XXXVII IBERIAN LATIN AMERICAN CONGRESS
ON COMPUTATIONAL METHODS IN ENGINEERING
BRASÍLIA - DF - BRAZIL

Topology optimization with local stress constraint on arbitrary polygonal meshes using a damage approach

Fernando V. da Senhora
Ivan F. M. de Menezes
Anderson Pereira

fernandosenhora@hotmail.com
ivan@puc-rio.br
anderson@tecgraf.puc-rio.br
Department of Mechanical Engineering,
Pontifical Catholic University of Rio de Janeiro (PUC-Rio),
Rua Marquês de São Vicente, 225, 22453-900, Rio de Janeiro, RJ, Brazil

Glaucio H. Paulino

paulino@gatech.edu
School of School of Civil and Environmental Engineering,
Georgia Institute of Technology (GATech),
790 Atlantic Drive, Atlanta, GA 30332, USA

Abstract. *This work addresses topology optimization with stress constraints using the damage approach by Verbart et al. (2016) through a polygonal element discretization in the spirit of the* `PolyTop` *code (Talischi et al., 2012b). In order to limit the maximum stress on the final structure, material in overstressed regions is considered damaged and so contributes less to the overall stiffness. Local stress constraints are replaced by one constraint requiring that the damaged and undamaged models have the same compliance. This drastically reduces the computational cost associated with the large number of constraints. Following the* `PolyTop` *philosophy, we developed a user-friendly MATLAB code using polygonal elements and the SIMP formulation for material interpolation. The popular L-bracket benchmark problem is solved to evaluate the effectiveness and efficiency of the method. The optimization is performed with the MMA algorithm (Svanberg, 1987). We conclude by making some comparative remarks about existing formulations and the need to properly address local stress constraints.*

Keywords: *Topology optimization, Stress constraints, Damage approach, Polygonal elements*

1 INTRODUCTION

Structure optimization is a promising field that has been gaining popularity due to the advances in manufacturing techniques, specially 3D printing. Despite its first studies date to the 1940s there is still a lot of space for improvements. This paper focus on topology optimization, a technique that uses a discrete version of the problem to find the optimal disposition of material in some aspect.

Traditional topology optimization approach of minimizing the compliance of a structure with volume constraint can lead to stress concentration in the final design. Such design would not match the limitations of material properties as it is physically requested granting it infeasible. In order to obtain a result closer to a valid final design, stress constraints must be incorporated in the formulation.

Two main challenges emerge while dealing with this constraints. The large number of constraints associated with stresses due to their local properties, and the singularity of optimal solution while using a density approach. As an element density goes to zero, its stress rises greatly which is not consistent with the physical reality since it does not make sense associating stresses to a void region.

In the past years many authors have proposed a variety of alternatives for dealing with those matters. The techniques used to face the large number of constraints can be divided in three main ideas: Assembling techniques in which the stress constraint are aggregated using a continuous function to mimic the maximum and so greatly reduces the number of constraints. Holmberg et al. (2013) use the p-norm strategy and Yang and Chen (1996) adopt the Kreisselmeier-Steinhauser function. The main drawback of this idea is that it loses the local characteristic of the stress. Le et al. (2009) and Pars et al. (2010) try to mitigate this problem by using regional clustering. The use of the augmented Lagrangian to eliminate the problem's constraints is presented in Pereira et al. (2004). Emmendoerfer Jr. and Fancello (2014) use this approach with level set function for material interpolation. Despite leading to a better interpretation of the constraints locality, it requires progressive update on optimization parameters, which can be time consuming. The active-set course of action only considers the constraints that are close to being violated. This raises a lot of concerns. It may not be viable for a large mesh and by the end of the optimization a large number of constraints may be active. Despite this Guo et al. (2011) were able to get some results.

With respect to the singularity issue the most popular approach is to use a relaxed version of the stress that assures that the constraint tends to zero as the density is reduced. Among the various relaxation proposals it is worth mentioning the qp-relaxation (Bruggi, 2008) and ε-relaxation (Cheng and Guo, 1997) that became very popular in recent works.

In this article both problems are dealt with the damage approach proposed by Verbart et al. (2016) considered an assembling technique. A monotonic performance function is chosen, in this case the compliance, to evaluate the entire structure. Then a damaged version of the structure is composed where regions with stresses above the admissible value are considered damaged and will contribute less to the overall stiffness. One constraint is introduced requiring that the compliance of both damaged and undamaged structures have the same value, and therefore a final design with reduced stress is obtained. The singular optima is naturally present in the solution space since regions with no material naturally don't exhibit damage.

Furthermore the main contributions of this work are the development of a MATLAB code and the use polygonal elements in the finite element analysis and topology optimization design with the damage approach. Some of the benefits of this type of mesh is the elimination of checkerboard pattern and a better representation of arbitrary domains. Talischi et al. (2010) explain thoroughly the concerns of using polygonal elements in topology optimization. The code retains the readability and efficiency of previous educational codes (Talischi et al., 2012a, Talischi et al., 2012b and Pereira et al., 2016).

The remainder of this paper is organized as follows: Section 2 gives a detailed explanation of the problem formulation. Section 3 presents the sensitivity derivation of the objective and constraint of the optimization problem. Section 4 gives a better explanation on the use of polygonal elements. Section 5 addresses the implementation issues. Section 6 shows some numerical results and section 7 presents some conclusions and considerations on stress constraints.

