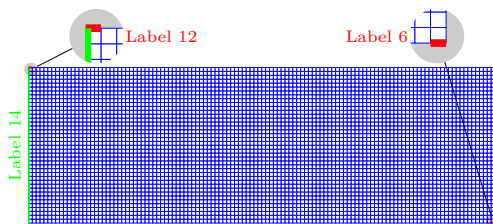


Figure 6: Mesh for the MBB problem

by `Toptimiz3D`.⁸

Once the mesh has been loaded, some information is shown at the *Mesh Information Panel* (see fig. 3a): the number of elements and nodes, the dimension of a bounding box containing the whole domain⁹ and some *measures* which
 170 corresponds to an estimation of the elements' diameters.¹⁰ In this case, as we have built a regular quadrilateral mesh with size 1, all measures are the same.

The software starts with some default choices which coincides with the parameters for this problem (the elastic constants and the SIMP parameter, for example), so we only need to change the volume fraction, the filter radius and
 175 we choose 500 iterations¹¹ in **Optimization's Parameters** panel at `Parameters` menu (see fig. 3d).

In order to introduce the load and support conditions we need first to determine the suitable boundary labels. We can right click with the mouse over the left side of the rectangle and this will be highlighted in red and we will see
 180 the number 14 in the left bottom corner of the *Graphic Panel*. However, as the

⁸`Toptimiz3D` converts the GMF format to the MFEM format, saving a new file named `-mfem.mesh`. We recommend using that new file in order to avoid the conversion each time we use the file.

⁹It is important to note that `Toptimiz3D` considers that mesh files are dimensionless so rest of data must be consistent with that.

¹⁰Namely, the shown values are the d th root of the minimum, mean and maximum d -measure, where d is the space dimension.

¹¹The meaning of this parameter depends on the method used. In case of MMA, the algorithm will stop when the number of functions evaluations exceeds that value.

other two relevant boundaries are very small it is difficult to get those labels with the mouse.¹² In this case we can use the *Boundary Labels* drop-down button (fig. 3a) and select the label from the list until we highlight the boundary we want. We will see that the label of the top left corner is 12 and the label
185 of the bottom right corner is 6. Next we select **Boundary Loads** from the **Boundary Conditions** submenu and introduce the corresponding force. In this case we need to introduce the force $\mathbf{t} = (0, f_{in})$ with $f_{in} = -1$ with label 12. In case of zero values it is not necessary to introduce the zero component but only **fy** and **Labels** (see fig. 3b).

190 Finally we choose **Fixed Displacements** from the same submenu and mark the **X** option with label 14 and **Y** option with label 6 for the roller supports (fig. 3f).

We select IPOPT method from **Optimization Method**, *Conic filter* from **Filter Type** and **Without Heaviside Projection** from the submenu **Heaviside Projection** of **Methods** menu.
195

Selecting **Processing** in the **Process** menu creates the C++ code and compiles it. Then the **Results** panel is shown and we can press the **Run** button to execute.

When the algorithm stops, we can see the result (fig. 7) in the *Graphic Panel* and also the *relaxed* Von Mises stress (see example 4.6) and the deformed configuration selecting the right drop-down button on the top of this panel.¹³
200 We can export the solution in VTK format selecting **Save VTK Results** from **File** menu.

It is a good idea to save a file with the configuration of the example so it
205 can be quickly used or modified. To do this, select **Save Config File** from **File** menu and save the `.topt` file. The configuration file `mbb.topt` for this problem is distributed with the software.

¹²It is possible if we zoom in and move the mesh.

¹³For visualization purpose, displacements are scaled to show deformations between 5% and 20%.



Figure 7: Optimal design for the MBB problem

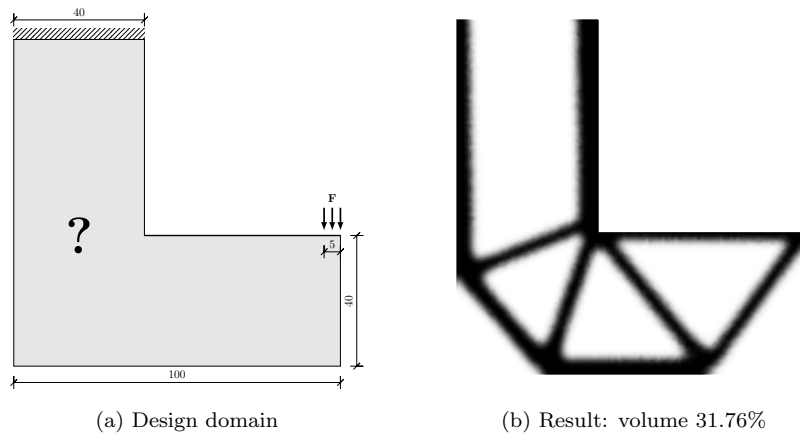


Figure 8: Design domain and result for a loaded knee structure

4.2. A loaded knee structure

Another typical example in structural topology optimization (cf. [27]) corresponds to the domain shown at figure 8a, where, in this case, we are considering a *Volume* problem, that is, we look for the design that minimizes the total weight of the structure under a compliance constraint.

The material constants, SIMP penalization factor and iterations are the same as in the previous example and now there is no volume fraction but a constraint on the compliance limited by a 40% of the minimum compliance.¹⁴

The typical initialization for the *Compliance* problem is a constant value

¹⁴This minimum compliance is computed for a total solid design.

equal to the volume fraction. In this case we use a constant value equal to 1, introduced at **Initial Density** panel. We use the conic filter with radius 2.5 and MMA as optimizer.

220 All data can be read from the configuration file `knee.topt` distributed with the software along with the mesh file `knee.mesh` with 14538 elements. We see that in the **Fixed Displacement** submenu the boxes **X** and **Y** are marked with label 2 and the boundary force $\mathbf{F} = (0, -\frac{3}{5})$ is acting on label 3.

The result (exported in VTK and post-process with ParaView) is shown at 225 fig. 8b. Once the execution is finished, the measure of non-discreteness (MND) (introduced in [31]) is shown, which gives a value of 18.86%. This high value is because we are not using Heaviside projections.

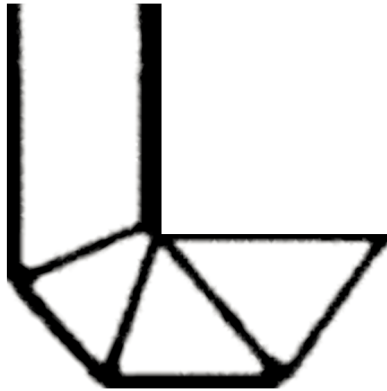


Figure 9: Volume with Heaviside projections: 25.11%

We can run the same example choosing **With Heaviside Projection**, and setting 4 Heaviside projection loops and 3 as Heaviside parameter update in 230 **Optimization's Parameters** panel.¹⁵ The new result is shown in fig. 9 and MND is reduced to 0.54%.

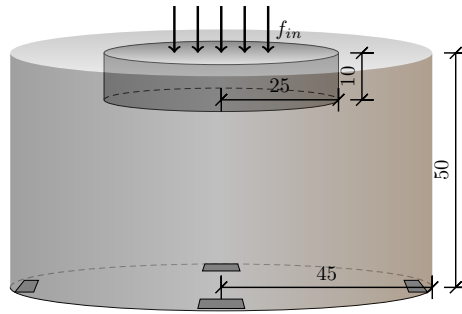
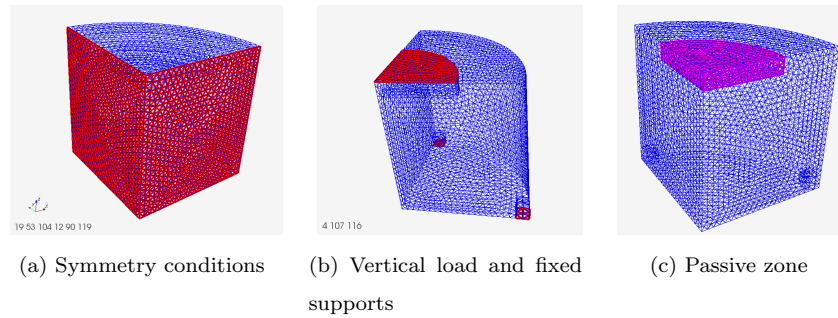


Figure 10: Domain design for the stool



(a) Symmetry conditions

(b) Vertical load and fixed supports

(c) Passive zone

Figure 11: Input data for the stool

4.3. The stool

We now show a small 3D example to check the capabilities of `Toptimiz3D` using 3D geometries. As it is well known, 3D examples are computationally costly, so we have considered a coarse mesh for a quarter of the domain shown at fig. 10 with 28062 tetrahedrons, where the small flat cylinder on the top is a passive zone, and it is clamped on the rectangular supports at the bottom. Configuration file (`stool.topt`) and mesh (`stool-mfem.mesh`) are in the `examples` folder.