2 Problem Formulation

The main objective of topology optimization is to find the best material distribution for a physical system. To achieve this goal, is necessary to have a measure of quality in order to classify a shape as better or worse than another. For structural design the compliance is traditionally chosen as measure to be minimized. However for the stress constrained version of the problem it is a common practice to use the total weight. Therefore the problem is to find the least amount of material necessary to satisfy the stress constraints. In other words find the lightest structure able to withstand the applied load.

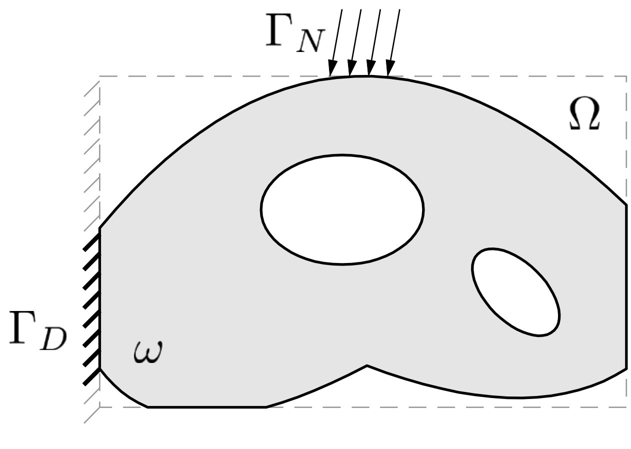


Figure 1: Design domain and boundary conditions. [Image from Talischi et al. (2012b)]

Figure 2 shows a representation of the problem. In Eq. (1) σ_ω represents the stresses over the structure ω and ρ its density. The topology optimization problem can be expressed as:

$$\begin{aligned} \underset{\omega}{\text{minimize}} \quad & M = \int_\omega \rho d\omega \\ \text{subject to} \quad & \frac{\sigma_\omega}{\sigma_{lim}} - 1 \leqslant 0 \end{aligned} \tag{1}$$

where M is the total weight of the structure.

The model is subjected to linear elasticity equations which means that the displacements U_ω must satisfy the variational problem in Eq. (2), where *D* is the constitutive tensor of the material and *t* is the load applied as a boundary condition:

$$\int_{\omega} \mathbf{D}\nabla\mathbf{U}_\omega : \nabla\mathbf{v}d\omega = \int_{\Gamma_N} \mathbf{t}\cdot\mathbf{v}d\Gamma_N \tag{2}$$

Considering the plane stress case, the constitutive tensor $\mathbf{D}$ is described in Eq. (3), where ν is the Poisson ratio and E_0 is the material Young's modulus:

$$\mathbf{D} = \frac{E_0}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \tag{3}$$

The nature of the problem is continuous but to make it manageable with the tools available today a discrete version is formulated.The original domain is discretized by finite element mesh and a design variable, denoted by *z*, is associated to each element, with the unit value representing a material region and zero a void region. This combinatorial problem is ill-posed and therefore, can be really troublesome to solve especially for a large number of variables. In order to use gradient based optimizers and, as consequence, be able to reduce the computational cost, a material interpolation with penalization formulation called SIMP (Bendsøe, 1989) is used. The design variables are allowed to vary continuously in the interval $[0, 1]$.

The design variables relate to the element density through a linear "hat" filter. The use of a filter regularizes the problem and allows certain control over the minimum size of artifacts in the final design . Furthermore it makes the problem seemingly convergent over mesh refinement. The discrete filter can be assembled as a matrix that is represented as $\mathbf{P}$ in Eq. (4) (refer to Bourdin (2001) and Borrvall and Petersson (2001) for more information on filtering) :

$$\rho(\mathbf{z}) = \mathbf{P}\mathbf{z} \tag{4}$$

The design's total weight, used as the objective function, can be easily calculated through Eq. (5), where factor v_i is the volume of each element (or the area in 2D problems):

$$M(\mathbf{z}) = \sum_{i}^{nElem} \rho_i(\mathbf{z})v_i \tag{5}$$

To avoid elements with density values in between zero and one, which don't have a strong physical meaning unless one is dealing with composite materials, a penalization factor, $p > 1$, is introduced, as shown in Eq. (6), where ε is a small positive value following the ersatz approach to avoid a singular stiffness matrix and E is the material interpolation function that relates the value of $(\mathbf{z})$ at a point to the stiffness at that point.

$$E(\mathbf{z}) = (1-\varepsilon)\left[\rho(\mathbf{z})\right]^p + \varepsilon \tag{6}$$

The element, denoted by the subindex "el", stiffness matrix is defined as:

$$\mathbf{k_{el}} = \int_\Omega E(\mathbf{z})\mathbf{B_{el}^T D B_{el}} d\Omega \tag{7}$$

where $\mathbf{B_{el}}$ is the strain displacement matrix as it follows from the finite element method. The assembling of the global stiffness matrix is then straightforward.