We have imposed symmetry conditions on the walls (see fig. 11a), preventing

¹⁵In this case, the maximum value of β in (12) is 27.

the displacement on the \mathbf{X} direction for labels 19, 53 and 104, and on the \mathbf{Y} direction for labels 12, 90 and 119.¹⁶ Also, the structure is clamped at 107 and 116. This means that we must introduce the labels 19, 53, 104, 107, 116 for \mathbf{X} ; 12, 90, 119, 107, 116 for \mathbf{Y} ; and 107, 116 for \mathbf{Z} (as space separated lists).

245 Finally, we have imposed a boundary load $(0, 0, f_{in})$ at label 4, where $f_{in} = -1$ (see fig. 11b), and the seat is considered as a passive zone with label 2 (fig. 11c).

The volume fraction is $V^* = 0.1$, we have used a conic filter with radius $r_h = 4$, MMA as solver and Heaviside projections with 3 loops and update
 250 parameter of 2. This means that the bigger value of the β parameter in (12) will be $\beta = 4$. Clearly that is not a big enough value to provide high contrast designs but it is obviously faster. However, as we comment below, the quality of the result is acceptable. Note that MND in this example is 5.26%.

The result processed in ParaView is shown at fig. 12. As usually in 3D
 255 cases, density values on the domain are not shown, but an optimal structure formed by mesh elements whose density is above a threshold parameter t_0 . In order to determine this threshold, `Toptimiz3D` evaluates the volume of the sets $\Omega_t = \{x \in \Omega : \bar{\rho} \geq t\}$, for $t \in [0, 1]$, where $\bar{\rho}$ is the filtered and (eventually) projected optimum density. Then t_0 such that $|\Omega_{t_0}| = V^*|\Omega|$ is computed.
 260 Selecting *Thresholding* from the right drop-down button at the top of Graphic Panel, the plot of $f(t) = \frac{|\Omega_t|}{|\Omega|}$ (red line) and its intersection with V^* (blue line) are shown, which gives the threshold value (see fig. 13).

This plot is interesting because it gives us some information about the quality of the numerical result in terms of intermediate density values. In this case, the
 265 plot in fig. 13 is showing that the obtained design has virtually no values between 0.2 and 0.8, and the rest of values are very close to 0 and above 0.9. Note that an ideal design where density values are equal to 0 or 1 will have a function f satisfying $f(0) = 1$ and $f(t) = V^*$, $\forall t > 0$, so the thresholding function

¹⁶Orientation axis can be activated from the `Axis On/Off` button on the top of the graphic panel.

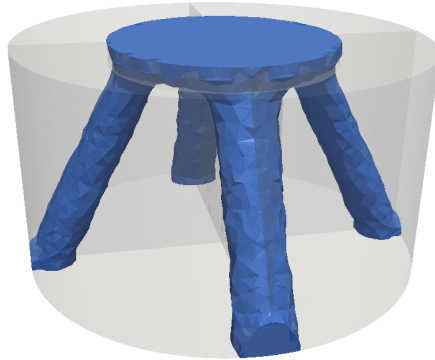


Figure 12: Result for the stool problem

obtained is not far from that. For example, the thresholding function for the
 270 second experiment of the loaded knee is shown at fig. 14. Obviously, increasing
 the number of loops of the Heaviside projection provides better designs but also
 increases the computational time.

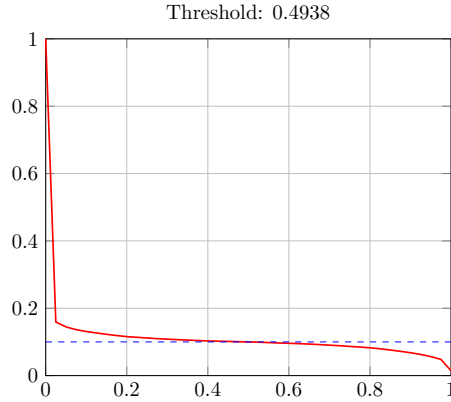


Figure 13: Thresholding function for the stool

4.4. A circular gripper

One of the main advantages of our software is the use of unstructured meshes
 275 which allows us to use non-trivial geometries. The following example is an adap-
 tation of the compliant mechanism gripper shown in [25] to a circular domain.

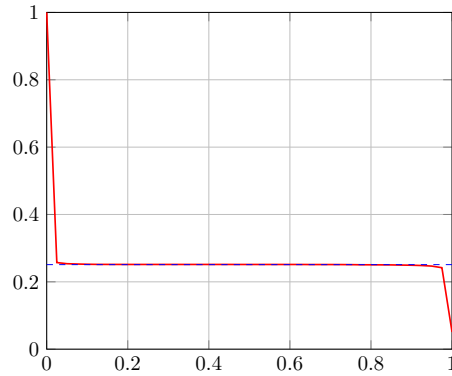


Figure 14: Thresholding function for the loaded knee

As in the first example, we have modeled only half the domain (see fig. 15). The design domain is half a circle, where the objective is to maximize the displacement u_{out} for an applied force $f_{in} = 1$, so we have now a *Mechanism* problem.

280 The spring stiffness coefficients are $k_{in} = 1$ and $k_{out} = 0.001$; there is also a support on the top and a roller support at the bottom (for symmetry assumptions), and a passive zone in the part of the annulus depicted in the figure. The volume fraction is $V^* = 0.25$ and we use the same elastic constants as in the previous examples.

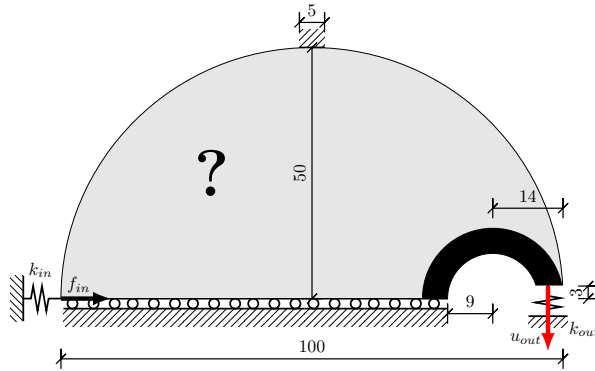


Figure 15: Design domain and dimensions for the circular gripper

285 The mesh (`circgrip.mesh` file) we have used in this problem is shown in

figure 16. The boundary parts have been highlighted in black for the roller support, red for the applied load and the top support, and green where the displacements are measured. The magenta color corresponds to the passive zone.

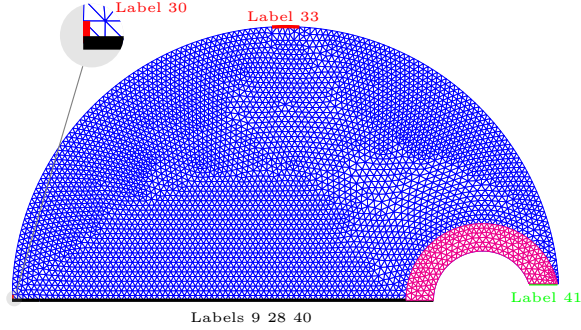


Figure 16: Mesh for the circular gripper with 8589 triangles

As before, all parameters can be read once we load the configuration file `circgrip.topt`. The spring supports are modeled as **Robin Boundary Conditions**¹⁷ where

$$M_{in} = \begin{pmatrix} k_{in} & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{h}_{in} = (f_{in}, 0)$$

for the horizontal support, and

$$M_{out} = \begin{pmatrix} 0 & 0 \\ 0 & k_{out} \end{pmatrix}, \quad \mathbf{h}_{out} = \mathbf{0}$$

290 for the vertical one (see fig. 3c). Also, we need to introduce the **Mechanism Objective**, where maximizing the displacement in the direction u_{out} corresponds to consider the vector $\mathbf{L} = (0, 1)$, that is, minimizing the opposite displacement.¹⁸

Finally note how the passive zone is introduced in fig. 3e (label 35) and the top and bottom roller supports, with corresponds to labels 33 (for top) and 9, 28

¹⁷ $M\mathbf{u} + \sigma(\mathbf{u}) \cdot \mathbf{n} = \mathbf{h}$.

¹⁸In this case, the objective function is $\int_{\Gamma_L} \mathbf{L} \cdot \mathbf{u}$.

295 and 40 (for bottom), respectively. In this case, we have used the Helmholtz filter with $r = 1$ and IPOPT with 4 loops of Heaviside projection and an update factor of 2. Figure 17 shows the complete optimal design, which has been generated with ParaView using the exported solution in VTK format.