The variational problem in Eq. (2) in its weak form is then transformed in a simple algebraic linear system of equations involving the global stiffness matrix $(\mathbf{K})$ and the applied load vector $(\mathbf{F})$.

$$\mathbf{KU} = \mathbf{F} \tag{8}$$

The stress measure of an element is defined as the p-mean of the von Mises stress evaluated at its Gauss integration points. Every time stress is mentioned through out this paper this is the meaning refereed to unless stated otherwise. Evaluating the stress at the Gauss points ensures a lower numerical error. The p-mean is described in Eq. (9) and is used here because it is a smooth approximation of the maximum function. In Eqs. (9) to (12), σ is the element stress, σ^{V_M} is the von Mises stress at the Gauss points, $\sigma^{\mathbf{Gauss}}$ are the Cauchy stress components at the gauss points, $\mathbf{V}$ is the von Mises matrix, and $\mathbf{U}$ are the displacements at equilibrium state.

$$\sigma = \left[\sum_i^{nGauss} \frac{\left(\sigma_i^{V_M}\right)^p}{nGauss}\right]^{1/p} \tag{9}$$

$$\sigma^{V_M} = \sqrt{\sigma^{\mathbf{Gauss}}\mathbf{V}\sigma^{\mathbf{Gauss}}} \tag{10}$$

$$\mathbf{V} = \begin{bmatrix} 1 & -0.5 & 0 \\ -0.5 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \tag{11}$$

$$\sigma^{\mathbf{Gauss}} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \mathbf{DBU_{el}} \tag{12}$$

$\mathbf{U_{el}}$ denotes the displacement's degrees of freedom of the element "**el**".

The discrete version of the optimization problem (see Eq. (1)) is presented in Eq. (13). Notice there is one stress constraint for each finite element:

$$\begin{aligned} \underset{\mathbf{z}}{\text{minimize}} \quad & M(\mathbf{z}) = \sum_{i}^{nElem} \rho_i(\mathbf{z}) v_i \\ \text{subject to} \quad & \frac{\sigma_i(\mathbf{z})}{\sigma_{lim}} - 1 \leqslant 0 \qquad i = 1 \cdots nElem \\ & 0 \leqslant \mathbf{z} \leqslant 1 \\ \text{with} \quad & \mathbf{KU} = \mathbf{F} \end{aligned} \tag{13}$$

2.1 Damage Model

The main drawback of the formulation presented in Eq. (13) is the large number of constraints. The more refined is the finite element mesh the higher is the computational cost to solve the problem. For this reason a different approach, called damage approach, has been proposed by Verbart et al. (2016).

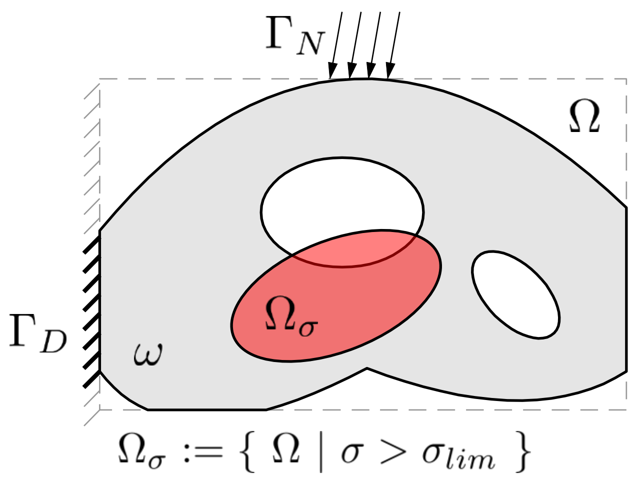


Figure 2: Representation of the damage model.

In the damage approach, a second model of the structure is evaluated where regions with stress values above a certain limit are considered damaged and so contribute less to the overall stiffness. A constraint is then introduced to impose that the compliance of the damage and undamaged models have the same value. The idea behind this approach is to lead the solution to a final design with no damage , i.e., with no stress violation. The damage model is merely a mathematical artifice and not a real representation of the physical damage. All variables marked with a tilde ($\widetilde{\ }$) refer to the damage model.

The material interpolation function for the stiffness of the damaged elements defined such that it respects the following conditions:

$$\begin{cases} \widetilde{E}(\mathbf{x}) < E(\mathbf{x}), & \forall \mathbf{x} \in \Omega_\sigma := \left\{\mathbf{x} \mid \sigma(\mathbf{x}) > \sigma_{lim}\right\} \\ \widetilde{E}(\mathbf{x}) = E(\mathbf{x}), & \forall \mathbf{x} \in \Omega \setminus \Omega_\sigma \end{cases} \tag{14}$$

Therefore the damaged material interpolation function that depends not only on $\mathbf{z}$ but also on the stress σ is related to the actual material interpolation function by:

$$\widetilde{E}(\mathbf{z}, \sigma) = \varepsilon + \beta(E(\mathbf{z}) - \varepsilon) \tag{15}$$

where β is a function that is equal to 1 if the stress is below the limit and monotonically decreasing otherwise. The choice of β in this paper is represented in Eq. (16), where the parameter α controls the steepness of function β as illustrated in Fig. 3.

$$\beta = \begin{cases} 1, & \text{if } \sigma \leqslant \sigma_{lim} \\ e^{-\alpha(\sigma/\sigma_{lim}-1)^2}, & \text{otherwise} \end{cases} \tag{16}$$