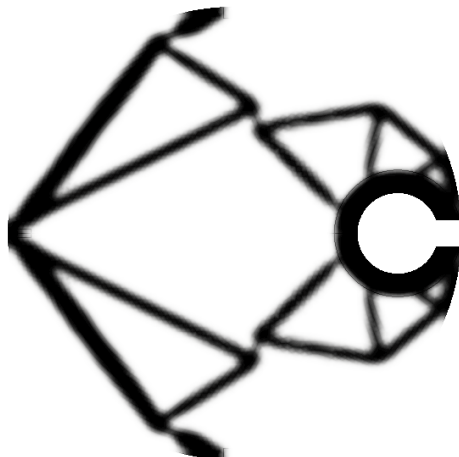


Figure 17: Optimal design for the circular gripper

4.5. A multiload case

300 Toptimiz3D is able to solve compliance *Multiload* problems like the one defined for the design domain shown at fig. 18 (cf. [33]). Here we consider the minimization of a weighted average of the compliances for each load \mathbf{f}_i , where $\mathbf{f}_1 = (0, 1)$ and $\mathbf{f}_2 = (0, -1)$. The material constants, SIMP penalization factor and iterations are the same as before, the volume fraction is $V^* = 0.2$, the
 305 conic filter radius is 6 and the weights for the loads are 0.5. In this case, the loads must be introduced through the **Boundary Multi Loads** panel from the **Boundary Conditions** submenu, where \mathbf{f}_1 is acting on the boundary labeled with number 10, \mathbf{f}_2 is acting on label 6, and the structure is clamped at label 14. We have used MMA with 4 loops of Heaviside projections and an update
 310 factor of 3. The mesh (`square-mfem.mesh` file) is shown at fig. 18b and the

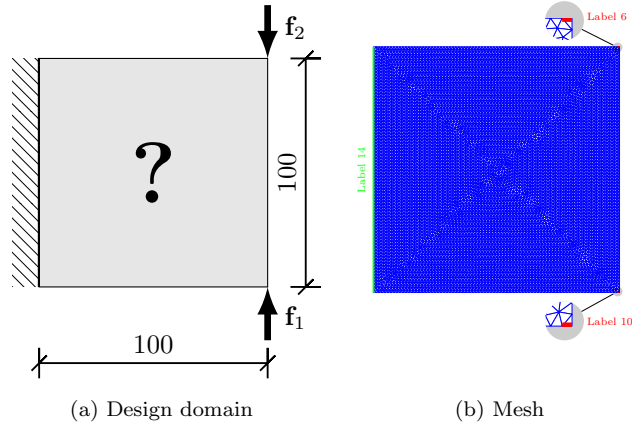


Figure 18: Design domain and mesh for the multiload example

configuration file `multiload.topt` are in the `examples` folder. Result is shown at fig. 19.

4.6. Stress constrained problem

We have also implemented a volume minimization problem under a stress
 315 constraint. Namely, we have used the global P -norm of the relaxed stress as
 it was introduced in [27], that is, $\sigma_r = \bar{\rho}^q \sigma_{VM}$, where σ_{VM} is the Von Mises
 stress computed using the elasticity tensor corresponding to the solid material



Figure 19: Result for the multiload problem

and the displacements obtained as the solution of the elasticity system (with the elasticity tensor defined by the SIMP interpolation). Here q is a relaxation parameter that can be defined by the user in the **Stress Parameters** panel at 320 `Parameters` menu along with P .

As in [27], we use the stress constraint $c\sigma_{PN} \leq \bar{\sigma}$ where $\bar{\sigma}$ is the maximum stress allowed (introduced in the **Optimization's Parameters** panel at `Parameters` menu, $\sigma_{PN} = \|\sigma_r\|_P$ and c is a constant which varies in each iteration according to

$$c^I = \alpha^I \frac{\sigma_{\max}^{I-1}}{\sigma_{PN}^{I-1}} + (1 - \alpha^I)c^{I-1}$$

The parameter α^I is supposed to be a control on the variations between iterations of c^I , however there were no information about that in [27]. From our experiments we have found that using $\alpha^I = 1$ for $I \leq I_0$ and $\alpha^I = 0$ for $I > I_0$, 325 for a given I_0 , works well. We name I_0 as *Threshold iterations* and it can be defined at **Stress Parameters** panel.

We have tested this algorithm with the loaded knee problem (example 4.2), using the same parameters except the maximum number of iterations which now is 2000. The specific stress parameters are $P = 24$, $q = 0.5$ and $I_0 = 20$, and the 330 maximum stress allowed $\bar{\sigma} = 2.4$. The obtained result is shown at fig. 20. The obtained volume is 19.84% and MND is 2.78% although we are using the same projection's parameters as in the example 4.2.

Fig. 21 shows a comparison of the stress distribution of the optimal solution of Example 4.2, without stress constraints, and this one, with stress constraints. 335 Fig. 21a corresponds to the relaxed Von Mises stress of the solution of fig. 9, where the maximum stress (in the re-entrant corner) is 4.4. The same scale has been used for showing the stress distribution in this example, fig. 21b, showing the avoidance of stress concentration areas and a very uniform stress distribution.

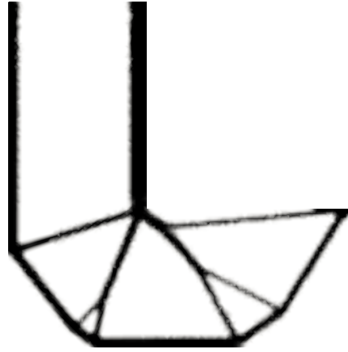


Figure 20: Result for the constrained stress problem

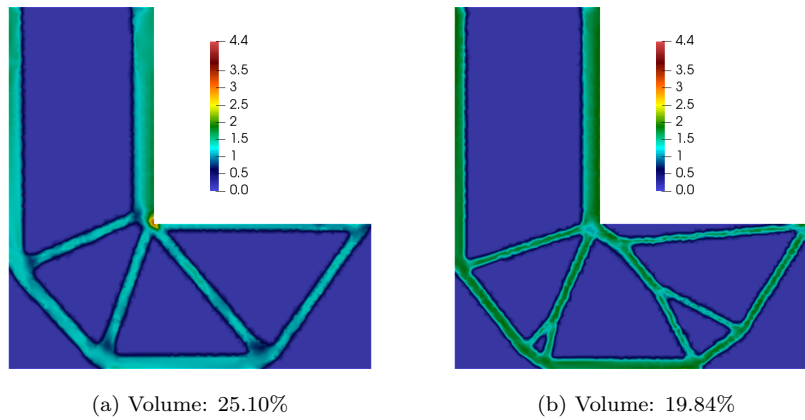


Figure 21: Stress comparison

340 4.7. An airplane bearing bracket

The last example was motivated by a challenge¹⁹ proposed by *Alcoa Fastening Systems & Rings* and *GRABCAD* enterprises and it was studied in [34] with slightly differences (the screw holder rectangular base from the original domain was changed by an X shaped piece). The domain design is shown at fig. 23a.

¹⁹<https://grabcad.com/challenges/airplane-bearing-bracket-challenge>

345 We have assumed an oblique tension on the bearing and clamped supports on
the accommodation of the screws (see figure 22). As in previous cases, we have
only modeled half the domain.

Result, that took around 12 hours of computational time for a mesh with
450452 tetrahedra in an standard personal computer, is shown at figs. 23(b)–(d).

350 5. Conclusions and final remarks

In this paper we present `Toptimiz3D`, a topology optimization software with
a graphical user interface for setting up and solving structural topology op-
timization problems. Possible problems are compliance optimization, in the
simple and multiload cases, volume minimization under compliance constraints,
355 stress-based optimization and optimal design of mechanisms. It can handle with
general geometries and unstructured meshes in both 2d and 3d. `Toptimiz3D`
permits the use of several optimization algorithms (MMA, IPOPT, optimal-
ity conditions) and numerical smoothing and projection techniques. We have
shown several case studies exhibiting different geometries, more specifically the
360 last example, to highlight one of the strengths of our software. As it has been
explained along the paper, the program allows the user to define a mesh and
set up the problem selecting different options to solve it directly through the
interface. For further improvement of the software, our effort in the near future
will be mainly placed on speeding up the code in 3d by using parallel computing
365 in order to incorporate this functionality. Some preliminary tests obtained to
date are very promising in terms of computational time and we will expect to
report our results soon.

Acknowledgements

This work has been supported by the Ministry of Science, Innovation and
370 Universities-State Research Agency (SPAIN) through project MTM2017-83740-
P.