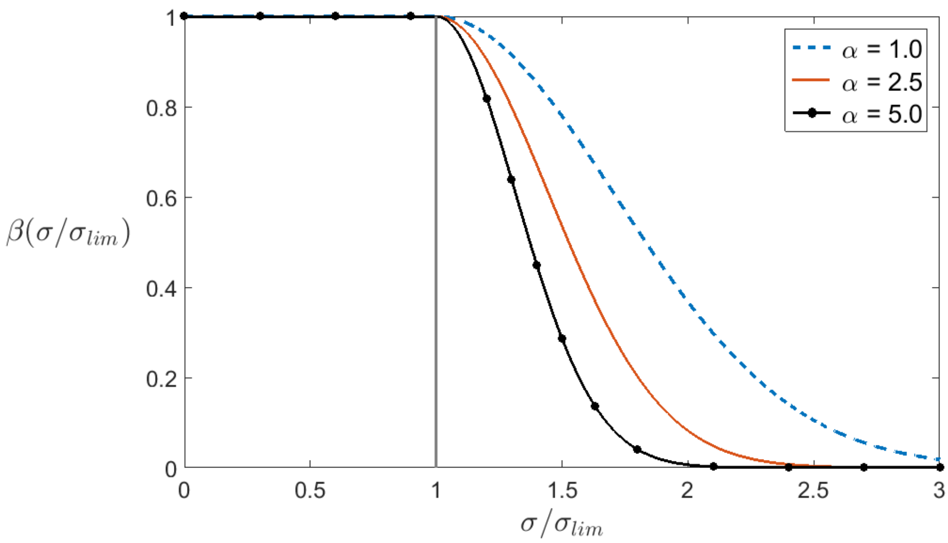


Figure 3: Influence of the numerical parameter α in the function β.

The compliance can be interpreted as the total work necessary to achieve an equilibrium state considering the boundary conditions,i.e.:

$$C = \int_{\Omega} U(\mathbf{x}) \cdot t(\mathbf{x}) d\mathbf{x} \tag{17}$$

Where C is the compliance, $U(\mathbf{x})$ are the displacements and $t(\mathbf{x})$ is the load applied.

Once the two models are assembled, the respective compliances are calculated. In the discrete form the integral in Eq. (17) can be replaced by the dot product between the global displacement and the external applied load vectors:

$$C = \mathbf{U}^T\mathbf{F} = \mathbf{U}^T\mathbf{K}\mathbf{U} \tag{18}$$

The compliance of the damage model is calculated in a similar way and will always be greater or equal to the undamaged one. Striving for a damage free final result leads to the formulation of the constraint in Eq. (19), where the parameter δ is a small positive value and is introduced because without it the constraint would be too strict and so be troublesome for traditional optimization algorithms:

$$g(\mathbf{z}) = \frac{\widetilde{C}}{C} - 1 - \delta \leqslant 0 \tag{19}$$

The new optimization problem can be expressed as:

$$\begin{aligned} \underset{\mathbf{z}}{\text{minimize}} \quad & M(\mathbf{z}) = \sum_{i}^{nElem} \rho_i(\mathbf{z}) v_i \\ \text{subject to} \quad & \frac{\widetilde{C}}{C} - 1 - \delta \leqslant 0 \\ & 0 \leqslant \mathbf{z} \leqslant 1 \\ \text{with} \quad & \mathbf{K}(\mathbf{z})\mathbf{U} = \mathbf{F} \end{aligned} \tag{20}$$

The advantage of this formulation is that it presents only one constraint that for some extent represents all the constraints in Eq. (13). This makes the problem substantially more manageable.

3 Sensitivity Analysis

The solution of the optimization problem (20) is usually obtained by means of gradient based methods such as the method of moving asymptotes (MMA) by Svanberg (1987). Therefore, they depend on the first order derivative of the objective and constraint functions over the optimization variables.

The sensitivity calculations are described below. To clarify the equations Einstein's tensor notation is used and so repeated index are summed over. As this system is governed by uniform Euclidean metric, counter-variant and co-variant tensors are interchangeable and all index are displayed subscripted.

The objective function is the total weight of the resulting structure and it is linear with respect to the optimization variables. Deriving the Eq. (5):

$$\frac{dM(\mathbf{z})}{dz_i} = P_{ij}v_i \tag{21}$$

The constraint function is non-linear and extensive and so its computation is costly. For clarification the derivation is done in several steps. Applying the quotient rule of derivation on Eq. (19):

$$\frac{dg}{dz_i} = \frac{1}{C}\frac{d\widetilde{C}}{dz_i} - \frac{\widetilde{C}}{C^2}\frac{dC}{dz_i} \tag{22}$$

Remembering here the equilibrium equations that will be used to derive the sensitivities through the adjoint method:

$$K_{jl}U_l - F_j = 0 \qquad\qquad \widetilde{K}_{jl}\widetilde{U}_l - F_j = 0 \tag{23}$$

The next step is to derive the compliance of the damage and undamaged model, U_jF_j and $\widetilde{U}_jF_j$, respectively. In the adjoint method the equilibrium equation times the adjoint variable is added to the derivative, i.e.:

$$\frac{dC}{dz_i} = \frac{d\left(U_jF_j\right)}{dz_i} = \frac{d\left(U_j\right)F_j}{dz_i} = \frac{dU_j}{dz_i}F_j + \mu_j\frac{d(K_{jl}U_l - F_j)}{dz_i} \tag{24}$$

Expliciting the terms with $\frac{dU_j}{dz_i}$:

$$\frac{dC}{dz_i} = \frac{dU_j}{dz_i}(F_j + \mu_jK_{jl}) + \mu_j\frac{dK_{jl}}{dz_i}U_l \tag{25}$$