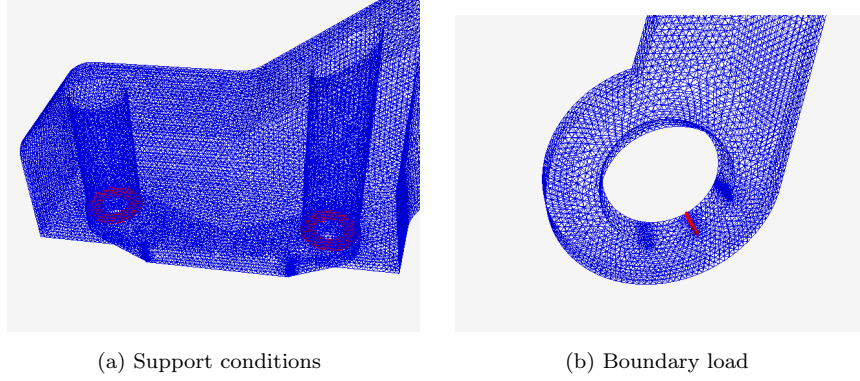


Figure 22: Input data for the airplane bearing bracket

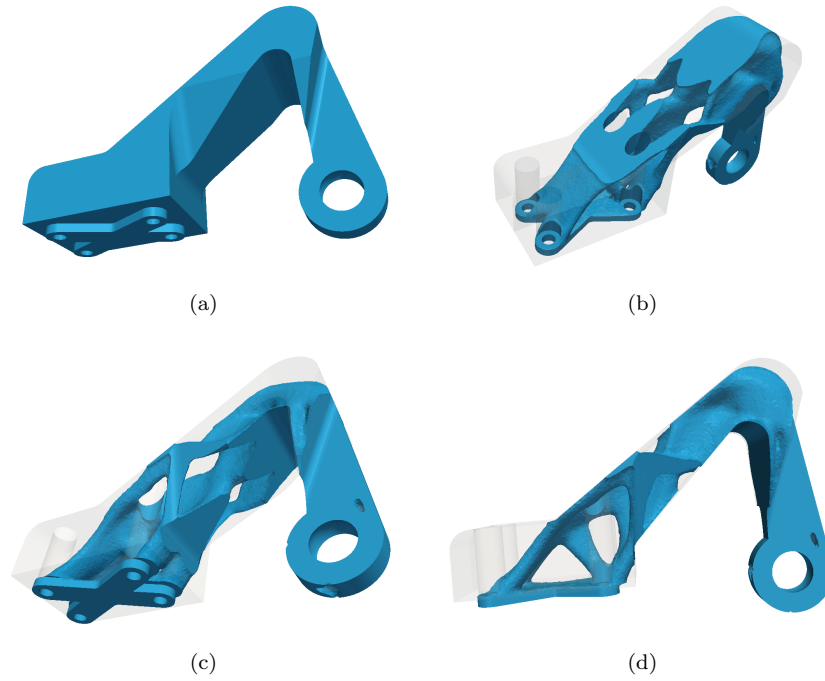


Figure 23: Domain (fig. (a)) and result (figs. (b)-(d)) for the airplane bearing bracket

Additional note

On behalf of all authors, the corresponding author states that there is no conflict of interest.

375 Replication of results

The examples shown in this paper can be reproduced with the software available at <https://gitlab.com/e-aranda/topt-mfem>.

References

- [1] M. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design
380 using a homogenization method, *Computer Methods in Applied Mechanics and Engineering* 71 (2) (1988) 197–224.
- [2] M. Bendsøe, *Optimization of structural topology, shape, and material*, Springer-Verlag, 1995.
- [3] G. Allaire, *Shape Optimization by the Homogenization Method*, Springer-
385 Verlag, 2002.
- [4] M. P. Bendsøe, Optimal shape design as a material distribution problem, *Structural optimization* 1 (4) (1989) 193–202.
- [5] M. Y. Wang, X. Wang, D. Guo, A level set method for structural topology
390 optimization, *Computer Methods in Applied Mechanics and Engineering* 192 (1) (2003) 227 – 246.
- [6] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, *Journal of Computational Physics* 194 (1) (2004) 363 – 393.
- [7] N. P. van Dijk, K. Maute, M. Langelaar, F. van Keulen, Level-set methods
395 for structural topology optimization: a review, *Structural and Multidisciplinary Optimization* 48 (3) (2013) 437–472.

- [8] M. Bendsøe, O. Sigmund, *Topology Optimization: theory, methods and applications*, Springer-Verlag, 2003.
- [9] J. Jensen, O. Sigmund, Topology optimization for nano-photonics, *Laser & Photonics Reviews* 5 (2) (2011) 308–321.
- 400 [10] O. Sigmund, K. Maute, Topology optimization approaches, *Structural and Multidisciplinary Optimization* 48 (6) (2013) 1031–1055.
- [11] B. S. Lazarov, F. Wang, O. Sigmund, Length scale and manufacturability in density-based topology optimization, *Archive of Applied Mechanics* 86 (1) 405 (2016) 189–218.
- [12] K. Liu, A. Tovar, An efficient 3d topology optimization code written in matlab, *Structural and Multidisciplinary Optimization* 50 (6) (2014) 1175–1196.
- [13] N. Aage, E. Andreassen, B. S. Lazarov, Topology optimization using PETSC: An easy-to-use, fully parallel, open source topology optimization 410 framework, *Structural and Multidisciplinary Optimization* 51 (3) (2015) 565–572.
- [14] O. Sigmund, A 99 line topology optimization code written in matlab, *Structural and Multidisciplinary Optimization* 21 (2) (2001) 120–127.
- 415 [15] G. Allaire, O. Pantz, Structural optimization with freefem++, *Structural and Multidisciplinary Optimization* 32 (3) (2006) 173–181.
- [16] V. J. Challis, A discrete level-set topology optimization code written in matlab, *Structural and Multidisciplinary Optimization* 41 (3) (2010) 453–464.
- 420 [17] E. Andreassen, A. Clausen, M. Schevenels, B. Lazarov, O. Sigmund, Efficient topology optimization in matlab using 88 lines of code, *Structural and Multidisciplinary Optimization* 43 (2011) 1–16.

- [18] C. Talischi, G. H. Paulino, A. Pereira, I. F. M. Menezes, Polytop: a matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes, *Structural and Multidisciplinary Optimization* 45 (3) (2012) 329–357.
- [19] Lawrence Livermore National Laboratory, MFEM: Modular Finite Element Methods Library, mfem.org. doi:10.11578/dc.20171025.1248.
- [20] W. J. Schroeder, B. Lorensen, K. Martin, The visualization toolkit: an object-oriented approach to 3D graphics, Kitware, 2004.
- [21] U. Ayachit, The paraview guide: A parallel visualization application (2015).
- [22] B. Bourdin, Filters in topology optimization, *International Journal for Numerical Methods in Engineering* 50 (9) (2001) 2143–2158.
- [23] B. S. Lazarov, O. Sigmund, Filters in topology optimization based on Helmholtz-type differential equations, *Internat. J. Numer. Methods Engrg.* 86 (6) (2011) 765–781.
- [24] J. K. Guest, J. H. Prévost, T. Belytschko, Achieving minimum length scale in topology optimization using nodal design variables and projection functions, *International Journal for Numerical Methods in Engineering* 61 (2) (2004) 238–254.
- [25] F. Wang, B. S. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, *Structural and Multidisciplinary Optimization* 43 (6) (2011) 767–784.
- [26] S. Xu, Y. Cai, G. Cheng, Volume preserving nonlinear density filter based on heaviside functions, *Structural and Multidisciplinary Optimization* 41 (4) (2010) 495–505.
- [27] C. Le, J. Norato, T. Bruns, C. Ha, D. Tortorelli, Stress-based topology optimization for continua, *Structural and Multidisciplinary Optimization* 41 (2010) 605–620.

- [28] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, *Internat. J. Numer. Methods Engrg.* 24 (2) (1987) 359–373.
- [29] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming. A Publication of the Mathematical Programming Society* 106 (1, Ser. A) (2006) 25–57.
- [30] P. Frey, *Medit: An interactive mesh visualization software*, Ph.D. thesis, INRIA (2001).
- [31] O. Sigmund, Morphology-based black and white filters for topology optimization, *Structural and Multidisciplinary Optimization* 33 (4-5) (2007) 401.
- [32] A. Ribes, A. Bruneton, A. Geay, *Salome: an open-source simulation platform integrating paraview*, Tech. rep. (07 2017).
- [33] O. Sigmund, A 99 line topology optimization code written in matlab, *Structural and multidisciplinary optimization* 21 (2) (2001) 120–127.
- [34] J. Urruchi-Mohíno, *Optimal design of flap supporting structure for a light aircraft (in spanish)*, Bachelor Thesis, ETSII-CR, UCLM, 18-1- 225485 (2018).