The value of μ is then chosen such that the first term will be eliminated. Recalling that the stiffness tensor K_{jl} is symmetric and the equivalence in Eq. (23), is easy to realize that $\mu_j = -U_j$. Replacing it in Eq. (25), and considering that F_j does not depend on the design z:

$$\frac{dC}{dz_i} = \frac{dU_j}{dz_i}(F_l - U_jK_{jl}) - U_j\frac{dK_{jl}}{dz_i}U_l = -U_j\frac{dK_{jl}}{dz_i}U_l \tag{26}$$

In the matrix notation this expression is written as $C = U^T K U$ matching the literature (Bendsoe and Sigmund, 2003). Similarly the sensitivity of the damage compliance is calculated as:

$$\frac{d\widetilde{C}}{dz_i} = \frac{d\widetilde{U}_j}{dz_i} F_j = \frac{d\widetilde{U}_j}{dz_i} F_j + \lambda_j \frac{d(K_{jl}U_l - F_j)}{dz_i} + \mu_j \frac{d(\widetilde{K}_{jl}\widetilde{U}_l - F_j)}{dz_i} \tag{27}$$

Since $\widetilde{K}$ is a function of both z and U it is necessary to use the chain rule of derivation:

$$\begin{aligned} \frac{d\widetilde{C}}{dz_i} =& \frac{d\widetilde{U}_j}{dz_i} F_j + \lambda_j \left(\frac{dK_{jl}}{dz_i} U_l + K_{jl} \frac{dU_l}{dz_i} \right) + \\ & + \mu_j \left(\widetilde{K}_{jl} \frac{d\widetilde{U}_l}{dz_i} + \left. \frac{d\widetilde{K}_{jl}}{dz_i} \right|_{U=const.} \widetilde{U}_l + \widetilde{U}_l \frac{d\widetilde{K}_{jl}}{dU_m} \frac{dU_m}{dz_i} \right) \end{aligned} \tag{28}$$

Collecting the terms with $\left. \frac{d\widetilde{U}}{dz} \right|_{U=const.}$ and $\frac{dU}{dz}$:

$$\begin{aligned} \frac{d\widetilde{C}}{dz_i} =& \left(F_l + \mu_j \widetilde{K}_{jl} \right) \left. \frac{d\widetilde{U}_l}{dz_i} \right|_{U=const.} + \mu_j \left. \frac{d\widetilde{K}_{jl}}{dz_i} \right|_{U=const.} \widetilde{U}_l + \\ & + \left(\lambda_j K_{jl} + \mu_j \frac{d\widetilde{K}_{jl}}{dU_m} \widetilde{U}_l \right) \frac{dU_m}{dz_i} + \lambda_j \frac{dK_{jl}}{dz_i} U_l \end{aligned} \tag{29}$$

As in Eq. (26) we choose $\mu_j = -\widetilde{U}_j$ in order to eliminate the terms in brackets:

$$\frac{d\widetilde{C}}{dz_i} = \mu_j \left. \frac{d\widetilde{K}_{jl}}{dz_i} \right|_{U=const.} \widetilde{U}_l + \left(\lambda_j K_{jl} - \widetilde{U}_j \frac{d\widetilde{K}_{jl}}{dU_m} \widetilde{U}_l \right) \frac{dU_m}{dz_i} + \lambda_j \frac{dK_{jl}}{dz_i} U_l \tag{30}$$

The other adjoint variable is a little bit more complicated. Focusing on the term inside the brackets multiplying $\frac{dU_m}{dz_i}$ that must be eliminated we have:

$$\lambda_j K_{jl} - \widetilde{U}_j \frac{d\widetilde{K}_{jl}}{dU_m} \widetilde{U}_l = 0 \tag{31}$$

The value of λ_j can be obtained by solving the system:

$$\lambda_j K_{jm} = \widetilde{U}_j \frac{d\widetilde{K}_{jt}}{dU_m} \widetilde{U}_t \tag{32}$$

The derivative that appears in the right hand side term of Eq. (32) can be expanded as:

$$\frac{d\widetilde{K}_{jt}}{dU_m} = \frac{d\widetilde{K}_{jt}}{d\widetilde{E}_s}\frac{d\widetilde{E}_s}{d\beta_q}\frac{d\beta_q}{d\sigma_r}\frac{d\sigma_r}{dU_m} \tag{33}$$

Although Eq. (33) seems complicated, most of the terms correspond to diagonal tensors and are very simple to calculate, as presented below:

$$\frac{d\widetilde{K}_{jt}}{d\widetilde{E}_s} = \mathbf{k} \tag{34}$$

Where $\mathbf{k}$ is stiffness matrix of the element s ;

$$\frac{d\widetilde{E}_s}{d\beta_q} = \begin{cases} 0, & \text{if } s \neq q \\ E_s + \varepsilon, & \text{otherwise;} \end{cases} \tag{35}$$

$$\frac{d\beta_q}{d\sigma_r} = \begin{cases} 0, & \text{if } q \neq r \\ 0, & \text{if } (q = r \text{ and } \sigma_q < \sigma_{lim}) \\ -\dfrac{2\alpha}{\sigma_{lim}}(\sigma_q/\sigma_{lim} - 1)e^{-\alpha(\sigma_q/\sigma_{lim}-1)^2}, & \text{otherwise} \end{cases} \tag{36}$$

Again the derivation of the stress tensor with respect to the displacements U_m is given by:

$$\frac{d\sigma_r}{dU_m} = \frac{d\sigma_r}{d\sigma_v^{V_M}}\frac{d\sigma_v^{V_M}}{d\sigma_w^{Gauss}}\frac{d\sigma_w^{Gauss}}{dU_m} \tag{37}$$

Where the index *r* runs through the elements, index *v* runs through the Gauss integration points, index *w* runs through the components of stress in the Gauss integration points and index *m* runs through the degrees of freedom.

$$\frac{d\sigma_r}{d\sigma_v^{V_M}} = \begin{cases} \dfrac{\left(\sigma_v^{V_M}\right)^{p-1}}{\sigma_r}, & \text{if Gauss point } v \text{ belongs to element } r \\ 0, & \text{otherwise} \end{cases} \tag{38}$$

$$\frac{d\sigma_v^{V_M}}{d\sigma_w^{Gauss}} = \begin{cases} V\dfrac{\sigma_w^{Gauss}}{\sigma_v^{VM}}, & \text{if Stress Component } w \text{ belongs to Gauss point } v \\ 0, & \text{otherwise} \end{cases} \tag{39}$$

$$\frac{d\sigma_w^{Gauss}}{dU_m} = \begin{cases} DB, & \text{if Stress Component } w \text{ and DOF } m \text{ belong to the same Element} \\ 0, & \text{otherwise} \end{cases} \tag{40}$$

It is important to mention that indexes *s, q, r, i* run through the elements, indexes *j, l, n, t, m, j* run through the degrees of freedom, index *v* runs through the Gauss Integration points and index *w* runs through the components of stress in the Gauss integration points.

Putting all together in Eq. (41) it seems quite long and complex. However with careful attention is possible to see that the whole right side contracts to a first order tensor, and so it is possible to find the value of λ by solving only one linear system which is the most computationally expensive part of calculations, i.e.:

$$\lambda_j K_{jm} = \widetilde{U}_j \frac{d\widetilde{K}_{jt}}{d\widetilde{E}_s} \frac{d\widetilde{E}_s}{d\beta_q} \frac{d\beta_q}{d\sigma_r} \frac{d\sigma_r}{d\sigma_v^{V_M}} \frac{d\sigma_v^{V_M}}{d\sigma_w^{Gauss}} \frac{d\sigma_w^{Gauss}}{dU_m} \widetilde{U}_t \tag{41}$$

Finally, simplifying the Eq. (29), one obtains :

$$\frac{d\widetilde{C}(z)}{dz_i} = -\widetilde{U}_j \left. \frac{d\widetilde{K}_{jl}}{dz_i} \right|_{U=cst} \widetilde{U}_l + \lambda_j \frac{dK_{jl}}{dz_i} U_l \tag{42}$$

the expression above is identical to the one presented in Verbart et al. (2016).

4 Polygonal Elements

The use of polygonal elements presents some advantages over the Q4 and other regular counterparts. First and more obvious it can be used over complex domains providing a good approximation without the need for advanced meshing techniques as it can be seen in Fig. 4. Moreover it naturally avoids the well known checkerboard and one-node connection problems.

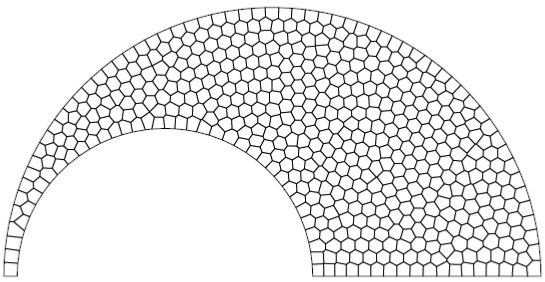


Figure 4: Polygonal mesh over complex geometry.

A more subtle matter is that structured meshes tend to have preferential directions. So the geometry of the final solution may be dependent on the geometry of the mesh. This type of mesh dependency can lead to suboptimal designs in topology optimization. Unstructured meshes tend to be a better representation for isotropic domains.

From the finite elements point of view this type of discretization leads to higher precision in the solution of linear elastic problems and are stable with respect to the Babuska-Brezzi condition (Pereira et al., 2016).

On the other hand, good unstructured meshes are harder to generate. The ones used in this paper are created using Voronoi diagrams and Loyd iterations as described in Talischi et al. (2012a).

For a more detailed discussion on the use of polygonal elements for topology optimization the reader must refer to Talischi et al. (2010).

5 Implementation

The implementation is done in MATLAB in the fashion of the freely available code `PolyTop` (Talischi et al., 2012b) that also features a polygonal mesh generator `PolyMesher` (Talischi et al., 2012a). In the same manner all input data and optimization parameter are stored in two structures `fem` and `opt`.

Following the philosophy of the previous code a modular framework is developed to allow easy modification and recycling of the implementation. The main difference is that now the objective and constraints computations are encapsulated inside a function in order to be used in optimization algorithms as sort the ones available in the optimization toolbox in MATLAB.

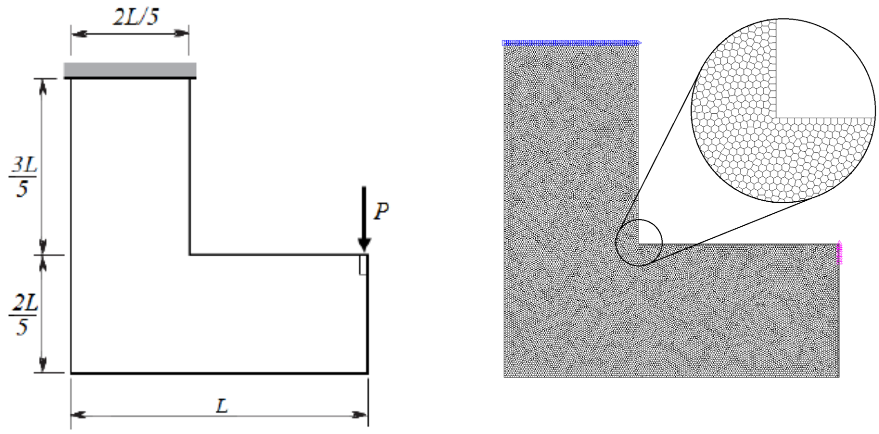
In order to calculate the stress, a large sparse matrix is stored that has each Gauss points displacement matrix of each element times the constitutive tensor at its diagonal in the variable `fem`. At the same time a array of indexes connecting each degree of freedom with its respective element is created.

Despite all of this stored variables the memory requirement isn't much higher since all the matrix stored this way are sparse and the arrays' sizes are in the order of magnitude of the degrees of freedom. This precalculation however greatly increase the speed of the code making use of the fast algebraic calculations of MATLAB. For the other terms in the structures `fem` and `opt` the reader must refer to Talischi et al. (2012b).

The variables `fem` and `opt` are used to create an anonymous function that receives only the optimization variable and returns the values and sensitivities of the objective and constraints. These anonymous functions are then passed out as an input for the optimizer that is now responsible for solving the problem.

6 Numerical Results

In the interest of testing the implementation this section presents numerical results obtained using the code developed. The 2D L-bracket benchmark problem was solved using the proposed formulation and the results are displayed below. This example demonstrates well the importance of stress constraints due to its geometrical singularity that causes an undesirable concentration of stresses at the interior corner.



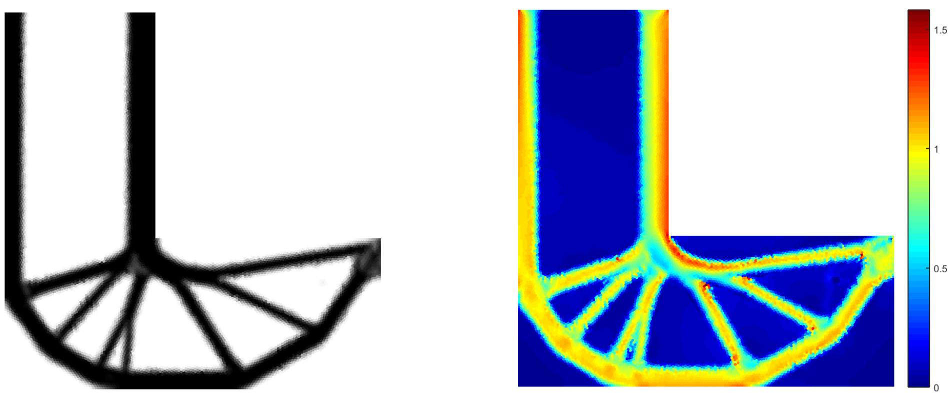
(a) L-bracket domain

(b) Polygonal mesh over the L-bracket domain

Figure 5: (a) L-bracket domain with boundary conditions. (b) polygonal mesh over the domain generated using the `PolyMesher` code (Talischi et al., 2012a).

The domain is described in Fig. 5 (a), where the length L and the model thickness are 1m. The load P is 1N and is distributed over 9 vertical elements to avoid stress concentration. The material's Young's modulus E_0 is 1 Pa and the Poisson ratio ν is 0.3 . The stress limit is 42 Pa. An unstructured mesh with 16384 polygonal elements and 65516 degrees of freedom, displayed in Fig. 5 (b), was used. As for the numerical parameter of this formulation α is set to 2.5 and δ to 0.01. The filter radius is 0.015 m. The penalization factor for the SIMP formulation is $p = 3$.

For the optimization algorithm MMA (Svanberg, 1987) the parameter *asyminit* = 0.01, *asyminc* = 1.2, *move* = 0.3 and *asymdecr* = 0.7 were adjusted for a more conservative method with a smoother convergence.



(a) Final Result

(b) Relaxed Stress

Figure 6: (a) Final result obtained using the method proposed (b) Relaxed stress to allow visualization.

Regarding the efficiency of the code the bottleneck is the solution of linear systems to find the displacements and the adjoint variable in Eq. (41). The total optimization procedure took

450 iterations and 38 min on a computer with a processor Intel Core i5-3330 CPU @ 3.00 GHz. The final weight of the result obtained is 33% of the fully filled domain, its compliance is $C = 298$ and the maximum stress is 1.6 of the stress limit. The stress displayed in Fig. 6 (b) is actually $\rho(z)^{0.5}\sigma$. This relaxation is done only for post-processing as the high numerical stress of the void region would hamper a good analysis.

It is worth to notice the smooth curve that the structure does around the sharp corner in the domain. This is characteristic of this problem with stress constraint and the results are consistent with the ones obtained by Verbart et al. (2016).

Although the high stress concentration at the corner was avoided, the stress limit was not strictly abode due to δ parameter in Eq. (19) that introduces a tolerance in the stress violation. Lowering the value of δ interferes with the optimization. This may seem like a serious issue but it is important to remember that topology optimization is just one stage in the final structure design. Its results still have to be interpreted in order to reach something that can be manufactured and this interpretation can lead to a piece with a higher stress even if the stress in the topology optimization result is completely under the limit. Moreover the resulting design can be submitted to others procedures like shape optimization to eliminate the stress violation as it is not severe.

7 Conclusions

This paper follows the `PolyTop` (Talischi et al., 2012b) philosophy where is presented a simple MATLAB code freely available that can be used for education and research purposes. The modular framework separates the finite element calculations from the optimizer making it easy to test different optimization algorithms. The objective and constraint functions have a standard format that can be used with MATLAB's optimization toolbox as well as other that use a similar input. The major changes in the code were in the analysis routine, where stresses and its sensitivities are computed. Furthermore the use of `PolyMesher` (Talischi et al., 2012a) makes it simple to test different problems with complex domains with polygonal elements.

The L-bracket topology optimization problem with stress constraint was solved using polygonal elements and the results were consistent with the literature. The implementation was able to achieve a black and white result where a severe stress concentration was not present although the structure did not strictly abode the stress limit.

For future work it is important to consider the fundamental error introduced by using assembling techniques to represent the structure overall stress. It is crucial to understand the inconsistency of this idea with the local property of the stress phenomenon. Thus a lot of caution is necessary when one makes use of this approach, as this will almost always leads to a suboptimal design. Moreover they tend to be overly sensitive to the adjustment of numerical parameter as well as to mesh refinement.

REFERENCES

Bendsøe, M. P. , 1989. "Optimal shape design as a material distribution problem". In: *Structural optimization* 1.4, pp. 193–202.

Bendsoe, M. and Sigmund, O. , 2003. *Topology Optimization: Theory, Methods, and Applications*. Engineering online library. Springer Berlin Heidelberg.

Borrvall, T. and Petersson, J. , 2001. "Topology optimization using regularized intermediate density control". In: *Computer Methods in Applied Mechanics and Engineering* 190.3738, pp. 4911 –4928.

Bourdin, B. , 2001. "Filters in topology optimization". In: *International Journal for Numerical Methods in Engineering* 50.9, pp. 2143–2158.

Bruggi, M. , 2008. "On an alternative approach to stress constraints relaxation in topology optimization". In: *Structural and Multidisciplinary Optimization* 36.2, pp. 125–141.

Cheng, G. D. and Guo, X. , 1997. "ϵ-relaxed approach in structural topology optimization". In: *Structural optimization* 13.4, pp. 258–266.

Emmendoerfer Jr., H. and Fancello, E. A. , 2014. "A level set approach for topology optimization with local stress constraints". PhD thesis. Department of Mechanical Engineering, Federal University of Santa Catarina, pp. 129–156.

Guo, X., Zhang, W. S., Wang, M. Y., and Wei, P. , 2011. "Stress-related topology optimization via level set approach". In: *Computer Methods in Applied Mechanics and Engineering* 200.4748, pp. 3439 –3452.

Holmberg, E., Torstenfelt, B., and Klarbring, A. , 2013. "Stress constrained topology optimization". In: *Structural and Multidisciplinary Optimization* 48.1, pp. 33–47.

Le, C., Norato, J., Bruns, T., Ha, C., and Tortorelli, D. , 2009. "Stress-based topology optimization for continua". In: *Structural and Multidisciplinary Optimization* 41.4, pp. 605–620.

Pars, J., Navarrina, F., Colominas, I., and Casteleiro, M. , 2010. "Block aggregation of stress constraints in topology optimization of structures." In: *Advances in Engineering Software* 41.3, pp. 433–441.

Pereira, A., Talischi, C., Paulino, G. H., M. Menezes, I. F., and Carvalho, M. S. , 2016. "Fluid flow topology optimization in `PolyTop`: stability and computational implementation". In: *Structural and Multidisciplinary Optimization*, pp. 1–20.

Pereira, J., Fancello, E., and Barcellos, C. , 2004. "Topology optimization of continuum structures with material failure constraints". In: *Structural and Multidisciplinary Optimization* 26.1, pp. 50–66.

Svanberg, K. , 1987. "The method of moving asymptotesa new method for structural optimization". In: *International Journal for Numerical Methods in Engineering* 24.2, pp. 359–373.

Talischi, C., Paulino, G. H., Pereira, A., and Menezes, I. F. M. , 2010. "Polygonal finite elements for topology optimization: A unifying paradigm". In: *International Journal for Numerical Methods in Engineering* 82.6, pp. 671–698.

Talischi, C., Paulino, G. H., Pereira, A., and Menezes, I. F. , 2012a. " `PolyMesher`: A general-purpose mesh generator for polygonal elements written in Matlab". In: *Struct. Multidiscip. Optim.* 45.3, pp. 309–328.

Talischi, C., Paulino, G. H., Pereira, A., and Menezes, I. F. , 2012b. " `PolyTop`: A Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes". In: *Struct. Multidiscip. Optim.* 45.3.

Verbart, A., Langelaar, M., and Keulen, F. v. , 2016. "Damage approach: A new method for topology optimization with local stress constraints". In: *Structural and Multidisciplinary Optimization* 53.5, pp. 1081–1098.

Yang, R. J. and Chen, C. J. , 1996. "Stress-based topology optimization". In: *Structural optimization* 12.2, pp. 98–105.